NTNU

Norwegian University of
Science and Technology

Design and Implementation of an Efficient, Reliable and Safe Work-Package Database System at CERN

Hans-Even Ramsevik Riksem

Master of Telematics - Communication Networks and Networked Services (2 year)

Submission date: October 2010

Supervisor: Bjarne Emil Helvik, ITEM

Co-supervisor: Martin Gastal, CERN, EN-MEF

Norwegian University of Science and Technology
Department of Telematics

Problem Description

The objective of this master thesis is related to the development of a database system going to be used by a work-package management application at CERN. The design, implementation, architecture and functionality applied in the database context will be described and evaluated regarding availability, safety and efficiency. Reliability issues will also be mentioned.

ER diagrams and DB relational schemes will be used to aid the description and evaluation of how the database is structured and designed. SQL statements will be provided for the implementation. A description of what Oracle solution CERN uses to increase availability and efficiency will be given, and there will also be a short description and evaluation of other techniques applied by CERN to increase reliability, availability, and efficiency.

If there should be enough time in the end, a description of how we interact with external CERN databases, will be added to the report. The access to these databases should allow retrieving and sharing information across existing systems.

Assignment given: 10. May 2010
Supervisor: Bjarne Emil Helvik, ITEM

Summary

The Activity Coordination Tool (ACT) is a web application designed to automate the planning and coordination of work packages. In the CMS experiment at CERN it is important that work packages in the underground facilities are properly planned in order to not jeopardize time schedules, equipments, budgets and safety. The subject of this thesis is the development of the database schema used by the ACT application. The schema has been developed from scratch in order to best fit the needs of CMS and to cover all aspects of the planning and coordination process not found in other CERN databases.

Models and diagrams of the database schema are based on a step by step description of the work package process. This step by step description was used to organize the data, to make the data and data relationships consistency, and to make the database structure flexible for extensions in the future. These models would eventually be used to implement the schema in the CMS online database.

Another database at CERN has many similarities with the ACT database. These two databases will eventually merge due to common interest, and the ACT database schema needed therefore to resemble some tables in this database in order to facilitate the merging. Some problems with network traversal and security needs to be solved before these two databases can be merged.

The architecture and usage of the database schema can affect the database performance but it doesn't contribute to the reliability of the databases system. The hardware and software components making up the database system itself are usually the main contributors to this. The CMS online network work hard to keep the performance and reliability of their database system as good as possible. Everything from disks to network connections is redundant. In addition to component redundancy are features provided by Oracle used to improve performance. The amount of server redundancy does however seem a bit exaggerated, and the performance could be slightly improved if other Oracle features were used.

Foreword

This thesis was performed by Hans-Even Ramsevik Riksem during the summer and autumn of 2010. It was written as a part of a *technical student* contract at CERN in Switzerland.

I want to thank my supervisors, Bjarne E. Helvik (NTNU) and Martin Gastal (CERN), for good supervision and guidance. Thank to Stephane Bally, Frank Glege, Mindaugas Janulis, Sebastian Bukowiec, James Cook, Barbara Beccati, Peter Sollander and Pedro Martel for information and guidance regarding development of the schema and for documentation about CERN databases. And finally, thank to Christian Gallrapp for startup help with LaTeX.

Contents

List of Figures

List of Tables

Glossary

EDH transport is a document similar to IS37. This type of document is created if a work activity involves use of one of the bigger cranes. The document will include the information a transport coordinator needs to perform his work.

IS37 is system for detection of e.g. fire. An IS37 request is a request for disabling a detection system in a given area. The IS37 procedure is followed when a level 3 safety system - like ODH, smoke detection, and AUG - needs to be disabled.

permits apply for both persons and for work activities. A permit explains how to protect against a danger, and it should describe how to limit the damages if there should be an accident. An example is a fire permit. It describes how to protect against fire and how to stop a fire if it should start.

work package is a subset of a project that can be assigned to a specific party for execution. A work package is defined by brief statements of activity description, resources including skills and expertise, estimates of duration, schedule, and risks. Work Packages are assigned a work authorization or control account. The abbreviation WP is often used for work packages.

Acronyms

ACT Activity Coordination Tool

ARCH Archiver

BLOB Binary Large Objects

CCC CERN Control Centre

CERN Organisation Europèenne pour la Recherche Nuclèaire (in English: The European Organization for Nuclear Research)

CKPT Checkpoint

CMS Compact Muon Solenoid

DBA Database Administrator

DBMS Database Management System

DBWR Database Writer

DCS Detector Control System

DDL Data Definition Language

DML Data Manipulation Language

DSS Detector Safety System

EAM Experimental Area Manager

ERD Entity Relationship Diagram

FK Foreign Key

FSFO Fast-Start Failover

HA High Availability

LGWR Log Writer

LHC Large Hadron Collider

LNS Log Network Server

MTF Mean Time to Failure

ODH Oxygen Deficiency Hazards

PK Primary Key

PMON Process Monitor

RAC Real Application Clusters

RAID Redundant Array of Inexpensive Disks

RDBMS Relational Database Management System

RFS Remote File Server

RMAN Recovery Manager

RP Radiation Protection

RPE Radiation Protection Experts

SCN System Change Number

SGA System Global Area

SMON System Monitor

SQL Structured Query Language

TAF Transparent Application Failover

TC Technical Coordinator

WPA Work Package Analysis

Chapter 1

Introduction

1.1 CERN

Organisation Europèenne pour la Recherche Nuclèaire (in English: The European Organization for Nuclear Research) (CERN), was founded in 1954 and is now one of the largest and most respected centers for scientific research in the world. CERN is located on the border between France and Switzerland, near Geneva. The business at CERN is fundamental physics which will be used to determine what the universe is made of and how it works [22].

CERN is run by 20 European Member States, but many non-European countries are also involved in different ways. There are over 2500 employees at CERN, and around 8000 visiting scientists (half of the world's particle physicists) come to CERN for their research. These people represent 580 universities and 85 nationalities [22].

The instruments used at CERN are particle accelerators and detectors. Accelerators boost beams of particles to high energies before they are made to collide with each other or with stationary targets (the latter happens if the beam must be dumped to protect the system when an error is detected). Detectors observe, and record, the results from collisions [22].

1.1.1 LHC

The Large Hadron Collider (LHC) is the particle accelerator used by physicists to study the fundamental building blocks of all things. Two beams (one in each direction) of subatomic particles, called hadrons, will gain energy while travelling around a 27km long circle-formed tunnel, located 100 meter under the ground. When the right energy level is reached the beams can be guided into a collision course in the four detectors around the LHC tunnel [22].

When there are collisions in LHC, bunches of particles collide close to the speed of light, and up to 40 million times per second. To not produce vast amounts of data, a trigger system saves only potentially interesting events. These triggers reduce the number of recorded events to around 100 per second. Despite this large reduction in observations, grid computing is deployed to distribute the calculations to computers all over the world [23].

1.1.2 Point 5 - The CMS Detector

There are 8 points around the LHC tunnel. Four of these points contain detectors; the other ones are used for maintenance purposes. There are two major detectors which are called ATLAS and Compact Muon Solenoid (CMS). This thesis is carried out for the CMS project. CMS is located at point 5 of the LHC tunnel [23]. Figure 1.1 gives an overview of LHCs topology.

CMS is designed to see a wide range of particles and phenomena produced by high-energy collisions in the LHC. The detector is like a giant filter, where each layer is designed to stop, track or measure a different type of particle emerging from proton-proton and heavy ion collisions. Finding the energy and momentum of a particle gives clues to its identity. Particular patterns of particles, also called signatures, are indications of new physics [23].

The CMS detector is like a cylindrical onion built around the beam pipe. This cylinder is made up of 15 slices, where the heaviest slice weighs as much as 2000 tones. The total weight of the CMS detector is about 12 500 tons (about the same weight as 30 jumbo jets), it is 15 meters in diameter and 21 meters long. In the CMS detector there is also a solenoid magnet producing a magnetic field of 4 Tesla (100 000 stronger then

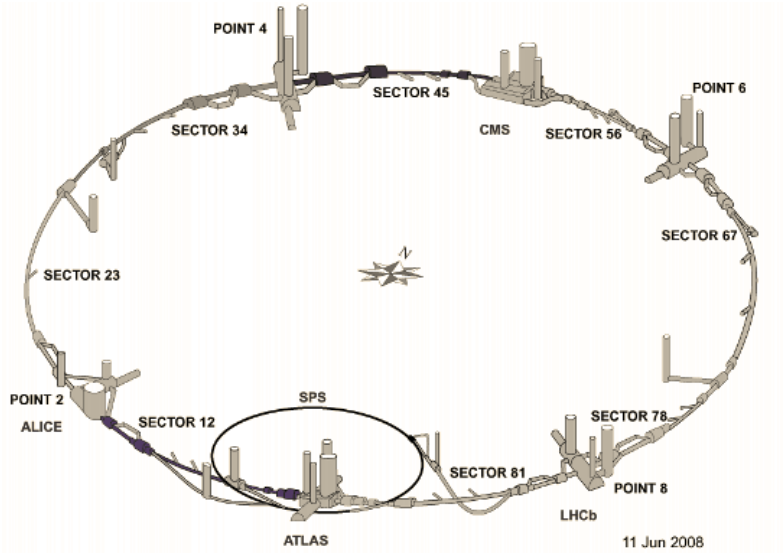

Figure 1.1: Topology overview of the LHC at CERN [40].

the magnetic field of the earth) [23]. Figure 1.2 gives an illustration of how CMS is constructed and how big it is.

1.2 Objective

Routines and procedures for planning and coordinating work activities at Point 5 are currently done by hand. This is tedious and time consuming and the Experimental Area Manager (EAM) of Point 5 has therefore proposed to develop a web application that will encompass all aspects of a work package's planning and coordination process.

The development starts from scratch and will encompass both creating a database schema and the application itself. The objectives of this thesis include developing the database schema (which will be used by the web application), to have a look at how reliability is provided by the database system, and how data is protected.

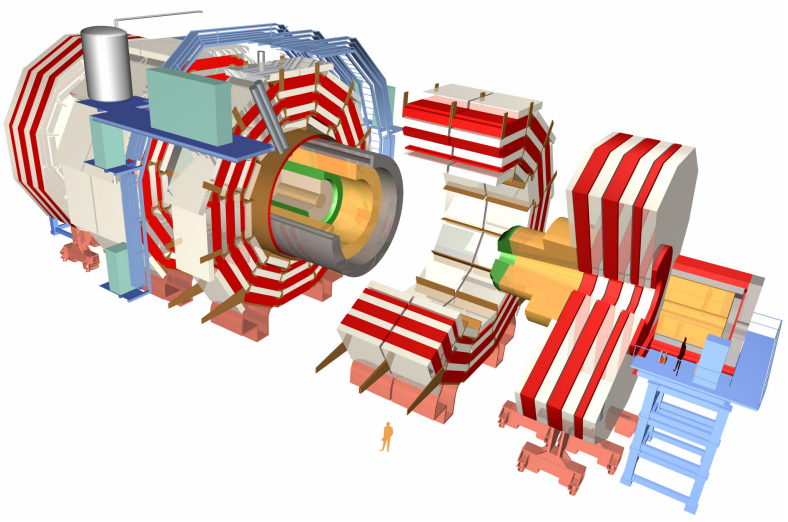

Figure 1.2: Illustration of the size and structure of the CMS detector [23].

The database schema should encompass all aspects of the planning and coordination process which is not covered by existing CERN databases. It should be easy to expand and modify, and interaction with existing database schemas should also be taken into account. The schema itself, its architecture, some common performance issues related to SQL, and guidelines used when creating the schema will be reviewed.

The database schema being developed will eventually be deployed in an Oracle database. Oracle provide a range of functionalities to reduce downtime and to protect data. The main Oracle functionalities used by CERN will be examined along with some of the main alternatives.

Component and data redundancy used at CERN will also be given a short examination.

1.3 Outline

Chapter 2 gives an introduction to the steps that a WP will go through. Examining the steps which a WP goes through will give a better understanding of what information needs to be stored in the schema. Information which will be collected from the users during the WP process will be presented in tables through this chapter.

Chapter 3 gives an overview of the existing CERN databases that will be interacted with. Through this chapter a short description of what information these external databases contain will be given. The ACT schema is implemented using SQL, and a short explanation of SQL and some performance/security issues related to SQL will be explained at the end of this chapter.

Chapter 4 covers the modeling part. Trying to implement something without modeling is not recommended and this chapter gives a description of the models used during the design of the ACT schema. Guidelines which should be applied during the design are also presented. At the end there is a presentation of the actual models and diagrams developed.

Chapter 5 provides information about how an Oracle database system handles issues related to reliability and availability. The main Oracle functionalities used by CERN will be examined along with the component and data redundancy solutions which CERN use.

Chapter 6 gives a discussion of the project.

Chapter 7 gives a conclusion of the work performed at CERN.

Chapter 2

The Work Package Process

The database schema, and the web application, is being developed from scratch. An overview of what information going to be stored in the database schema was therefore needed.

The first part of this chapter explains the difference between roles and functions which can be associated with a user of the application.

The second part of this chapter presents an overview of the steps which a WP process goes through. All steps, from a WP is initiated till it is completed are covered. Tables will present the data which users should provide through the web application. The descriptions of each step should also clarify what data needs to be collected from the user interaction.

Most WP data will be stored in the ACT schema, but there is also relevant data contained in other CERN databases. What information the existing CERN databases provide will be given a short textual description in 3.1.

2.1 Roles and Functions

In each WP, persons are assigned different responsibility areas involving different privileges. Roles and functions are used to be able to distinguish between what a person's responsibility and privilege were in the different WPs.

A role and/or function will be assigned to a person when he somehow gets involved in a WP. Some might have read only privilege to the information contained in a WP, some might have the ability to change parts of the information, and a few people can modify everything.

Being aware of the presence of roles and functions will ease the understanding of the application and what data needs to be stored. A short description of the different roles and functions is given in Appendix A along with a graphically summary of the privilege allocation.

- A **role** is a person's employment status at CERN. This information is found in EMDb and is used when determining a person's privileges for all WPs he has been involved in.

- A **function** is almost the same as the role, but it doesn't depend on a person's employment status at CERN. A function is just assigned for a specific WP. The table containing the functions is, like the role table, found in EMDb.

If a person doesn't have a role (employment status) that's relevant in the ACT, he can still make WP requests, be a supervisor or a contact person. He'll therefore be assigned one (or possible more) function for that WP.

If a person should have both a role and one, or more, functions in a WP, then the union of all the privileges found in the role and in the functions will be granted to that person for that specific WP.

2.2 WP Process Steps

An overview of the WP steps, and a coarse iteration pattern, are shown in figure 2.1. All these steps will be explained in sequence in this section.

The WP process has been divide into several steps in order to make it more lucid. If everything was placed in one step it would be a lot of information to fill in, making it seem more prohibitive for the user. An important objective with the web application is to make it easy to use. If it's not easy to use, then fewer will use it. Also, by dividing the process into steps will make it fit better with the routines and procedures which are used at CERN.

To not repeat the same details in all steps: through the whole WP sequence it should be possible to upload documentation (text documents, pictures, comments, etc.) regarding the work that's being done. It should also be possible to save the form being filled in, at all step, in order to continue the fill-in process another time.

Most information collected about a WP through these steps is stored in the ACT database.

2.2.1 WP Declaration

A person which is part of a CERN group, or experiment, explained in Appendix D can access the ACT web application. It's been decided to limit the access to these groups and experiments (and not everyone with a CERN account) because of security issues. Sensitive information is contained in this network, and access restrictions are therefore reasonable. The list of allowed groups can be expanded, if necessary. Individual persons can also be added for a shorter period of time. The majority of potential users are however found in the already added groups and experiments.

When a user is logged in to the application, he can create a new WP request. If the user is interested in viewing the WPs he has been involved in, he can browse and retrieve these WPs. The user will also have the ability to browses and search all registered WPs, but then with limited privileges (read only access to a limited amount of information). Browsing old WPs should provide the ability to reuse them when making a new request (reusing information contained in old WPs is useful since many work activities contain

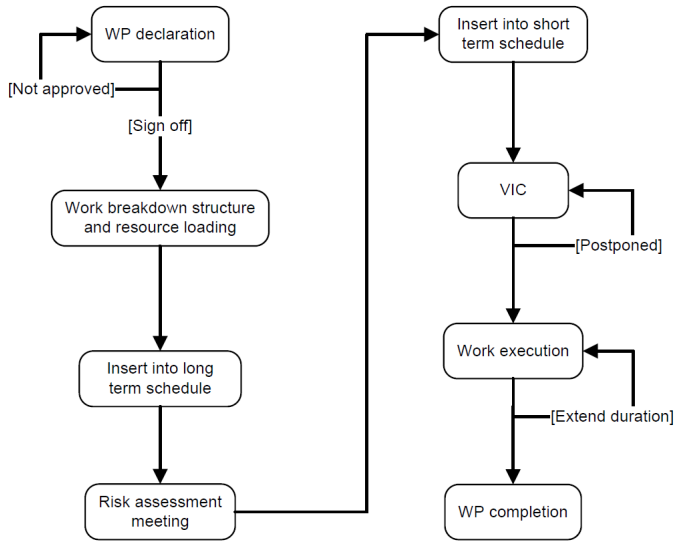

Figure 2.1: Sequence of steps which a work package goes through at CERN.

about the same information).

When creating a new WP there are two types to choose between. The options are short access WP and standard access WP. The short access type will comprise work which has a small scope and a short duration, making it unnecessary to go through all steps which a standard WP type goes through. It will have fewer fields to be filled in and it can be regarded as a light version of the standard access type. The focus in the first development, and in this thesis, is the standard access type.

When a standard access type is created, the requester will be asked to fill in basic information about the WP. This information enables EAM to decide whether or not this WP should be allowed to proceed. Table 2.1 gives an overview of the information to be filled in during the WP declaration.

When the request is submitted by the requester, EAM will receive an email notification about the new WP request. Once EAM is logged into the application, he'll have the option of viewing all unsigned WPs. He has the privilege of signing off the WP requests, but if the information provided in a request is insufficient, then EAM will ask

Information to be provided	**Description**
Supervisor	The person who will be the supervisor of this WP. This is not necessarily the same person as the requestor.
WP Title	Each WP should have a describing title.
Preferred start date	When it's preferred to perform the work.
Expected duration	How long is the work expected to last. This value does not need to be exact, but should provide an indication of the estimated duration.
Category	A WPs category represents its nature. This can be maintenance, repair, upgrade, installation, relocation, test, etc.
Priority	A WP will have a priority stating how urgent it is. The priority can be high, medium, and low.
Location	Each WP will be performed on one, or more, predefined location.
Magnet state	If the work should be performed in the underground area, the requestor must also tell if the magnet must be off when the work should be performed.
Object	If a maintenance activity takes place on a machine element call DFBX12, then the object is DFBX12. An object can be about the same as a location. It's however not the same as a location because it could be a very precise device, like "central beam pipe", which is spread over many locations.
System	An object belongs to a system. A system is a general category which enables the requestor to give more information about the involved objects. Examples of systems are the cryogenic system, specific detector parts (like ECAL and HCAL), or the cavern ventilation.
Description	A textual description of the WP which is being requested.
Schedule preference	In what type of shutdown period the work should be carried out. This can be technical stop, extended technical stop, or shutdown.

Table 2.1: Information to be filled in during the declaration stage of a WP.

Information to be provided	Description
Activity breakdown, including checkpoint	Each WP can be broken down into activities which need to be performed in a certain order. In the sequence of activities there might also be checkpoints. The checkpoints cannot be passed before the preceding activities have finished, and some sort of quality control has been performed.
Tooling list	A list of all tools involved in that WP.
Material list	A list of all materials involved in that WP.
Equipment list	A list of all equipments involved in that WP.
Stakeholders	A list of persons, like contractors, CERN service groups and collaborators, which are somehow participating in the WP.

Table 2.2: Information to be provided during the work package analysis.

the requestor to provide more precise information before signing it. This process goes on till EAM think that the provided information is sufficient, and then he'll sign the WP and allow it to proceed to the next stage.

2.2.2 Work Breakdown Structure and Resource Loading

This is the Work Package Analysis (WPA) stage which is reached when the WP request has been signed off. The person how's registered as the WP supervisor will receive a request for more detailed information regarding the planned WP.

This stage is much like the request. The WP supervisor log in to the application, he brows the WP(s) he's involved in to find the newly created WP. When this is opened a new form for filling in information is displayed. An overview of information to fill in at this stage is provided in table 2.2.

Through the WPA stage, the supervisor will need to think through the work that's going to be carried out. This should give him better overview of the work to be carried out and its complexity (some persons with the supervisor function doesn't always think too much before initiating a WP).

Not all information needs to be filled in at this stage. For example, if tools or stakehold-

ers don't exist in the CERN databases when the supervisor is filling in this information, he should be allowed to proceed with the option of filling in this information at a later point in time.

The following paragraphs are provided to give a short description of what tools, equipment and materials are in the ACT context. These descriptions should be fairly trivial.

Tools are something that's used to perform a work. Some examples of tools are drills, hammers, crowbars, etc.

Equipment can easily be mixed with tools, but there is a difference. Equipment refers to things that's brought in to the cavern and is installed in the facility. It might also be something that's brought in to replace similar equipment. Examples of equipment can be power supplies and sensors.

Material refers to necessities for performing a work. This can be screws, iron bars, pipes, a piece of wood, etc.

2.2.3 Insert into Long Term Plan

When a WPA is submitted by the supervisor, it should contain enough information for the EAM to insert that WP into a long term schedule. The long term schedule will show a rough estimate of when the different WPs can be carried out, and in what order this might happen. The long term plan can be thought of as a Gantt chart, where an updated version will be published, through the ACT application, to a web page on a regular basis. People will have access this schedule where they will have the possibility to give comments. If EAM and/or the supervisor have missed some (critical) details, they might get a reminder of this through these comments.

After a period, the schedule will be "final", and no more comments can be given regarding the schedule. The layout of the schedule is then showing the order in which the planned WPs should be performed during the shutdown.

2.2.4 Risk Assessment Meeting

A WP is subject in a risk assessment meeting 2-4 weeks before it is planned to start. The ACT application should provide the functionality needed for arranging these meetings and distributing invitations. Information about the meeting will be stored in the database along with the related WP information.

In the time between WPA submission and the risk assessment meeting, more detailed information should be provided by the WP supervisor. The information provided at this point in the process is not critical for scheduling but is used when making decisions during the risk assessment meeting. This information could also be filled in during the risk assessment meeting if it's not provided before the meeting start. If there are changes to make to already existing WP information, this can also be done during the meeting. Table 2.3 summarizes the information to provide before/during the risk assessment meeting.

During the meeting they should also have access to information about Radiation Protection (RP) in order to be able to plan when and where work can be performed. This information can be an RP map and/or values from different RP sweeps. The RP-group should have the ability to upload this information through the web application.

By using the information made available during the risk assessment meeting, the attendees can, if needed, make safety recommendations, submit a permits request, submit an IS37 request, submit an EDH transport request, create a list of potential alarms and alarm receivers, and give the WP a precise start date.

A summary of information that's needed by the AET application should also be generate. The AET information should be sent to AET 1 day before the WP is planned to start, and not when the meeting has ended. The interaction with AET is however not prioritized in the first versions of the application since AET also is being developed at the time of writing.

2.2.5 Insert into short term schedule

When the risk assessment meeting is completed, EAM has enough information to put this WP into a short term schedule. This schedule will show day by day information

Information to be provided	**Description**
Name of workers	These names will be used to check if the workers have the right access rights and the right safety (CTA) training.
Work methods	This is a document containing information about the work that's going to be done.
Expected waste	The expected waste field should tell what waste might be produced during the WP. If a grinder is used, then there will be some metal waste, if a WP involves some pipes that's filled with a liquid, then there might be some liquid waste, etc.
Need shutdown of neighboring facilities	If a WP will affect neighboring facilities, this should be stated here. E.g. if a WP is close to some equipment that's connected to high voltage, it might be vice to turn off this high voltage equipment before starting the work.
List of materials going in/out (or to relocate)	Since equipment in the cavern gets radioactive during operation, CERN is forced by the law of France (since Point 5 is located in France) to register these products and make them traceable. There is already a system that's tailored for this, so the purpose of this field in the ACT application is to flag if there should be any movement on any equipment.

Table 2.3: Information to provide before/during the risk assessment meeting.

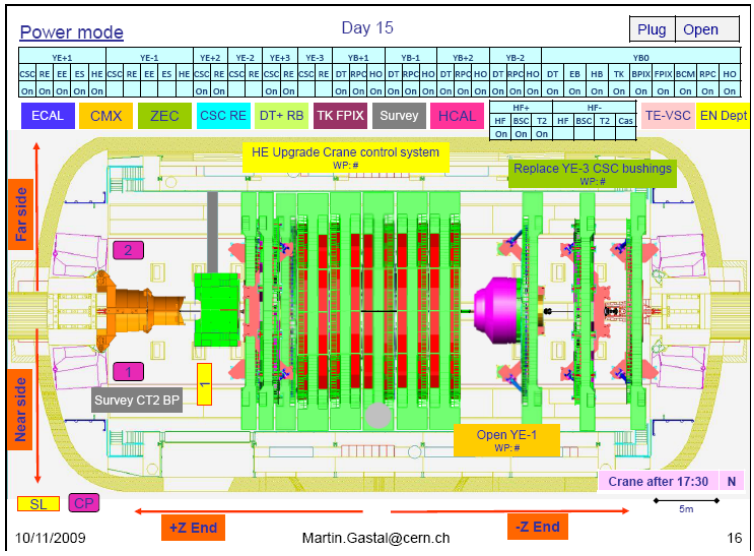

Figure 2.2: Illustration of a day by day schedule for WPs.

and might look like figure 2.2.

Some of the rectangles in figure 2.2 represent WPs that's ongoing that day. The schedule is published on a web page, through the ACT application, in order for people to see what's going on. In a later version of the application should the rectangles, which are representing the WPs, be click-able in order to display information about the WPs of interest.

The short term schedule will provide the CMS control room valuable information. The shift leaders will, with this schedule, have the ability to check information like start date, duration, locations and how is going to perform the work. By using this information, they will have a better view of what's going on, where the activities are located, and who is supposed to perform the work.

2.2.6 VIC

One day before a WP is scheduled to start there will be a safety inspection. The safety inspection will check if the safety recommendations, made during the risk assessment meeting, are applied before the WP is allowed to start. VIC is the name of the meeting which goes through the safety issues. If there should be something that's not fulfilling the requirements, the WP might get its start date postponed, or in the worst case, the WP will be canceled.

If a WP's start date is postponed, there will be a new safety inspection, and a new VIC meeting, before the WP can try to start again.

The person performing the safety inspection should be able to add documentation about the inspection. If a WP is postponed or canceled, then documenting this is important.

2.2.7 Work Execution

The name of this stage should make it quite self-explanatory. This is where the work is actually performed.

If a RP sweep is needed before the work start, then predefined measurement locations should be given to the Radiation Protection Experts (RPE).

Progress reports and other types of documentation can be added to the WP data during the work execution. If the scope has to change during the work execution (if there's an unforeseen problem), the end date of the WP should be changeable.

In later versions of the application, a change in the end date of a WP should automatically extend access rights for the people involved in that WP. This will be managed by the AET database system. AET is, at the time of writing, not providing this type of functionality since it is still begin developed. It's also a bit challenging to implement this type of functionality in AET, and it might therefore be a while the functionality is available.

2.2.8 WP Completion

When the work is completed there will be different types of documentation that should be uploaded (this can be information relating to nonconformities, lessons learned, setbacks, unforeseen situations, pictures, etc.). It's also important that the supervisor confirms that the WP is completed (either if it was successful or not successful). The supervisor confirms a WP completion when he closes it.

Chapter 3

Databases and SQL

There are several databases at CERN which contain information needed during the planning and coordination process of a WP. Interacting with these databases will make the system more complex and can potentially create some security issues. By reading this chapter, one should obtain an overview of the databases needed by ACT and a short description of some database related challenges encountered.

All WP related data could have been placed in the ACT database schema, but that solution could easily create synchronization problems with existing CERN databases. Duplicating the same data into many databases is not a good idea, and is also the reason why existing CERN databases are included.

This chapter will first provide an overview of the databases that's of interest for the ACT application. Then there will be a short explanation of SQL, and how SQL is used in an Oracle database. Some SQL related issues regarding performance and security will be mentioned before giving a short introduction to database links. The section about database links is added because interactions with external databases use database links.

3.1 Databases that ACT Will Access

Seven databases were selected from the myriad of databases at CERN. At the time of writing, most of these databases have provided read access to a selected part of their data. EMDb, ADaMS, Foundation and the self developed database schema are the ones of interest during the first development phase.

Trying to integrate all seven databases in the first release is not feasible. Knowing that these databases were going to be used, the schema could be formed in a way which facilitated expansions to these databases in the future. Some of these databases does however not provide too much information, and is intended for adding functionality at a later point in the development. Some of these databases is also not going to be used directly, but is added because they provide, through an interface, a service which will be useful for ACT.

Figure 3.1 provides an overview of all the databases that somehow are involved in the ACT application. The ACT schema is contained in the database used by the CMS online network.

A short explanation about what information these databases provide is found in Appendix E.

3.2 SQL

Structured Query Language (SQL) is used to create the database schema in the database, and to access the data contained in that schema. This section is not intending to explain how SQL works, but will give some basic SQL information and mention some performance issues which one should be aware of when designing and implementing a database schema. After reading this section it should be a bit clearer why the different concepts is used in the ACT schema, and to identify issues related to these concepts.

SQL comprises one of the fundamental building blocks of modern database architecture, and is used in the majority of database systems existing today. SQL is an ANSI and ISO standard defining the methods used to create and manipulate relational databases [41, 42].

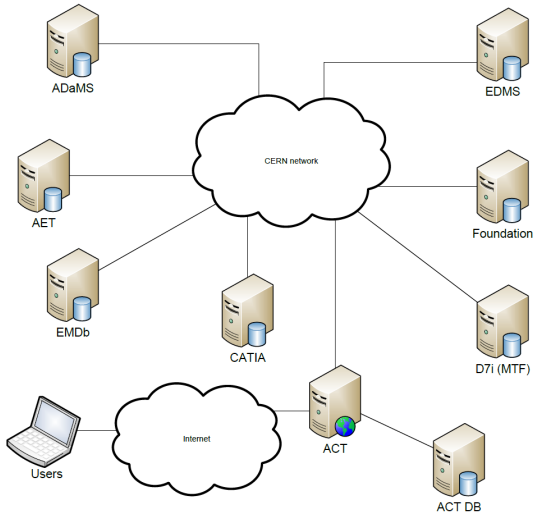

Figure 3.1: Overview of all databases that somehow are involved in the ACT application.

SQL is a nonprocedural language; users describe in SQL what they want done, and the SQL language compiler automatically generates a procedure to navigate the database and perform the desired task.

3.2.1 Constraints

There are many types of constraints in SQL. The most common constraints used when creating the ACT schema are summarized in following bullet points [53]:

- The primary key constraint uniquely identifies each row (also referred to as a record) in a database table. A primary key constraint automatically has a unique constraint defined on it. A primary key column can also never contain null values.

- A foreign key in one table points to a primary key in another table.

- The not null constraint enforces a field to always contain a value. This means that one cannot insert a new record, or update a record, without adding a value to this field.
- The unique and primary key constraints both provide a guarantee for uniqueness for a column or set of columns. Note that there can be many unique constraints per table, but only one primary key constraint per table.
- The default constraint is used to insert a default value into a column. The default value will be added to all new records, if no other value is specified
- The check constraints limit the value range that can be placed in a column. If there is a check constraint on a single column it allows only certain values for this column.

3.2.2 View

In order to let other applications access information contained in the ACT schemas, views were created. Views were also created in external databases in order for the ACT application to access information contained in these databases.

In SQL, a view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. A view is essentially very close to a real database table, except for the fact that a real table store data, while a view is a set of SQL queries which will result in a table [43, 44]. Figure 3.2 give an illustration of a view usage.

One advantage of views is that they hide the complexity of the underlying business logic for the end users and/or external applications. Another benefit of views is that they can have computed columns (a column containing a result of an operation from other columns) [44].

The main benefit of views in the ACT context is the security. A view facilitates sharing of a limited amount of information to the external users which is interested in the information contained in the ACT database. By creating a view, containing the information of interest, it's easier to ensure that users are only able to retrieve (and in rare cases, modify) the data provided through the view. The remaining data in the underlying tables are not accessible for the external users [43, 44].

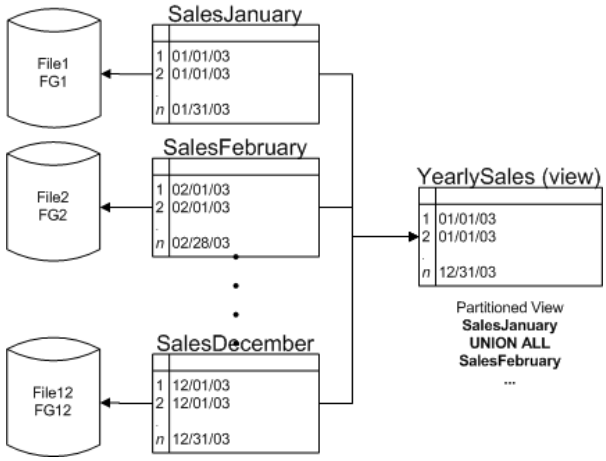

Figure 3.2: This illustrates a view. The sales data is contained in separate tables, one for each month. When the sale for the whole year is needed, a view is created of all the month sales, in order to not duplicate information into many tables [56].

3.2.3 Index

Indexes are used in Oracle to provide quick access to rows in a table. Indexes were used to improve the performance of the ACT schema [46].

Oracle does not limit the number of indexes you can create on a table, so it might be tempting to just add indexes to everything, since it's making the database access go faster. However, the more indexes added, the more overhead is incurred as the table is altered [46]. Having a large number of indexes on a table will most likely result in faster select statements, but slower insert, update, and delete statements [50]. The difference between an ordinary table and an index-organized table is:

- A row in an ordinary table, were indexes aren't applied, has a stable physical location. Once it's given its first physical location, it never completely moves. Even if the row is partially moved with the addition of new data, there is always a row piece at the original physical address (identified by the original physical row id) from which the system can find the rest of the row. As long as the row exists, its physical row id does not change [47].

- A row in an index-organized table does not have a stable physical location. An index-organized table is, on the one hand, like an ordinary table with an index on one or more of its columns. It's unique, in that it holds its data, not in stable rows, but in sorted order in the leaves of a B*-tree[1] index built on the table's primary key. These rows may move around to retain the sorted order. Changes to index-organized table data (for example, adding new rows, or updating or deleting existing rows) can cause an index leaf to split and the existing row to be moved to a different slot, or even to a different block [47]. The index-organized table will eliminate one I/O, namely, the read of the table, and is therefore more efficient [47].

If a table is mostly accessed as read-only, then using indexes might be useful. If, however, a table is heavily updated, then the overhead introduced due to reorganizing the B*-tree will reduce the performance instead of improving it. So, there is a trade-off between the speed of retrieving table data and the speed of accomplishing updates on the table [46].

In the ACT schema, indexes were created on certain columns which will be accesses frequently. To not inhibit the performance too much, excessive use of indexes were avoiding. The main guidelines followed when creating indexes was found in [46] and they can be summarized to:

- Columns used for joins should be indexed to improve performance on joins of multiple tables.
- One should create indexes if less than 15% of the rows in a large table are frequently retrieved.
- Small tables do not require indexes.

3.2.4 Join

When reading data from the ACT schema, there is sometimes a need to read related information contained in two or more tables. Join operations are used to obtain this.

[1] The B-tree data structure is a standard for organizing indexes in a database system. The B-tree guarantees at least 50% storage utilization, that is, at any given time, the tree has each of its nodes at least 50% full. The B*- tree is an improvement to the B-tree and it guarantees 66% storage utilization [48].

These can, if not used correctly, impact the performance/response time of the database.

The join keyword is used in an SQL statement to query data from two or more tables, based on a relationship between certain columns in these tables. By relating tables in the database and by using join operations, one can bind data together, across tables, without repeating all of the data in every table [49].

Since foreign keys (which relate tables in a database to each other) are often used in joins, creating an index on any foreign key can improve performance [50].

Table joins can be a big contributor of performance problems, especially if the joins include more than two tables, or if the tables are very large. Here are some tips provided by [51, 52] which can help optimizing joins:

- If you have two or more tables that are frequently joined together, then the columns used for the joins should have an appropriate index.
- For best performance, the columns used in joins should be of the same data types. And if possible, they should be numeric data types rather than character types.
- Avoid joining tables based on columns with few unique values. For best performance, joins should be done on columns that have unique indexes.
- If you have to regularly join four or more tables to get the record-set you need, consider a lower normalization level of the tables so that the number of joined tables is reduced. Often, by adding one or two columns from one table to another, joins can be reduced.

Join operations are something that cannot be precluded since information sometimes need to be obtained from two, or more, tables. In order to not reduce the performance of the database schema, a moderate usage of join operations have been deployed.

When designing a schema, one should have join operations in mind. Join operations contradict normalization (explained in 4.1 on page 30), but both concepts should be represented in the final diagram. This is a balance between performance and manageability.

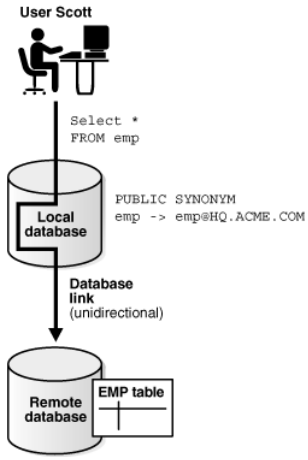

Figure 3.3: This illustration depicts a database link between the user, Scott, and a remote database. The link passes through the local database, and because the link to the remote database is stored on the local database, information contained in the remote database is accessible [55].

3.3 Database Links

Not all information needed by the ACT application is located in the database system used by the CMS online network. Database links is used to get access to information contained in the other databases. Figure 3.3 gives an illustration of how a database link should work.

A database link is a connection between two physical database servers that allows a client to access them as one logical database. The great advantage of database links is that a local user can, through the use of a database link, access a remote database without having to be a user on the remote database. Database links act as a pointer that defines a one-way[2] communication path from an Oracle database server to another database server. The link pointer is defined as an entry in a data dictionary table, and to access the link one must be connected to the *local* database that contains the data

[2] A database link connection is one-way in the sense that a client connected to a local database A can use a link stored in database A to access information in a remote database B, but users connected to database B cannot use the same link to access data in database A [54].

dictionary entry [54].

Database links are either private or public. If they are private, then only the user who created the link has access; if they are public, then all database users have access. Determine the type of database links to employ in a distributed database depends on the specific requirements of the application using the system. Private database links are used in the ACT schema since they are more secure then public or global links.

When creating the link, one should also determine which user should connect to the remote database to access the data. A *fixed user* link is used for the ACT schema since the user then needs a username/password to connect to the remote database. A benefit of a fixed user link is that it connects a user in a primary database to a remote database with the security context of the user specified in the connect string. The username and password associated with the user are stored with other link information in data dictionary tables.

There is however a problem which is limiting usability of database links in ACT. The problem encountered is that the database where the ACT schema is deployed, and much other information is held, is inside a network with strict security rules. External databases can be accessed from the inside of this network, but from the outside it's hard to establish a connection to anything inside the network.

Database links are used inside the CMS online network, and to access some information on the outside. The databases on the outside will however not be able to get information from the ACT database schema. This will limit the amount of potential users of the ACT database schema, but access restrictions are essential in a network containing so much sensitive information.

Chapter 4

Modeling

So far, most focus has been on describing how things work at CERN. Since the development started from scratch, it was important to have good background knowledge before actually starting to create the models of the system.

The process of modeling the system and achieving new knowledge about the system is often associated with each other. The models presented in this chapter have been redrawn several times due to new knowledge and requirements. The history of the models development is not mentioned, and only the resulting models are presented.

Both ER diagram and relational DB schemas were created when modeling the system. Before explaining these model types, and presenting the resulting models, normalization will be looked at. Normalization is a set of guidelines which apply for both the ER diagrams and the relational DB schemas. After the normalization is explained, the ER diagrams will be examined. The ER diagrams were used in the initial part of the modeling to create a conceptual model of the schema. The ER diagrams made for the ACT schema will also be presented here. At the end of this chapter will the relational DB schemas of ACT be presented along with the process of transforming ER diagrams into relational DB schemas.

4.1 Normalization

Schema design is a key performance factor and is largely a tradeoff between good read performance and good write performance. In short; normalization helps write performance, denormalization helps read performance [50].

Normalization was used when creating the models of the database system. By using normalization, the resulting database schema will be more flexible for changes in the future. The impact normalization have on the amount of join operations (see 3.2.4 on page 24) was also taken into account.

Normalization is a process of organizing and structuring data in a database, where the main goals are: eliminating redundant data and ensuring that data dependencies make sense [30, 31, 32].

The guidelines for database normalization are divided into 5 levels. These normalization levels are often called normal forms. The lowest level of normalization is referred to as the first normal form (1NF). If the three first levels of guidelines are applied, the database is considered to be in third normal form (3NF) [30, 31]. When moving up one normal form level, all requirements of lower normal forms must be satisfied [29].

4NF and 5NF are seldom seen in practice and they are not applied in the ACT schema. The ACT schema aim to meet the requirements for the third normal form.

Knowing the principles of normalization and applying them to database design isn't all that complicated and it can drastically improve the performance of a Database Management System (DBMS). Normalization also decreases the chance that the database integrity could be compromised due to tedious maintenance procedures [30, 31, 32].

As with many formal rules and specifications, real world scenarios do not always allow for perfect compliance with the normalization guidelines [31]. A complete normalization of tables is of course desirable, but is sometimes not practical. With normalization, the amount of tables will increase. More tables will most likely result in more join operations (see 3.2.4 on page 24). More join operations is quite costly, resulting in decreased performance for most for most database management systems [29].

When designing the ACT schema, some persons recommended to break the normalization guidelines in order to improve the schema's performance. This recommendation

was based on the cost of join operations. Other persons recommended obeying the normalization guidelines, and this is what eventually was done. The resulting schema will be more flexible for changes in the future when it's obeying the normalization guidelines. The performance will be slightly slower, but much performance can be gained if the joins are performed the right way.

The following subsections gives a short overview of the requirements implied by the three first normalization levels.

4.1.1 First Normal Form

- Related data should be put in separate tables [32].
- Each value contained in columns of a table should be atomic[1] [29, 30].
- There shouldn't be multiple columns in a table storing similar information [31, 32].
- Each set of data (i.e. each row) contained in these tables should also be uniquely identified by a primary key [29, 30].

4.1.2 Second Normal Form

- The first normal form must be fulfilled [29, 30].
- Redundant/repeating data should be placed in separate tables, and then be related using foreign keys [30, 31, 32].

4.1.3 Third Normal Form

- All 2NF requirements must be met.

[1] By atomic it's meant that there are no sets of values within a column. If a table contains columns which are named in plural, this is a good indicator that the data contained in these columns can be put into a separate table or be split up in more columns [29].

– All columns in a table should depend directly on the primary key (if the primary key is a composed key, the columns should depend on all the values which make up the primary key) [29, 30, 32]. One column depending on another column which in turns depends on the primary key is called a transitive dependency, and this does not apply with 3NF [29, 31].

4.2 ER diagrams

Entity Relationship Diagram (ERD), also referred to as ER diagram, is a high-level data model that's useful when developing a conceptual design of a database. ER diagrams are usually drawn early in the development stage. These diagrams will help designer(s) to better understand the system being developed, to specify the desired components of the database, and the relationships among those components [24, 25].

ER diagrams are static representations of the logical structure of a database. The diagrams provide good information when trying to understand how a database works [26].

A short explanation of basic ER diagram symbols is presented in Appendix F.

4.2.1 ER Diagrams for the ACT Database

ER diagrams were created in the early stage of the ACT application[2]. During the first development of the ER diagrams, the aim was to store all information related to a WP in the ACT database schema. This changed when discovering some CERN databases already containing relevant information (like location information found in EMDb, the tooling information in MTF, 3D images contained in CATIA, etc.). The first drafts of the ER diagram did however provide a better understand about how data should be involved and the relationship between this data.

The ER diagram didn't become perfect in the first attempt. Much data should be organized and it should also fulfill the requirements of the third normal form. The process

[2]The ER diagrams for the ACT database were created using Microsoft Visio.

of presenting a solution, redesigning it and present again was repeated several times. Part of the resulting ER diagram is shown in figure 4.1.

Most of the tables are named in a way which should make them as self-explanatory as possible. There is however some tables which might need some extra explanation. For a complete set of diagrams, and a short description of the majority of entities found in the diagram, see Appendix B.

The ER diagrams also show some tables contained in external CERN databases that will be used by ACT.

4.2.1.1 Why Some Tables are Duplicated from EMDb

The ACT schema contains tables which already existing in EMDb. There are three tables which are duplicated, if not counting the tables created by the relation between the tables. These tables are the WP table, the WP Persons table and the Equipment table.

Duplication of data is in general not a good idea. The reason for duplicating some tables anyway is because the tables are used frequently, write access is needed, and EMDb is located in another network. The network where EMDb is deployed is also not under the control of CMS personnel. If the existing schema in EMDb had been used, control over the schema would be reduced.

The WP table, and the WP Persons table, is almost identical to the Interventions[3] table and the Int_persons tables found in EMDb. The same structure have been used in the tables because the WP functionality found in EMDb will eventually migrate to ACT. By using the same structure, the migration will be easier.

The Equipment table is strictly speaking not duplicated. The equipment table found in EMDb covers the equipments in the radioactive area of Point 5. ACT does however need to store information about all equipments. An equipment table has therefore been created, which will cover the equipments not found in EMDb.

ACT will eventually take over the WP functionality found in the EMDb application.

[3] What's referred to as a work package in ACT, is referred to as an intervention in EMDb.

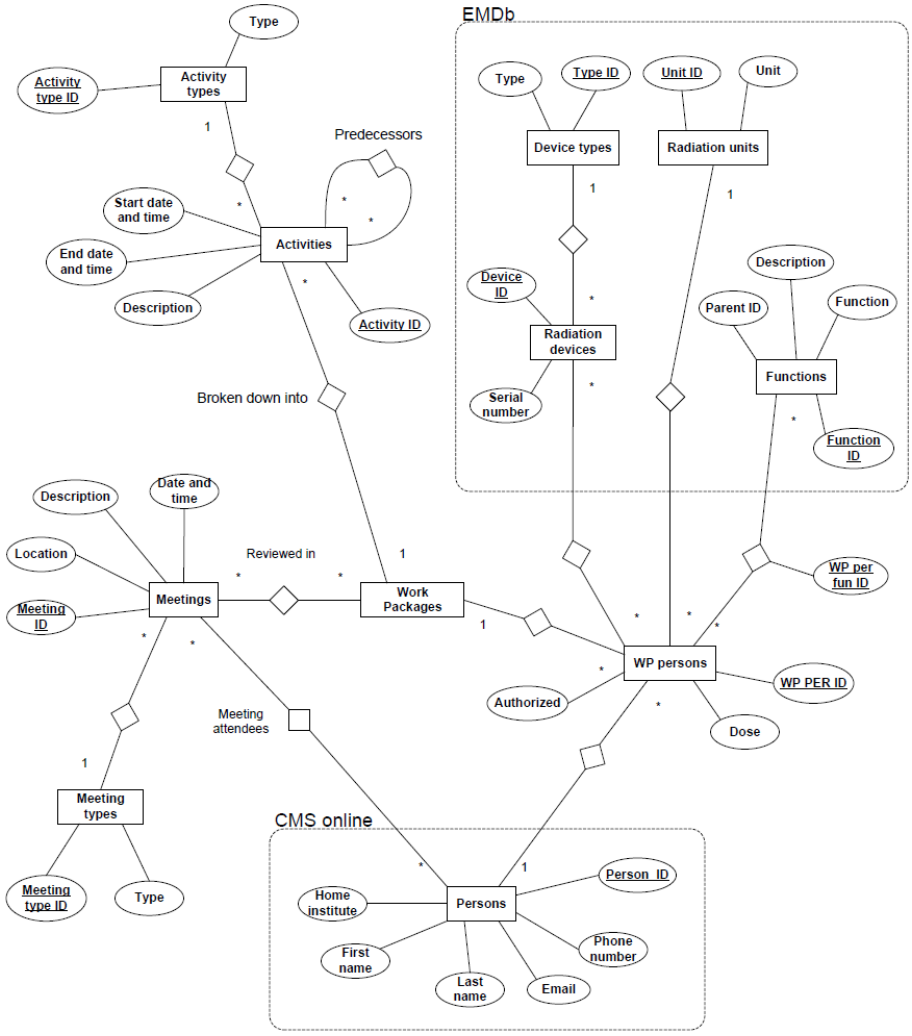

Figure 4.1: Part of the ER diagram created for ACT.

Even though ACT should handle the WPs, the EMDb application will still need to access data stored about WPs. This is where it becomes a bit tricky. The problem is that the ACT schema is deployed inside the CMS online network, and EMDb is placed in a more general CERN network. The CMS online network is very strict regarding connections going in to the network, for security reasons. It is therefore not be possible for EMDb to access the ACT data. The temporary solution is that ACT will make a copy into a offline table, which EMDb can access to retrieve the wanted information. In the future, both these databases aim to be on the same network.

4.2.1.2 CMS online Copy of External Information

The CMS online database makes a copy of some database schemas found in external database. This includes information about persons, which is originally in Foundation. The copied information is used in order to avoid accessing another external database.

The disadvantage of using the local copy is its update interval. The current update interval is once a day. This is ok in most cases but consider the following scenario:

A worker for a WP arrives the day of which the work is supposed to start. He's then using the morning to register in the CERN database, and will be registered in the Foundation database in the afternoon. This information is however not visible for the ACT application before the CMS online database has made a new copy.

A 24 hour resolution in the update frequency is however accepted by the persons who are going to use the ACT application. The benefit of using the local copy is also that you don't need another connection to an external database.

4.2.1.3 The Other Databases

If comparing the diagrams in Appendix B and figure 3.1, it can be seen that not all the databases are included in the diagrams. EDMS, ADaMS and CATIA provide services like file storage, access control information, and 3D images respectively. Neither read or write access is needed to these databases. The ACT application will use the service provided to obtain the information of interest.

AET isn't functional at the time of writing, since it is still being developed. AET will, like EDMS, ADaMS and CATIA, just provide a service for ACT. AET is included since it's planned to be used in the future, and should be part of the planning process.

4.3 Relational Database

A database is simply a collection of data. The data is organized in tables where the table's columns categories what information is held by that table. Each row is a set of columns, and all rows from the same table have the same set of columns assosiated with it[4]. Tables in a database are linked together in order to describe the relationship between them [34, 36, 38].

In addition to being relatively easy to create and access, a relational database has the important advantage of being easy to extend [36]. The "relational" part refers to how data inside a table is related and how tables in the database are stored and organized [35]. The "relational" part also stem from the fact that a relational database is based on the principles of relational algebra [37].

Relational tables follow certain integrity rules to ensure that the data they contain stay accurate and are always accessible. This is obtained through normalization of the database (see 4.1 on page 30). All operations on a relational database are performed on the tables themselves or produce another table as a result [38].

4.3.1 Relational DB Schema

The ER diagram represents the conceptual level of database design. It's used during the design of a database and it is main objective is to represent the real world by capturing the application requirements.

A relational schema represents the logical level of database design, and is closer to what the final database schema will look like. The logical design attempts to map a conceptual ER diagram to a format that can be accepted by a database system [33, 39].

[4] Tables are sometimes also referred to as a relation, columns are sometimes referred to as fields, and rows are sometimes referred to as a record or entries.

It's easier to implement a database when referring to a relational DB schema, compared to using ER diagrams. Before looking at the relational schemas for ACT, the process of transforming an ER diagram to a relational DB schema will be presented.

4.3.1.1 Mapping ER Diagram to a Relational Schema

Mapping a conceptual ER diagram to a relational schema isn't always straightforward. Concepts found in the ER diagrams may not be implemented directly in tables, and text-based representation of tables (like SQL) isn't as comprehensive as the graphical models. Thus, pre-mapping work should be performed on the ER diagrams in order to bridge the gap. However, not all transformations are possible to fulfill without losing information.

The most usual pre-mapping is to convert all n-ary ($n > 2$) relationships into binary relationships. For example: in a ternary relationship ($n = 3$) it can be quite tricky to understand the meaning of all the relationships. By replacing the ternary relationship with binary relationships opens for more semantics about the real-world situation [39].

When transforming an n-ary relationship into binary relationships, there is a chance of losing information. The ER diagrams made for ACT didn't contain any ternary (or higher) relationships, and information loss due to transformations was therefore avoided.

If the ER diagram contain multi valued attributes (repeating values), these should be separated into separate entities, and all weak entities should be turned into regular entities. Neither of these was present in the ER diagram for ACT.

After performing these pre-mapping operations, the ER model is ready to be mapped to relation schemas in a rather straightforward way [39]. In ACT there were no need to do any of these pre-mapping operations. This reduced the amount of time used when creating the relational schemas.

[33, 39] describes how to transform an ER diagram into a relational DB schema. It should be noted that this list is not a comprehensive description, and it includes only the transforms that's relevant for the ACT schema.

- An entity, in the ER diagram, turns into a table.
- Each attribute of an entity turns into a column in the resulting table. The Primary Key (PK) of an entity will also be the primary key in table. A primary key in the table will have "PK" written in front of it.
- All relationships will in some way turn into a Foreign Key (FK), but where these FK is placed depends on the cardinality constraint present for that relationship. The following bullet points summarize how to transform the different relationships, found in an ER diagram, into relationships in a relational DB schema:
  - To represent a 1:M (one-to-many or many-to-one) relationship, take the primary key from the table on the "1" side and insert it as a foreign key into the table on the "M" side.
  - To represent a 1:1 relationship there are two options. One should consider if it makes more sense to leave this as two separate tables, or to join the two tables together. This will depend on whether records will usually exist in both tables, how data will be accesses, if joins will be made constantly or only occasionally when data is queried, etc. If choosing to retain two tables for a 1:1 relationship, where should the foreign key be placed? If the relationship is mandatory for one entity but not the other, then put foreign key into the table for which participation is mandatory. If the relationship is mandatory on both sides, there is probably little point in representing it as two tables. If it is mandatory for neither, put the foreign key in the table that will be accessed less frequently – this minimizes the number of joins required when querying.
  - There is no direct representation of a M:N relationship in the relational model. You will need to turn each M:N relationship into a separate table of its own. This table will usually have its own primary key which is obtained from a combination of two foreign keys – stemming from the primary keys of the tables in that relationship.

If the 1:1 or 1:M relationship is mandatory, you will need a "not null" constraint on the foreign key. This means that a row cannot exist in this table without a related row on the other side of the relationship.

If a relationship has attributes, then they need to go into a table. Where to put these attributes depends on the type of the relationship. In a 1:1 or 1:M relationship, put them

the same place the foreign key goes. In a M:N relationship, put them in the new table you create for that relationship [33, 39].

4.3.2 Relational DB Schema for the ACT Database

The relational DB schema provided in this section[5] gives the result from translating the ER diagrams. Figure 4.2 show a part of the resulting relational DB schema. The complete relational schema can be found in Appendix C.

The relational DB schemas for ACT represent the same information as the ER diagrams, only in a different way. For more information about the tables see the section about the ER diagrams for ACT (see 4.2.1), and/or Appendix B.

The resulting schema is consistent with the initial plans. The schema doesn't duplicate too much information, it's able to cooperate with existing CERN databases, and most important, it covers the data which is needed for the planning and coordination processes at Point 5. Some compromises have been made, but the final result is suitable for everyone affected by this development.

[5] The relational DB schemas were created using the DbSchema database tool.

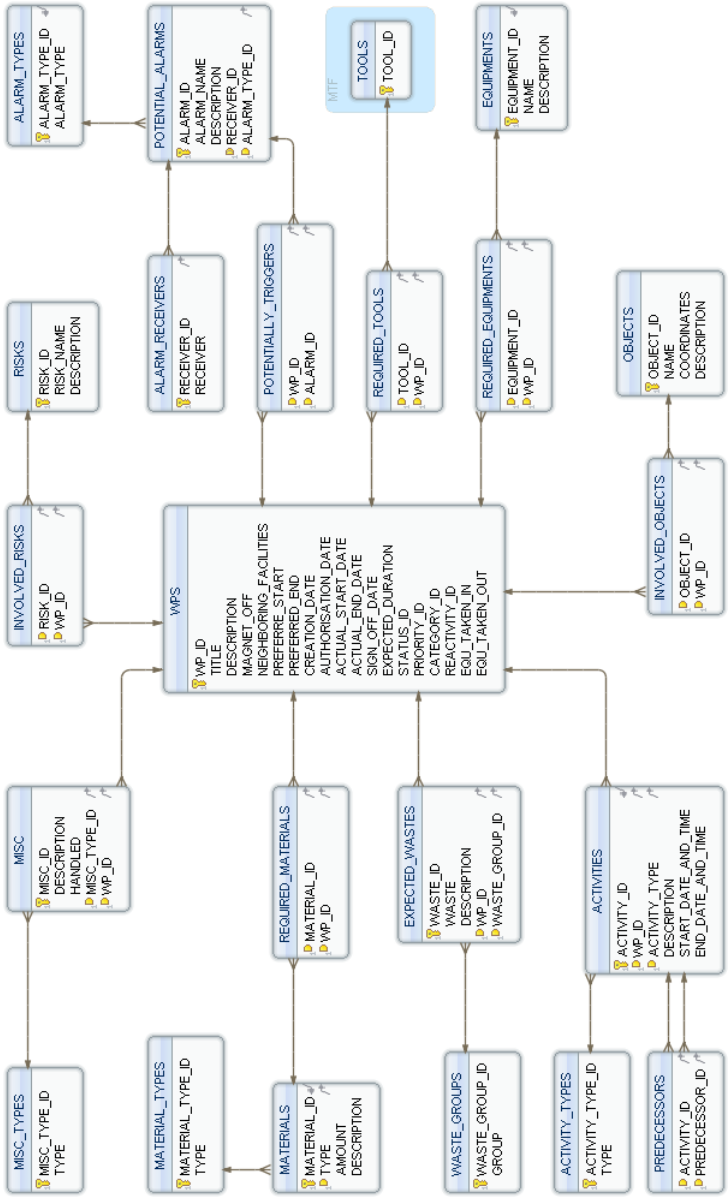

Figure 4.2: Part of the relational schema created for ACT.

Chapter 5

Oracle Database and Component Redundancy

The chapters so far have focused on the WP concept, the diagrams created, and some performance issues related to SQL and normalization. This chapter is not directly related to the work done in the thesis, but is explaining the system on which the resulting ACT schema will be deployed.

The structure and usage of the schema might have an impact on the performance but it doesn't impact reliability and stability of the database on which it is deployed. The hardware and software making up the database system are the main contributors to this.

High availability will be defined in the start of this chapter, and an overview of factors affecting the availability will be given. The Oracle features RAC, RMAN and Data Guard will then be presented respectively. These features are used in the CMS online network in order to increase performance, reduce downtime and reduce the chances of data loss. At the end of this chapter there will be a description of Redundant Array of Inexpensive Disks (RAID) and network redundancy. These are also deployed by CERN in order to improve reliability and availability of the database system.

When reading about these features and utilities provided by Oracle, there are some

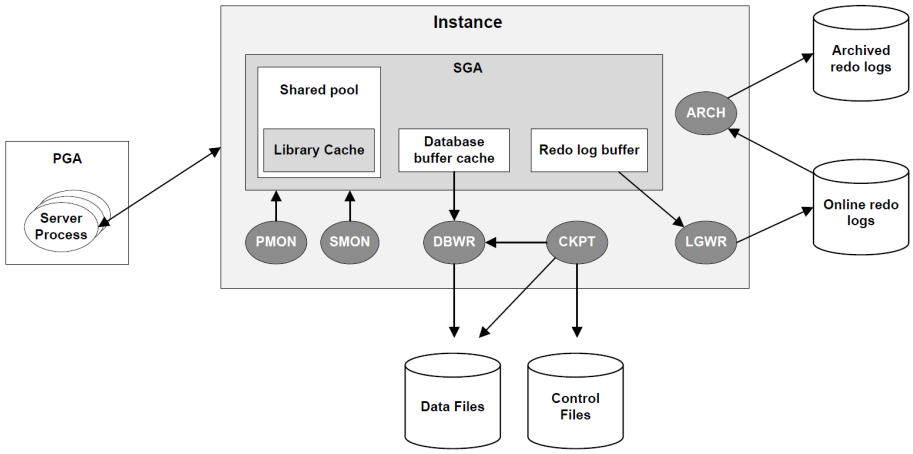

Figure 5.1: Fundamental processes for memory and storage interactions between an instance and an Oracle database. This figure is based on information found in [1, 2, 20].

technical terms and concepts appearing. A short explanation of the main terms and concepts used in an Oracle database system can be found in Appendix G.

Figure 5.1 provides an overview of how the different background processes and memory structures interact in an Oracle database. Information found in this figure might facilitate the understanding of this chapter.

5.1 High Availability

A system is defined as available when it can be accessed by users and at the same time provide the expected functionalities at the expected performance [1].

The importance of high availability will of course vary from application to application, but when designing a High Availability (HA) solution it's important to consider causes of both unplanned and planned downtime [17]. A comprehensive HA solution eliminates single points of failure and should keep users unaware of failures in the system [2, 4].

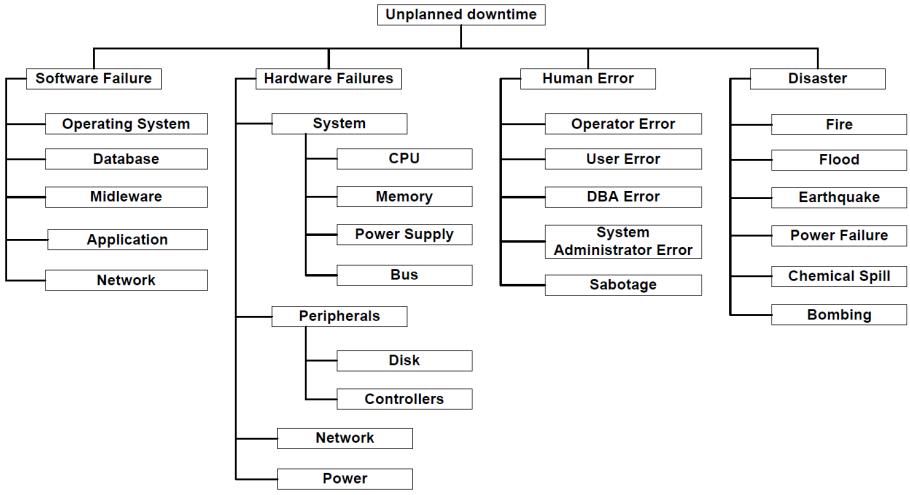

Figure 5.2: Rough overview of common causes of unplanned downtime. This figure is based on figure 11-1 in [1].

High availability is important for the databases contained in the CMS online network at CERN. These databases contain information which is, among others, used by shift leaders. An outage of this system will hamper the possibility to continue an ongoing operation.

All components in the total system can contribute to potential causes of downtime. Some of the causes of downtime can be prevented quite easily, while others might require investments [1]. The most common of system failures are found in figure 5.2.

5.2 Real Application Clusters

Oracle Real Application Clusters (RAC) enables multiple instances to share access to an Oracle database. The result is a single database system that spans multiple hardware systems providing HA and redundancy in case of failures in the cluster [17].

Each server (also referred to as a node) in the cluster will have its own Oracle in-

stance running. Figure 5.3 show a RAC database with two servers accessing the same database.

Allowing multiple servers to access the same database will allow users to continue their work with the database even if a server fails. The cluster, as a whole, will therefore be more fault tolerant preventing a single server from bringing the whole system down. This will reduce the downtime of the system.

Essentially, in a RAC configuration there will be at least 2 different servers, and a maximum of 100, each with its own memory, background processes, and local disks. Each instance has its own redo logs, but the control files and datafiles are shared between the instances [2, 4, 9, 10].

ACT is deployed in a database system where 6 serves are accessing one database. Beside gaining reliability and reducing downtime, an opportunity for load balancing arises with more servers.

The following subsections provide a short description of the main advantages obtained by using RAC. All these features contribute to improve the availability and reliability of the database system.

5.2.1 Failover

Failover in RAC is the ability of a surviving node in the cluster to assume the responsibility of a failed node. Although failover doesn't directly address the issue of reliability regarding the underlying hardware, automatic failover can reduce the downtime from hardware failure [1].

With simple hardware failover (when RAC is not deployed) there is only one active instance and the database is completely unavailable until node failover, instance startup, and crash recovery are completed. This can result in relatively long periods of time where the database is not reachable for the users [1].

With RAC, only the users connected to the instance which fails will be affected. The other users will not notice this failure. Oracle RAC systems provide two methods of failover, depending on when a failure occurs:

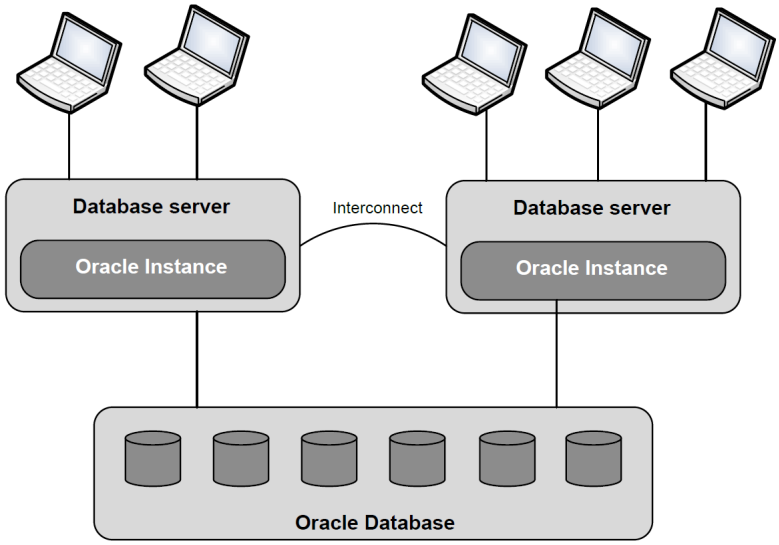

Figure 5.3: Oracle RAC database with two instances accessing the database [1].

- If a failure occurs at connection time, the application can be failed over to another active node in the cluster. This is called connection failover.
- If a failure occur after a connection is established, the connection is moved to another active Oracle RAC node in the cluster by using Transparent Application Failover (TAF). This happens without the user application having to re-establish the connection [9].

5.2.2 Instance Recovery

Instance recovery occurs after a database instance fails and the content of the System Global Area (SGA) is lost [4]. RAC instances provide protection for each other - if an instance fails, one of the surviving instances will detect the failure and automatically initiate RAC recovery [1].

During normal operation, Oracle writes database changes to the current online redo

log file when a transaction is committed. The database buffer cache does however not need to be flushed for each commit. This will result in a situation when the online redo log file contains changes that have not yet reflected in the database datafiles. When an Oracle instance is restarted after a failure, it will detect this failure through information in the control file and in the headers of the database files. *Instance recovery* closes the cap that might exists between the datafiles and the redo log files by replaying all changes made from the last completed checkpoint to the point of instance failure [1].

5.2.3 Load Balancing

RAC is a solution from Oracle where all instances are active and users are spread between the different instances [1]. To balance the load among these instances, Oracle RAC provides two solutions:

- Server load balancing distributes processing workload among Oracle RAC nodes [9].
- Client load balancing distributes new connection among Oracle RAC nodes such that no server is overwhelmed with connection requests [9].

The database system where the ACT schema is deployed use a combination of these load balancing solutions to benefit from the advantages they provide.

5.3 Recovery Manager

Recovery Manager (RMAN) is an Oracle utility for managing database backup and recovery of the database. RMAN determines the most efficient method of executing the requested backup, restoration, or recovery operation and then submits these operations for processing [17].

The log of RMAN activities is known as the *RMAN repository*. The repository is updated each time RMAN performs a backup operation, and the log information can be stored in the control file or in an independent recovery catalog. The independent

recovery catalog is a schema which stores RMAN repository information in a separate recovery catalog database [7, 8, 11].

RMAN is utilized by the CMS online database. Through RMAN they benefit from a range of techniques and features for backup and recovery that's not available with user-managed backup and recovery.

The next sections will give a short explanation of how RMAN works with Data Guard, the control file, and the repository catalog. There will also be given a short description of some of the backup and recovery alternatives that can be deployed with RMAN (note that some of these alternatives might also be available without RMAN). The solutions used for the CMS online database will be highlighted through these sections.

5.3.1 RMAN and Data Guard

RMAN and Data Guard (see 5.4 on page 52) are complementary technologies that together make a complete Oracle solution for recovery and high availability.

RMAN is used to create the Data Guard standby database. After the standby database has been created, RMAN can connect to the standby database and take backups [2].

5.3.2 RMAN with Control File

When the control file is used as the repository, RMAN use the information contained in the control file to decide what needs to be backed up. When the backups is done, RMAN writes a record of that backup to the control file, along with checkpoint information [2].

When RMAN starts a backup operation it needs to know the most recent checkpoint information and the file schematic at the time the backup begins. RMAN also needs the information in the control file to stay consistent for the duration of the backup operation. At the same time should the control file also be available for usage by the Relational Database Management System (RDBMS).

With frequent updates of the database, there is no chance that RMAN can have a con-

sistent view of the control file without locking the file. Locking the control file will hamper normal operation of the database and is not a good solution. To get around this, RMAN uses the *snapshot control file*, an exact copy of the control file that's only used by RMAN. Before a backup operation starts, RMAN refreshes the snapshot control file from the actual control file [2].

The records of RMAN backups belong to the category of circular reuse records in the control file. This means that the records will get aged out if the control file. This can be catastrophic to a recovery situation: without the record of the backups in the control file, it is as though the backups never took place. There is however an initial parameter that specifies how old a backup record should get. Instead of deleting records before time, the control file will expand to meet the required age of the records [2].

5.3.3 RMAN with Recovery Catalog

Control files are the default option for implementing a repository. Using control files, however, limits the functionality of RMAN, while a recovery catalog provides the use of all RMAN features.

The repository is a separate database, or schema, that contains metadata about the target database[1] and all backup and recovery information needed for database maintenance purposes. The information stored in the recovery catalog will also provide the ability to restore files and recover the database [2, 4, 5].

The database where the ACT schema is deployed use the recovery catalog solution. By using the recovery catalog database, they benefit from all features available in RMAN, and the risk of records getting aged out to soon is removed. One disadvantage of using an RMAN recovery catalog is that it needs to be managed itself. However, this is a small database or schema, so maintenance is minimal [2].

The server where the catalog resides at CERN is (and should be) separate from the primary and standby sites so in the event of disaster at either the standby or primary site, the ability to recover from the last backups will not be impacted.

[1] The target database is the database which RMAN is going to perform backup, restore, and recovery operations on [2].

5.3.4 Online and Offline Backups

RMAN supports both online (hot) and offline (cold) backups [2].

5.3.4.1 Offline Backups

Offline (cold) backups are the simplest type of backups. These backups are taken when the database is shut down, which also means it's not available for users during this backup operation.

Offline backups are the only option if the database is in noarchivelog mode. If there are no archive logs, only a *complete database recovery* can be performed, where the database is essentially restored as of the time of the backup. All work done since the time of the last backup is lost and a complete recovery must be performed even if just a single datafile is damaged [2].

Offline backups of a database in archivelog mode will give the ability to recover the database to the time of failure [5].

5.3.4.2 Online Backups

The ACT schema is deployed in a database where online backups are taken.

With online (hot) backups the database can continue its operation while the backup is being performed. This means that end users can continue their normal operations against the database while the backup is going on.

Online backups are only possible if the database is running in archivelog mode. In this mode it's possible to restore the damaged datafile(s) without necessarily restoring all datafiles. It should be noted that when recovering from a backup it's important that all files making up the database must be recovered to the same point in time, in order to keep the entire database consistent [2, 5].

5.3.5 Physical or Logical Backups

Backups made with RMAN can be divided into physical and logical backups.

Physical backups are performed on the entire database without regard for the underlying logical data structure. Physical backups are the foundation of any sound backup and recovery strategy, and is the solution deployed in the CMS online network [5, 7].

Logical backups are a useful supplement to physical backups in many circumstances but they do not provide sufficient protection against data loss without physical backups [7]. With logical backups one can select what logical database structures should be backed up (e.g. tables and indexes). A database can be restored in a more granular fashion with logical backups compared to what's possible with a physical backup. A major drawback with logical backups is that they can only be used for restore, and it's not possible to use it for recovery [5].

5.3.5.1 Consistent or Inconsistent Physical Backups

Physical backups can be divided into consistent and inconsistent backups.

Consistent backups are those created when the database is in a consistent state, that is, when all changes in the redo log have been applied to the datafiles. A database restored from a consistent backup can be opened immediately, without undergoing media recovery. A consistent backup can only be created after a consistent shutdown [7].

For inconsistent backups, a backup is taken while the database is open. However, when a database is restored from an inconsistent backup, it must undergo media recovery in order to apply any pending changes from the online and archive redo log before it can be opened again. Using inconsistent backups requires that the database is in archivelog mode [7].

The CMS online database use inconsistent backups. By using this solution there is no need to shut down the database before making a backup, and thereby reducing the downtime of the database. During a restore, however, the downtime will be a bit longer compared to consistent backups, since redo have to be applied to the database before it

can be opened again. Backup operations is, usually, more frequent then restore/recover activities, so the accumulated downtime will be less with inconsistent backups.

5.3.6 Incremental and Full Backup

When performing a full backup, the entire database is backed up [1].

With incremental backups, RMAN will back up only database blocks that have changed since the last backup. RMAN decides what information to backup through the use of its repository information. Incremental backups are smaller and faster than full backups; they require less disk usage and use less network bandwidth [1]. A benefit of incremental backups is the reduction in overall backup times, the disadvantage appear on the recovery side. Because Oracle will need to use several backup sets to recover the database, the time required to recover the database can significantly increase [2].

Both these backup options are used in the CMS online network depending on what part of the database is being backed up.

Figure 5.4 gives an illustration of how data backups are utilized. In the example there is a full backup at System Change Number (SCN) 100. Redo logs generated during the operation of the database capture all changes that occur between SCN 100 and SCN 500. Along the way, some logs fill and are archived. At SCN 500, the datafiles of the database are lost due to media failure. The database is then returned to its state at SCN 500 by restoring the datafiles from the backup taken at SCN 100 and then applying transactions captured in the archived and online redo logs [7].

5.3.7 Complete and Incomplete Recovery

Recovery can be performed on any database that's in archivelog mode, whether the backup were online or offline. There are two fundamental types of recovery:

Complete recovery is recovering a database to the most recent point in time, without the loss of any committed transactions. Generally, the term recovery refers to complete recovery [5, 7].

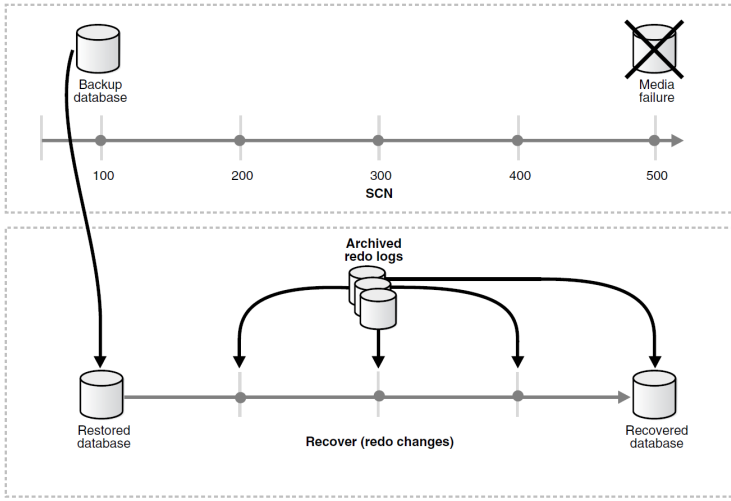

Figure 5.4: Basic example of how backing up, restoring and recovering a database is performed using redo and SCN [7].

Incomplete, or point-in-time, recovery is when the database is restored and then optionally rolled forward to a predetermined point in time by applying some, but not all, of the logs. This produces a version of the database that is not current and is often done to bring the database back to a time before the problem occurred [5, 7].

5.4 Data Guard

Data Guard is used to ensure high availability, data protection, and disaster recovery for enterprise data [17].

The primary goals of Data Guard are:

- The standby database should be a synchronized copy of the primary database [4].
- Provide a high degree of isolation between primary and standby databases. This prevents problems that occur at the primary database from impacting the standby

database [4].

- Provide data availability [4].

5.4.1 Primary Database

A Data Guard configuration contains one production database, also referred to as the primary database. The primary database can be either a single-instance Oracle database or an Oracle RAC database [12].

The primary database must be in archivelog mode to be able to use Data Guard. The primary and standby databases must also have their own control files for Data Guard to work [12].

5.4.2 Standby Database

A standby database is a transactional consistent copy of an Oracle production database which is initially created from a backup copy of the primary database. A Data Guard configuration can contain up to nine standby databases.

Once created, Data Guard automatically maintains the standby database by transmitting primary database redo data to the standby system and then applying the redo to the standby database. As with the primary database, a standby database can be either a single-instance Oracle database or an Oracle RAC database [1, 12].

A standby database provides a safeguard against data corruption and user errors. Storage level physical corruptions on the primary database do not propagate to the standby database. Redo data is also validated when it's applied to the standby database [12].

If the connection is lost between the primary and one or more standby databases, redo being generated in the primary cannot be sent to those standby databases. Once connectivity is re-established, the missing log sequence (or the gap) is automatically detected by Data guard, and the necessary redo logs are automatically transmitted to the standby database [4, 12].

A standby database can be either physical or logical [4]. The standby database used for the production database in the CMS online network is a physical standby database. Both the primary and standby database in the CMS online network is RAC databases.

5.4.2.1 Physical Standby Database

The disk structure in a physical standby database is identical to the primary database on a block-by-block basis. A physical standby database is kept synchronized with the primary database through Redo Apply (see 5.4.6.1 on page 59) [12, 17].

A physical standby database enables a robust and efficient disaster recovery solution. Switchover and failover capabilities allow an easy role reversal between primary and physical standby databases, minimizing the downtime of the primary database [12].

5.4.2.2 Logical Standby Database

A logical standby database is initially created as an identical copy of the primary database can later be altered to have a different physical organization and structure of the data. In a logical database the data, not the database itself, is redundant.

The logical standby database is kept synchronized with the primary database through SQL Apply (see 5.4.6.2 on page 60) [17].

An advantage of logical standby databases is their ability to host additional database schemas beyond the ones that are protected in the Data Guard configuration. In addition are users at any time allowed to access the standby database for queries and reporting purposes. A logical stanby database can therefore be used for other business purposes then just disaster recovery [1, 12]. This is a feature of no interest for the CMS online network. Their standby database shouldn't be used for anything other than being a standby database.

Restrictions on datatypes, types of tables, types of Data Definition Language (DDL) operations, and Data Manipulation Language (DML) operations are also reasons why the logical standby database solution is not deployed in CMS online [12].

5.4.3 Redo Transport Service

Data Guard Redo Transport Service coordinates the transmission of redo from a primary database to the standby database. Figure 5.5 gives an overview of how this process.

When the LGWR process in the primary database is writing redo to its online redo log, a separate Data Guard process, called the Log Network Server (LNS), reads from the redo log buffer and passes this redo information to Oracle Net Services for transmission to the standby database [4]. Redo records transmitted by the LNS are received at the standby database by another Data Guard process called the Remote File Server (RFS).

Data Guard can, through the LNS process, transport redo data synchronously or asynchronously:

- Synchronous transport (SYNC) doesn't allow the LGWR process to acknowledge a commit as successful, and proceed with the next transaction, until the LNS process can confirm that the redo has been written to at least one standby database in the configuration. This guarantee will generate interdependencies between the primary and standby database and can impact the performance of the primary database [1, 4].
- Asynchronous transport (ASYNC) is different from SYNC transport since it does not require that the LGWR process is waiting for acknowledgment from the LNS process. This creates a near zero performance impact on the primary database. The behavior of ASYNC transport enables the primary database to buffer a large amount of redo called transport lag. A drawback of ASYNC is the increased potential for data loss. If a failure destroys the primary database before any transport lag is reduced to zero, any committed transactions that are a part of the transport lag will be lost [1, 4].

5.4.4 Data Guard Protection Modes

Some businesses can't tolerate data loss and are willing to sacrifice performance for data protection. Other businesses demand high performance and will allow potential loss of data to achieve this.

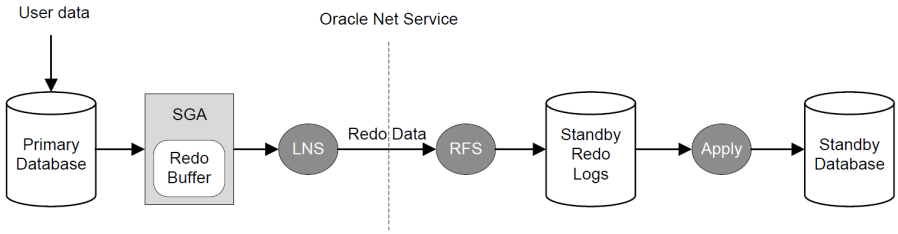

Figure 5.5: Illustration of how the redo transport service is sending redo information between the primary database and the standby database [4].

Data Guard protection modes implement rules that govern how the configuration will respond to failures [4, 12]. The different protection modes will be described next.

5.4.4.1 Maximum Protection

Maximum protection offers the highest level of data protection. It requires SYNC redo transport which guarantees no data loss [12]. Most implementations of maximum protection mode have a minimum of two standby databases at different locations, so that failure of an individual standby database doesn't impact the availability of the primary database [4].

5.4.4.2 Maximum Availability

Maximum availability mode is similar to maximum protection mode. It also requires SYNC redo transport, but will wait for a predefined period of time before giving up the standby destination and allowing primary database processing to proceed. This prevents a failure in communication between the primary and standby database from impacting the availability of the primary database. Data Guard will remove the transport lag once the primary database is able to re-establish a connection to the standby database [4, 12].

5.4.4.3 Maximum Performance

This mode emphasizes primary database performance over data protection and therefore use ASYNC redo transport. A potential data loss can occur if the transport lag is not zero at the time of failure [4, 12].

Despite the fact that data contained in the transport lag might be lost in case of a failure, this is the solution deployed in the CMS online network. RAID-1 (see 5.6.1 on page 61) is used at the primary, and standby, databases. By using RAID-1, data is redundant and a media failure at the primary database can be recovered. The chance of losing data due to disk failure is therefore reduced.

5.4.5 Role Management Services

An Oracle database operates in one of two roles: primary or standby. Using Data Guard, one can change the role of a database using either a switchover or a failover operation [12].

5.4.5.1 Switchover

A switchover is a planned role reversal between the primary database and one of its standby databases. This role reversal is planned in all cases and there will be no data loss. The primary database must complete all redo generation on the production data before allowing the switchover to commence [4].

A switchover is particularly useful for minimizing downtime during planned maintenance, the only production database downtime is the time required to execute the switchover [12].

5.4.5.2 Failover

A failover is an unplanned, and irreversible, role transition of a standby database to the primary role. This is only done in the event of a failure of the primary database. Al-

though the process is similar to switchover, the primary database never has the chance to write an end-of-redo record. From the perspective of the standby database, redo transport has suddenly gone dormant. At this point, whether or not a failover results in data loss depends upon the Data Guard protection mode (see 5.4.4 on page 55) in effect at the time of failure [4, 12].

A Database Administrator (DBA) has the choice of configuring manual or automatic failover. In either manual or automatic case, executing a database failover is very fast once the decision to perform the failover has been made [4].

- Manual failover, which is the solution used in the CMS online network, give the administrator complete control of role transitions. Manual failover will lengthen the outage and increase the chances of errors due to the human element involved.

- With more aggressive recovery time requirements, more can be gained from implementing automatic failover. Data Guard's Fast-Start Failover (FSFO) automatically detects the failure, evaluates the status of the Data Guard configuration, and, if appropriate, execute the failover to a previously chosen standby database. FSFO ensures that the failover occurs only when everything meets the rules specified. The failed primary is also not allowed to open after a failover in order to avoid any chance of the split-brain scenario[2]. Each Data Guard FSFO configuration involves one Observer[3] . If that Observer is not running or is not reachable, the automatic failover ceases to be automatic [4].

The requirements for the database where the ACT schema is deployed are currently lax enough to be achieved using manual failover. By using manual failover there is a risk of introducing errors due to human interaction. The persons responsible for this database system have long experience, but there is still a chance of making mistakes. The main benefit by using manual failover is the increase in control of when, and what, is happening.

[2] A split-brain is two independent database each operating as the same primary database. This can happen if someone manages to restarts the original primary database after a failover to the standby database has been performed [4].

[3] The observer is a separate computer which job is to maintain a connection with the primary and the target standby, monitor the health of the configuration and perform the failover when required [4].

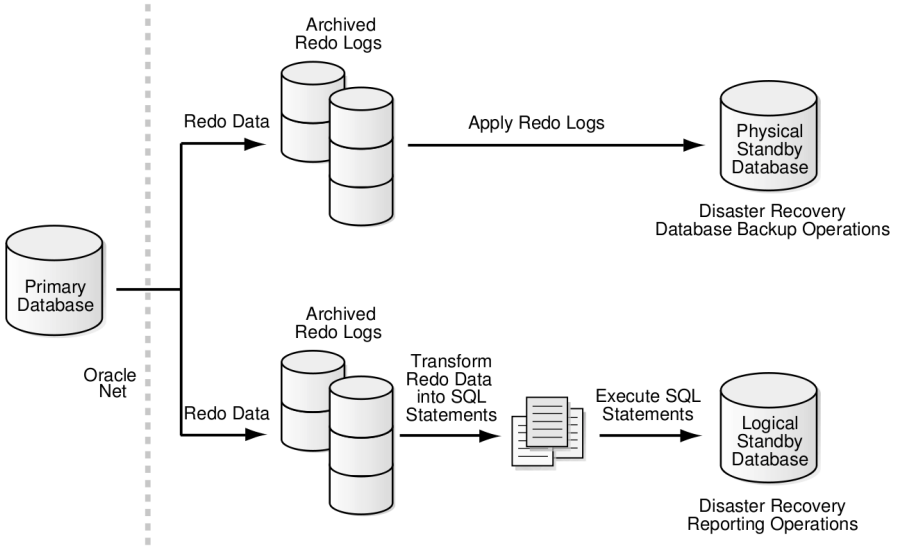

Figure 5.6: Ways to apply data to a standby database in a Data Guard configuration [63].

5.4.6 Redo Apply and SQL Apply

When redo data is transmitted from the primary database to the standby database, it's put into the archived redo logs of the standby system. This can be done in two ways; redo apply for physical standby databases and SQL apply for logical standby databases [12]. Figure 5.6 illustrates how these works.

5.4.6.1 Redo Apply

Redo apply maintains a physical standby database (like the one in the CMS online network). With redo apply, redo data is applied to the standby database using standard recovering techniques on the Oracle database server. In addition is Oracle processes used to validate the redo before it is applied to the standby database [4, 12].

5.4.6.2 SQL Apply

SQL Apply build SQL statements from redo data received from the primary database. These SQL statements are then applied to the standby database. SQL apply, like redo apply, prevent any modifications from being made to the data it is replicating [4, 12].

Disadvantages with this solution is that SQL Apply doesn't support all data types and it requires extra processing compared to redo apply [4, 12].

5.5 Oracle at CERN

To summarize the Oracle database features deployed in the CMS online network:

- RAC creates a redundancy of servers by allowing many servers to interact with the same database. Users are distributed among the servers, and if a server fails, users will automatically be reconnected to another server and the failure will in many cases not be noticeable. RAC also speeds up the instance recovery process.
- All RMAN features is available. Online backups are used, allowing backups to be taken while the database is running. The backups taken are inconsistent, physically, backups.
- Data Guard is used to operate and manage the standby database. The standby database used for the production database is a physical standby database.

Maximum performance is the Data Guard protection mode used in the CMS online network. The maximum performance can result in data loss if the transport lag contains data when a failure occurs. Since RAID-1 is used at the primary and standby databases, data is redundant and the chances of losing data contained in the transport lag is reduced.

Physically, inconsistent, backups will reduce the downtime experienced by users during a backup operation. It will however increase the duration of a restore operation. The frequency of backup operations are however much higher than restore operations in the

CMS online network. The accumulated downtime is therefore, in the long run, lower with this solution.

Manual failover is used in Data Guard to achieve more control of what's happening in the case of a failover. The disadvantage of this is the increase in outage time due to human interaction. Despite all the effort in reducing the downtime (RAC load balancing, automatic instance recovery, automatic failover of users, online and inconsistent backups), this is pointing in the opposite direction. If performance is important, and the downtime should be as low as possible, CERN should in the future look into migration to automatic failover, despite the complexity introduced.

The number of servers in the RAC system is abundant. Only two of six servers are active during normal operation. The remaining servers can be activated to off-load the active servers, or to perform repairs without affecting the availability of the system. The same effect could most likely be achieved with 4 servers. No operational statistics were available (for publication) to show the actual usage load or downtime of the server and database system.

5.6 Component Redundancy

Disks have the shortest Mean Time to Failure (MTF) of any of the components in a computer system and are the largest area of hardware failure. Protection from disk failure is usually accomplished using RAID technology [1].

5.6.1 RAID

RAID basically takes multiple hard drives and allows them to be used as one large hard drive. Depending on what RAID level is being used, grouping disks into arrays can result in data redundancy and/or increased performance. Spreading I/O operations across multiple spindles will also reduce the contention on individual drives [1, 16].

Some basic terms used in RAID are:

Mirroring takes a copy of the data and puts it on a separate hard drive. When a hard drive fails, the system can continue to operate due to the redundant data. Mirroring also result in low downtime and data recovery is relatively simple. Mirroring ties up both drives during the write process, which can reduce performance. On the other hand, read operations can be distributed to the two drives, which increases performance. Mirroring is a good solution to ensure the safety of data, the main trade off with mirroring is the cost and wasted space involved with having two copies of the same data.

Parity makes data redundant by the use of parity data. Parity data is calculated using the XOR operation on the data stored on the disks. By storing this parity data there's no need to have two copies of the same data. The amount of fault tolerance with parity data is however not as high as in mirroring. Generation the parity data also requires some computing power, and the parity data has to be calculated each time a write operations take place. In addition is recovering from a failure more complicated with parity compared to mirroring.

Striping improves the performance of the array by breaking the data into pieces and distributing these pieces across all the drives. The time it takes to read data from disk, using striping, equals the time it takes to read these small pieces of data from the disks (in parallel). This is n times faster than reading the whole file from one disk, where n is the number of disks in the striping array. The same is true for writing files to disk. The more hard drives there are, the more the performance increase. The data is however not redundant with striping.

The standard RAID configurations are specified as RAID-0 through RAID-5. All of the configurations, except RAID-0, provide some form of redundancy. RAID-0 is the simplest level of RAID, and it just involves striping. The reliability in RAID-0 is actually getting worse compared to just having a single disk. This is one of the main reasons why RAID-0 is not being used in the CMS online network [14, 16].

RAID-1, which is based on mirroring, is the RAID configuration used in the CMS online network. The RAID-1 configuration in the CMS online network involve 120 disks. Rebuilding a lost drive is very simple with RAID-1 since there is still a copy of the data after a failure. There are however few performance benefits with RAID-1. The write performance is the same as with a single drive, but the read performance will be doubled [13, 15, 16]. Figure 5.7 gives a illustration of RAID-1.

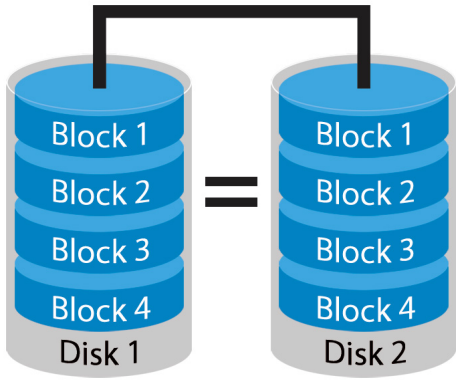

Figure 5.7: Illustration of RAID-1, data mirroring [21].

5.6.2 Network Redundancy

As mentioned in the section about RAC, the database system where the ACT schema is deployed has 6 servers accessing the same database. The servers are connected through fiber channels to the production database. The topology of the network is a star connection, and to make this topology more faults tolerant, each connection is duplicated, including the switches. The topology is illustrated in figure 5.8.

5.6.3 Other Redundancy Measures at CERN

In addition to the disk redundancy and the network redundancy, the CMS online database has redundant power supply. There is also a climate control system to provide a stable temperature where the database system is deployed.

Backups of the CMS online database are taken on a regular basis. The standby database is physically separate from the primary database, such that a complete site failure shouldn't bring the whole system down. Both the primary and standby database in the CMS online network is RAC databases.

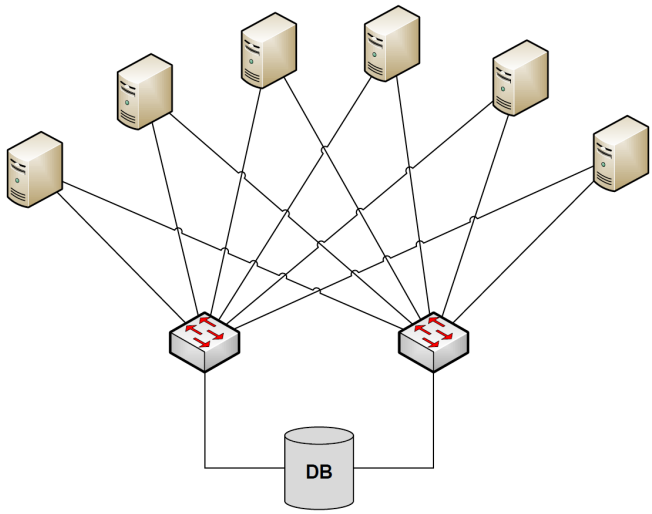

Figure 5.8: Illustration of the network topology for the database system containing the ACT schema.

Chapter 6

Discussion

The ACT schema is, at the time of writing, fully functional and deployed in the CMS online database. It encompasses all the needed aspects of the planning and coordination process, and its structure is flexible for extensions and changes in the future. All work done with the schema has turned it into something better, which also has triggered the interest of a huge part of the CMS community at CERN.

The scope of the ACT schema grew fast and it soon became bigger than what was initially expected due to new ideas, procedures and functionality arising through the development process. All these changes were a good test for the flexibility which the schema should achieve through normalization.

It was early decided that information already contained in existing CERN databases should be reused and not duplicated. A duplication of data could easily create synchronization and maintenance problems of the data. The fact that a database outside the CMS online network cannot access data contained in a database inside the network made this goal hard to obtain. The EMDb database, which contains the most relevant information for the ACT, is located on the outside of the network while the ACT database is located on the inside. The result was a duplication of some EMDb data into the ACT schema, which is not ideal.

Despite some performance and consistency issues, which can be improved in the schema

itself, reliability, availability and performance experienced by the user mainly stem from the database system. The Oracle database system used in the CMS online network improves both performance and data redundancy. During the work with the ACT schema, a deeper look at the database system was performed. One thing which was noticed during the study was the fact that manual failover is used. This was not expected since it will increase the downtime and also increase the chances of (human made) errors.

The redundancy found in the CMS online database system was however impeccable. The data itself is redundant, the network connections and switches are redundant, the power supply is redundant, the servers are redundant, and the database itself is redundant. Of the six servers accessing the database, only two servers are active at normal operation. The remaining servers are just waiting for an error or a maintenance activity. This makes the number of redundant servers seem a bit excessive.

Chapter 7

Conclusion

A database schema designed for planning and coordination of work package activities at CMS has been produced. All aspects not covered by other CERN databases have been implemented in the schema where flexibility for extensions and modifications in the future has been a focus throughout the development.

The Oracle database where the schema is deployed provides both data redundancy and increased performance. Components redundancy in the database system itself also increases the reliability of the database system.

The network containing the database has strict security rules which limit the amount of potential users. An easy and secure way of dynamically adding temporary users should be looked into. A closer look should also be taken at the challenges which will arise when eventually merging the schema found in EMDb and the one developed for ACT. Extensions should also be made to the schema in order to integrate experiments like ATLAS, ALICE, LHCb and LHC.

Bibliography

[1] Greenwald, R., et al., *Oracle Essentials - Oracle Database 11g*, 4th Edition, ISBN-13: 978-0-596-51454-9, pp. 24-27, 33-54, 133-135, 163-164, 217-218, 253-289.

[2] Hart, F., et al., *Oracle Database 10g - RMAN Backup & Recovery*, 2007, ISBN-13: 978-0-07-226317-6, pp. 6, 8-11, 14-19, 24-26, 42-45, 250-251, 498-499, 513.

[3] Karam S., et al., *Easy Oracle Jumpstart - Oracle Database Management Concepts and Administration*, 2006, ISBN-13: 978-0-9759135-5-0, pp. 16-22, 27-37, 178.

[4] Carpenter, L., et al., *Oracle Data Guard 11g - Handbook*, 2009, ISBN-13: 978-0-07-162111-3, pp. 2-24, 108-109, 115, 172-173, 230, 300-311, 335-337, 378-379, 382, 412, 474.

[5] Abramson, I., et al., *A Beginner's Guide - Learn Oracle Database Essentials*, ISBN: 0-07-223078-9, pp.94-95, 127-129, 158-168, 185-187, 192-193.

[6] Ashdown, L., et al., *Oracle Database Concepts*, http://download.oracle.com/docs/cd/E11882_01/server.112/e10713.pdf , cited 13.06.2010, pp. 15-36.

[7] Romero, A., et al., *Oracle Database - Backup and Recovery Basics*, 2005, http://youngcow.net/doc/oracle10g/backup.102/b14192.pdf , cited 13.06.2010, pp. 15-30.

[8] Ashdown L., et al., *Oracle Database - Backup and Recovery User's Guide*, 2007, http://www.comp.dit.ie/btierney/oracle11gdoc/backup.111/b28270.pdf , cited 20.06.2010, pp. 29-37.

[9] *Using Oracle Real Application Clusters (RAC) - DataDirect Connect for ODBC*, 2004, http://www.datadirect.com/developer/jdbc/docs/jdbc_oracle_rac.pdf , cited 21.06.2010, pp. 1-10.

[10] Bauer, M., et al., *Oracle Database, Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide*, 2007, http://www.comp.dit.ie/btierney/oracle11gdoc/rac.111/b28254.pdf , cited 21.06.2010, pp. 21-28.

[11] Kuhn, D., et al., *Getting Started with RMAN*, 2002, http://www.darlkuhn.com/dba/ppt/kuhn.pdf , cited 22.06.2010, pp. 1-6.

[12] Rich, K., *Oracle Data Guard - Concepts and Administration*, 2007, http://www.comp.dit.ie/btierney/oracle11gdoc/server.111/b28294.pdf , cited 25.06.2010, pp. 27-44.

[13] *Overview of Redundant Arrays of Inexpensive Disks (RAID)*, http://support.microsoft.com/kb/100110 , cited 04.07.2010.

[14] Wong, W., *An Introduction to RAID*, http://electronicdesign.com/content/print.aspx?topic=an-introduction-to-raid4947 , cited 04.07.2010.

[15] Hendrics, G., *An Introduuction to RAID*, http://ezinearticles.com/?An-Introduction-to-RAID&id=137314 , cited 04.07.2010.

[16] Solinap, T., *RAID: An In-Depth Guide to RAID Technology*, http://www.midwestdatarecovery.com/understanding-raid-technology.html , cited 04.07.2010.

[17] To, L., et al., *Oracle Database High Availability Overview*, 2009, http://download.oracle.com/docs/cd/B28359_01/server.111/b28281.pdf , cited 06.07.2010, pp. 11-24, 30-31, 37-39, 73-86, 91.

[18] http://onlineappsdba.com/wp-content/uploads/2008/01/db_block_extent_segment.jpg , cited 30.06.2010.

[19] http://merc.tv/img/fig/oraphylog.png , cited 01.07.2010.

[20] http://www.ucertify.com/article/articleImages/i3701342c.gif , cited 28.06.2010.

[21] http://www.soundonsound.com/sos/oct07/images/DataProtection_02_RAID1.jpg , cited 04.07.2010.

[22] http://public.web.cern.ch/public/en/About/About-en.html , cited 06.04.2010.

[23] http://cms.web.cern.ch/cms/index.html , cited 06.04.2010.

[24] http://wofford-ecs.org/DataAndVisualization/ermodel/material.htm , cited 13.07.2010.

[25] http://www.databasedesign.co.uk/bookdatabasesafirstcourse/chap3/ chap3.htm , cited 13.07.2010.

[26] http://folkworm.ceri.memphis.edu/ew/SCHEMA_DOC/comparison/ erd.htm , cited 14.07.2010.

[27] http://www.smartdraw.com/resources/tutorials/entity-relationship-diagrams/ , cited 17.07.2010.

[28] http://msdn.microsoft.com/en-us/library/ms175464.aspx , cited 17.07.2010.

[29] http://dev.mysql.com/tech-resources/articles/intro-to-normalization.html , cited 20.07.2010.

[30] http://databases.about.com/od/specificproducts/a/normalization.htm , cited 20.07.2010.

[31] http://support.microsoft.com/kb/100139 , cited 21.07.2010.

[32] http://www.devshed.com/c/a/MySQL/An-Introduction-to-Database-Normalization , cited 21.07.2010.

[33] http://tomandmaria.com/Tom/Teaching/Drexel210/translation_ hints.htm , cited 22.07.2010.

[34] http://download.oracle.com/docs/cd/E17409_01/javase/tutorial/jdbc/overview/database.html , cited 22.07.2010.

[35] http://cplus.about.com/od/introductiontoprogramming/p/database.htm , 24.07.2010.

[36] http://searchsqlserver.techtarget.com/definition/relational-database , cited 24.07.2010.

[37] http://computer.howstuffworks.com/question599.htm , cited 24.07.2010.

[38] http://computer.howstuffworks.com/framed.htm?parent=relational-database.htm&url=http://www.edm2.com/0612/msql7.html , cited 25.07.2010.

[39] http://ltu164.ltu.edu/mmaa/doc/erd.htm , cited 25.07.2010.

[40] http://blogs.nature.com/news/thegreatbeyond/2008/06/ , cited 09.08.2010.

[41] http://www.w3schools.com/sql/sql_intro.asp , cited 14.09.2010.

[42] http://www.kb.iu.edu/data/ahux.html , cited 14.09.2010.

[43] http://www.sql-server-performance.com/articles/dev/views_in_sql_ server_p1.aspx , cited 14.09.2010.

[44] http://www.mysqltutorial.org/introduction-sql-views.aspx , cited 14.09.2010.

[45] http://download-west.oracle.com/docs/cd/A87860_01/doc/server.817/ a76965/c14sqlpl.htm#5943 , cited 14.09.2010.

[46] http://download-west.oracle.com/docs/cd/A87860_01/doc/appdev.817/ a76939/adg06idx.htm#11977 , cited 13.09.2010.

[47] http://download-west.oracle.com/docs/cd/A87860_01/doc/appdev.817/ a76939/adg07iot.htm#2031 , cited 13.09.2010.

[48] Toptsis, A., *B**-tree: A Data Organization Method for High Storage Utilization*, 1993, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=315364 , cited 15.09.2010, pp. 1.

[49] http://www.w3schools.com/Sql/sql_join.asp , cited 15.09.2010.

[50] Meier, J.D., et al., *Chapter 14 - Improving SQL Server Performance*, 2004, http://msdn.microsoft.com/en-us/library/ff647793.aspx , cited 15.09.2010.

[51] McGehee, B., *SQL Server Performance Tuning for SQL Server Developers*, 2000, http://www.databasejournal.com/features/mssql/article.php/1466951/ SQL-Server-Performance-Tuning-for-SQL-Server-Developers.htm , cited 15.09.2010.

[52] McGehee, B., *Performance Tuning SQL Server Joins*, 2006, http://www.sql-server-performance.com/tips/tuning_joins_p1.aspx , cited 15.09.2010.

[53] http://www.w3schools.com/sql/default.asp , cited 16.09.2010.

[54] Baylis, R., et al., *Oracle Database Administrator's Guide, 10g Release 1*, 2003, http://www.stanford.edu/dept/itss/docs/oracle/10g/server.101/ b10739.pdf, cited 17.09.2010, pp. 898-908.

[55] http://www.stanford.edu/dept/itss/docs/oracle/10g/server.101/b10739/ds_concepts.htm#i1007669 , cited. 17.09.2010.

[56] http://i.msdn.microsoft.com/ms345146.sql2k5partition_01%28en-US,SQL.90%29.gif , cited 17.09.2010.

[57] http://public.web.cern.ch/public/en/lhc/LHCExperiments-en.html , cited 23.09.2010.

[58] http://en-dep.web.cern.ch/en-dep/Groups/ , cited 23.09.2010.

[59] http://gs-dep.web.cern.ch/gs-dep/structure/ , cited 24.09.2010.

[60] http://it-cs.web.cern.ch/it-cs/ , cited 24.09.2010.

[61] http://te-dep.web.cern.ch/te-dep/structure/VSC/ , cited 24.09.2010.

[62] https://espace.cern.ch/be-dep/ABP/default.aspx , cited 24.09.2010.

[63] Oracle Corporation, *Oracle Data Guard Concepts and Administration*, 2002, http://download.oracle.com/docs/cd/B10501_01/server.920/ a96653.pdf , cited 26.09.2010, pp. 45.

Appendix A

Privileges of Roles and Functions

Roles and functions are used to distinguish between users of the web application, and to determine what privileges they might have. This chapter provides a short explanation of the different roles and functions which are relevant for ACT. At the end of this chapter there is also a graphical summary of the distribution of privileges.

A.1 Roles

A role is a person's employment status at CERN. A person's role will not change from one WP to another.

A.1.1 TC and EAM

The Technical Coordinator (TC) and EAM will have the same privileges in ACT. They are responsible for coordinating all work activities and will therefore need the ability

to read, modify and approve information contained in *all* WPs. TC will, in most cases, only be involved if some extensive work is being planned. EAM will be the main person when it comes to the planning and coordination of WPs. They will have all the privileges in the application:

- Fill-in and modify all work package requests that's received.
- Broadcast information regarding all WPs.
- Browse, retrieve and modify information from the WP database.
- Approve or reject all received WP requests.
- Arrange meetings for the different WPs, and distribute invitations for these meetings.
- Upload documents/pictures/comments for all WPs.
- Close a WP, meaning that a WP is flagged as finished.
- Manage access requests (if a WP needs more time to finish, TC or EAM can grant more time which will result in an expanded access period).

A.1.2 Safety Coordinator

The safety coordinator performs the safety inspection that's required before the start of every scheduled WP. The safety coordinator will during the safety inspection check if the safety recommendations made during the risk assessment meeting have been implemented correctly. The safety coordinator is taking part of the risk assessment meeting and act as an safety consultant provided by CERN. A safety coordinator has the following privileges:

- Browse and retrieve information from the WP database.
- Arrange VIC meetings.
- Postpone or cancel WPs.
- Upload documents/pictures/comments regarding the WPs he's involved in.

- The ability to read WP information (schedules, safety recommendations, CCC alarms, etc.) that's getting distributed.

A.1.3 Transport Coordinator

If a WP involves transport with one of the bigger cranes, a transport coordinator will be supervising the transport contractor on site. The transport coordinator is the contact person for all transport and/or handling activities. A transport coordinator has the following privileges:

- Upload documents/pictures/comments regarding the WPs he's involved in.
- To read information that's broadcasted (schedules, safety recommendations, CCC alarms, etc.) regarding WPs.

A.2 Functions

A function is almost the same as the role, but it doesn't depend on a person's employment status at CERN. A function is assigned for a specific WP, and one person can be assigned many functions for one WP. What function(s) a person is assigned in one WP doesn't depend on his function in the other WPs he has been involved in.

A.2.1 Requestor

This is the person which created the WP request and provided the initial information. A requestor has the following privileges:

- Fill-in work package requests and modify the request that he's involved in.
- Browse, retrieve and reuse information from the WP database.
- Upload documents/pictures/comments regarding the WPs he's involved in.

– Read access to information that's broadcasted (schedules, safety recommendations, CCC alarms, etc.) regarding WPs.

A.2.2 Supervisor

This is the person that's responsible for all activities in a WP. Should there be something wrong (delays, damages, safety issues, etc.), then he'll be the responsible person. He should provide information related to the WPs he's the supervisor for, and he should also be available if there should be any questions regarding his WP(s). Note that a WP can only have *one* supervisor. A supervisor has the following privileges:

– Fill-in work package requests and modify the request that he's involved in (a supervisor can [but doesn't need to] be the same person as the requestor).

– Browse, retrieve and reuse information from the WP database.

– Upload documents/pictures/comments regarding the WPs he's involved in.

– When a WP is completed, it is supervisor should close it to confirm that it's finished.

– Should have the ability to read information (schedules, safety recommendations, CCC alarms, etc.) that's broadcasted regarding WPs.

A.2.3 RP Contact

The RP contact is the person responsible for giving information and answers regarding radiation protection. A RP contact has the following privileges:

– Upload documents/pictures/comments regarding the WPs he's involved in. This will in most cases be RP maps and/or measurement values from a RP sweep.

– A RP contact should also have the ability to read information (schedules, safety recommendations, CCC alarms, etc.) that's broadcasted regarding WPs.

A.2.4 CCC Contact

A CERN Control Centre (CCC) contact can view the information (schedules, safety recommendations, CCC alarms, etc.) which are being broadcasted. If a WP potentially can trigger some alarms which will end up with CCC, then this person will provide answers and information regarding these potential alarms.

A.2.5 DCS/DSS Contact

A Detector Control System (DCS)/Detector Safety System (DSS) contact should, like the CCC contact, only be able to read information that's being broadcasted regarding WPs. A DCS/DSS contact is used to get information about a safety system that might get triggered during a WP.

A.2.6 Field Crew

These are the workers that perform the actual work in a WP. A person in the field crew has the following privileges:

- Read information regarding the WPs he's involved in.
- Upload documents/pictures/comments regarding the WPs he's involved in.

A.2.7 Others

This function is reserved for persons who are not involved in any WP, and they will only have read access to a limited amount of information.

A.3 Graphically Summary

The following figures give a graphically overview of the distribution of privileges. The ovals represent privileges, and the stick man is representing a user. Note that there are no separation between roles and functions in these figures.

These figures were last updated 17.August 2010.

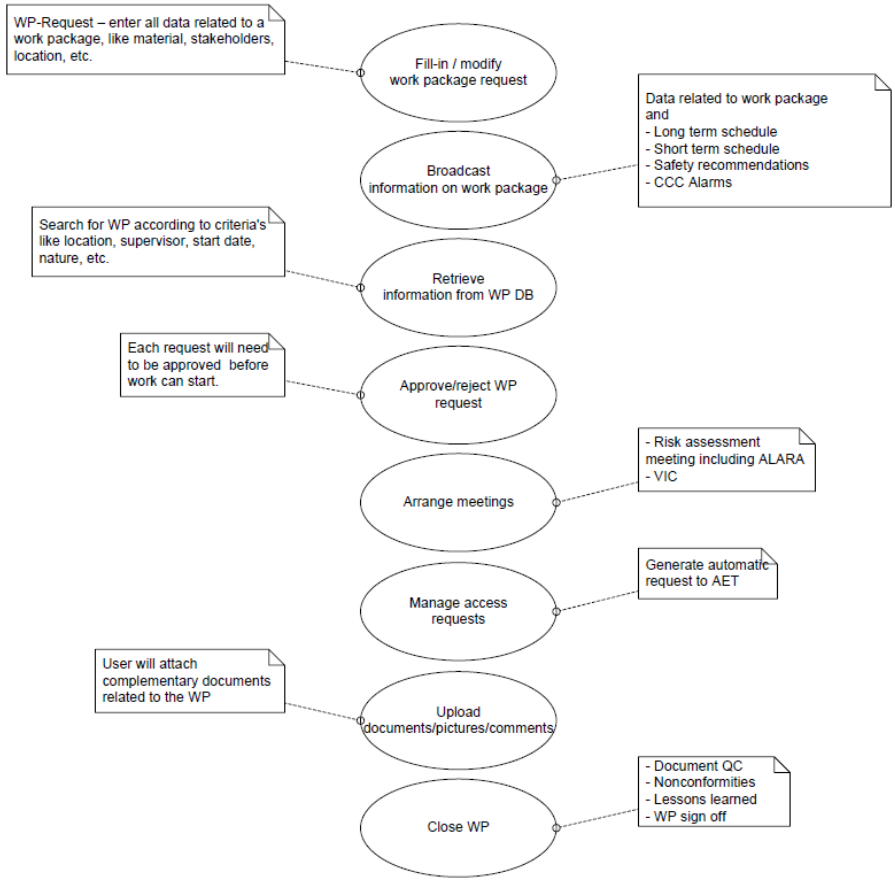

Figure A.1: Overview of the different privileges that could be assigned to the users of ACT.

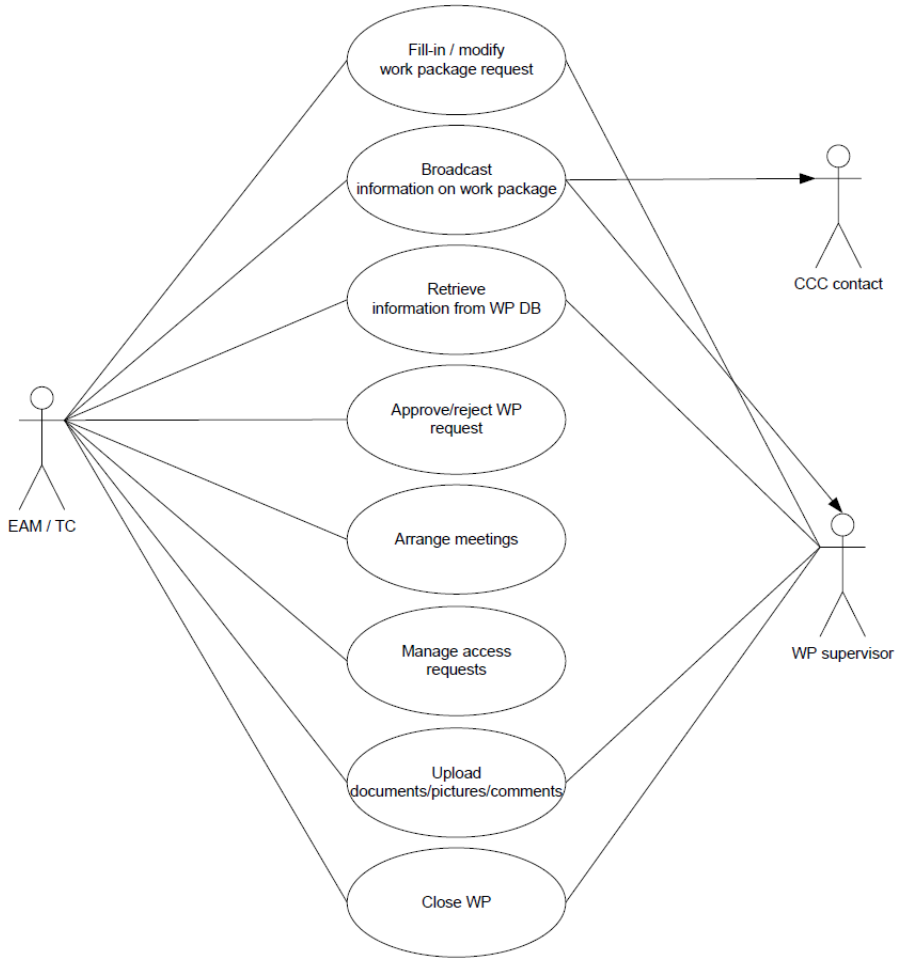

Figure A.2: Privileges of EAM, TC, CCC contact and WP supervisor.

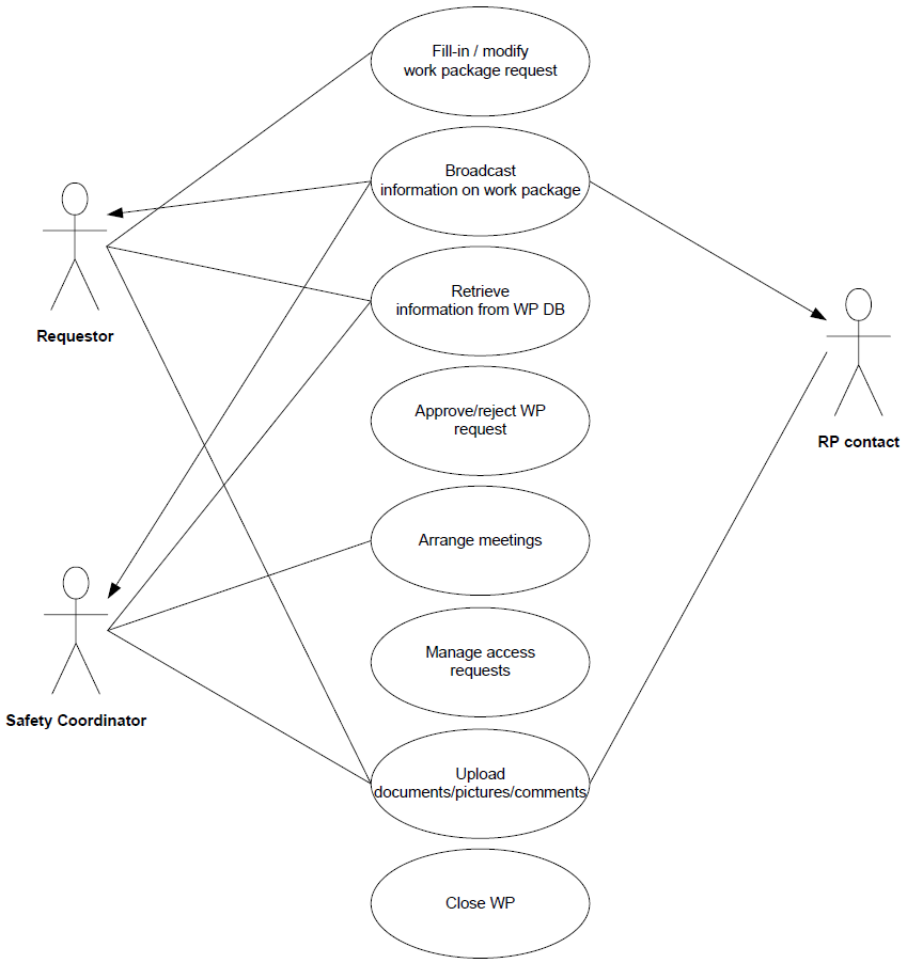

Figure A.3: Privileges of WP requestor, safety coordinator and RP contact.

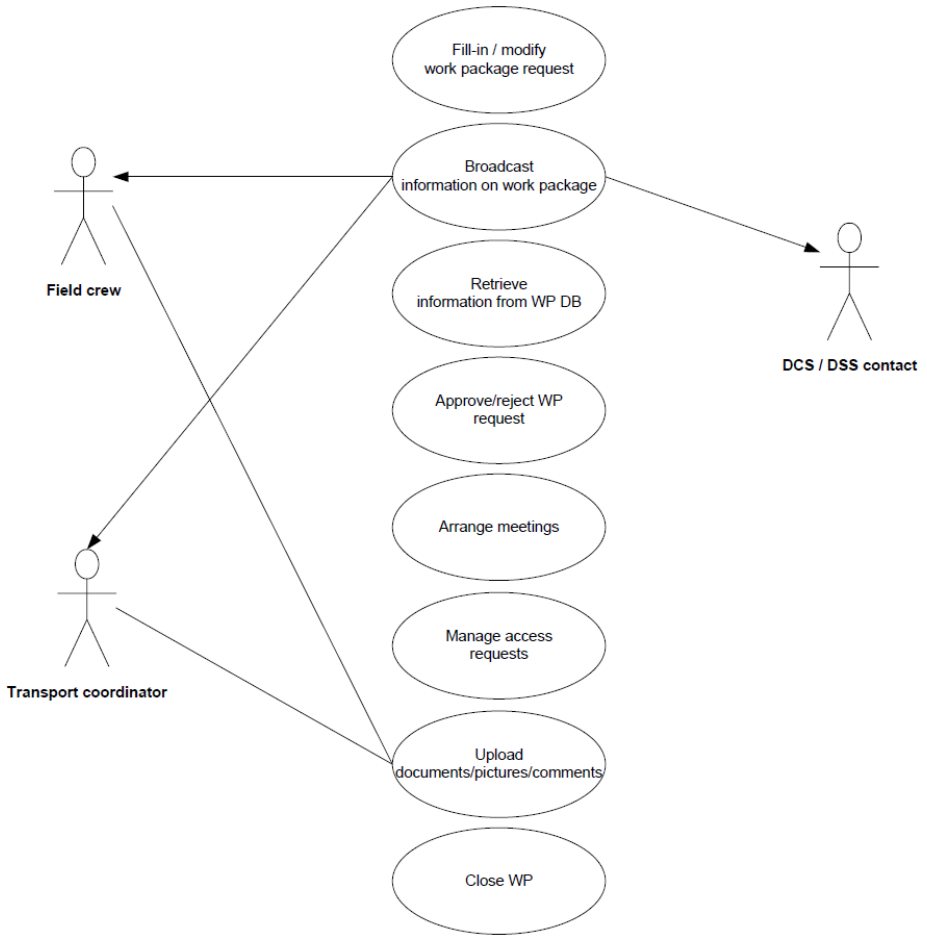

Figure A.4: Privileges of field crew, transport coordinator and DCS/DSS contact.

Appendix B

ER Diagrams for ACT

This chapter gives a short description of the majority of entities found in the ER diagrams of ACT. At the end of the chapter is all the ER diagrams created for ACT presented.

B.1 Short Explanation of the Entities

This section provides a short explanation of tables which are found in the ACT. Not all tables are explained because their naming should make their purpose fairly understandable.

Status is a table which contains the state of a WP. Its value is set by the application and can be:

- Unsigned - A WP is in this state from a WP request is submitted till it's signed off by EAM or TC.
- Foreseen - This is the state which a WP has from it is signed off till the risk assessment meeting is completed.

- Scheduled - After the risk assessment meeting the WP will be in the scheduled state.
- Ongoing - This is the state in which a WP will be in between its start date and till it is completion.
- Completed - When a WP is completed, it stay in the database and is searchable at later points in time. The transition to this state happens when a WP is closed.
- Canceled - If a WP at some stage should be cancelled, then it'll end up in this state.

Reactivity gives the preferred work hours of a WP. For example; daytime, daytime including weekends, any time, etc.

Categories tell the nature of the WP. This can be maintenance, repair, upgrade, relocation, installation, and test.

Priority gives the priority which should be assigned to the WP. Can be high, medium or low.

Neighboring facilities should contain information if a WP indirectly depends on a given state of its neighboring facilities. For example, if there's a need to shut down electricity of neighboring facilities to avoid electrical shock.

Magnet off should state if a WP requires that the magnet is off. This applies only to work in the underground facilities.

Activity types can be activity or checkpoint. A checkpoint is some sort of control activity from which the other WP activities can't continue from without being passed.

WP persons contain a list of persons involved in a specific WP. This table also maps a person's name to a function for that WP.

Functions contain the functions which can be assigned to a person involved in a WP. This table does, at the time of writing also contain the role. These will be distinguished at a later point in the development.

Meeting type can be VIC or risk assessment meeting.

Misc is a general table which will contain all temporary information and information which isn't covered by other tables. E.g. if a tool isn't found in MTF, then the user could continue by saying that the tool will be certified, and the temporary information will be stored in the misc table.

Risks should contain short information about risks which a WP will involve. E.g. risk of falling, risk of Oxygen Deficiency Hazards (ODH), risk of fire, etc.

Potential alarms is a list of alarms which the WP might trigger. It should also contain information about the potential receiver of that alarm. In this way, the receivers can get a warning about potential alarms.

Expected waste is a list of waste that will be produced during the WP execution.

Waste group is a fixed list of waste types that might be produced. Examples of types can be liquid, metal, wood chips, etc.

Comments table which stores all comments related to a WP.

Reference documents a table which stores all documents related to a WP. These documents are stored on a file server, and the information in this table is used when accessing the documents.

Ref doc type a table used to group the documents for a WP. This can be IS37, EDH transport, permits, pictures, and all type of text documentation.

Materials is a table of necessities for performing the work. This can be screws, iron bars, pipes, a piece of wood, etc.

Equipments is a table referring to equipments which will be brought in to the cavern for installation in the facilities. It might also be equipment that's brought in for replacing similar equipment. Examples of equipment can be power supplies and sensors.

Objects is about the same as location, but in a different way. An object could be a very precise device like "central beam pipe", which is spread over many locations.

Meetings contain information regarding the meetings held for a specific WP.

Tools is a list of the tools which is needed to perform the work of a WP.

Locations is a list of locations where the WP needs to perform some work. A location is referred to as a zone in the EMDb table.

B.2 ER Diagrams for ACT

The ER diagrams found in this section were last updated 22.September 2010.

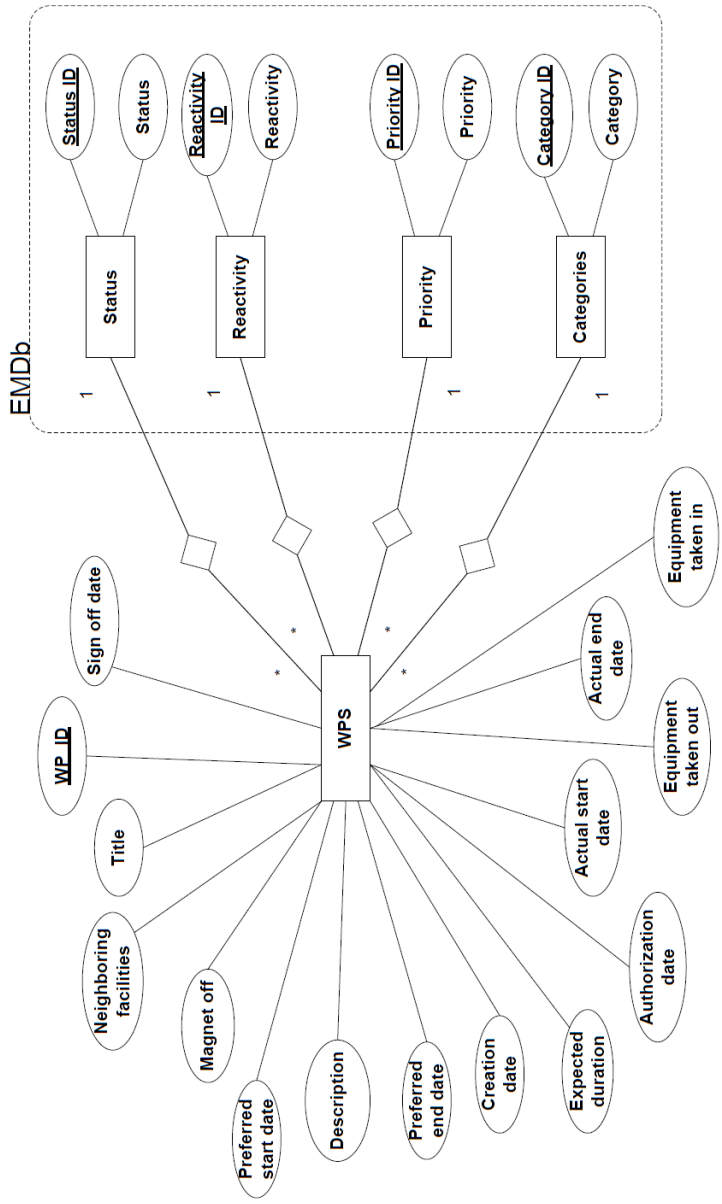

Figure B.1: Part 1/5 of the ER diagrams for ACT.

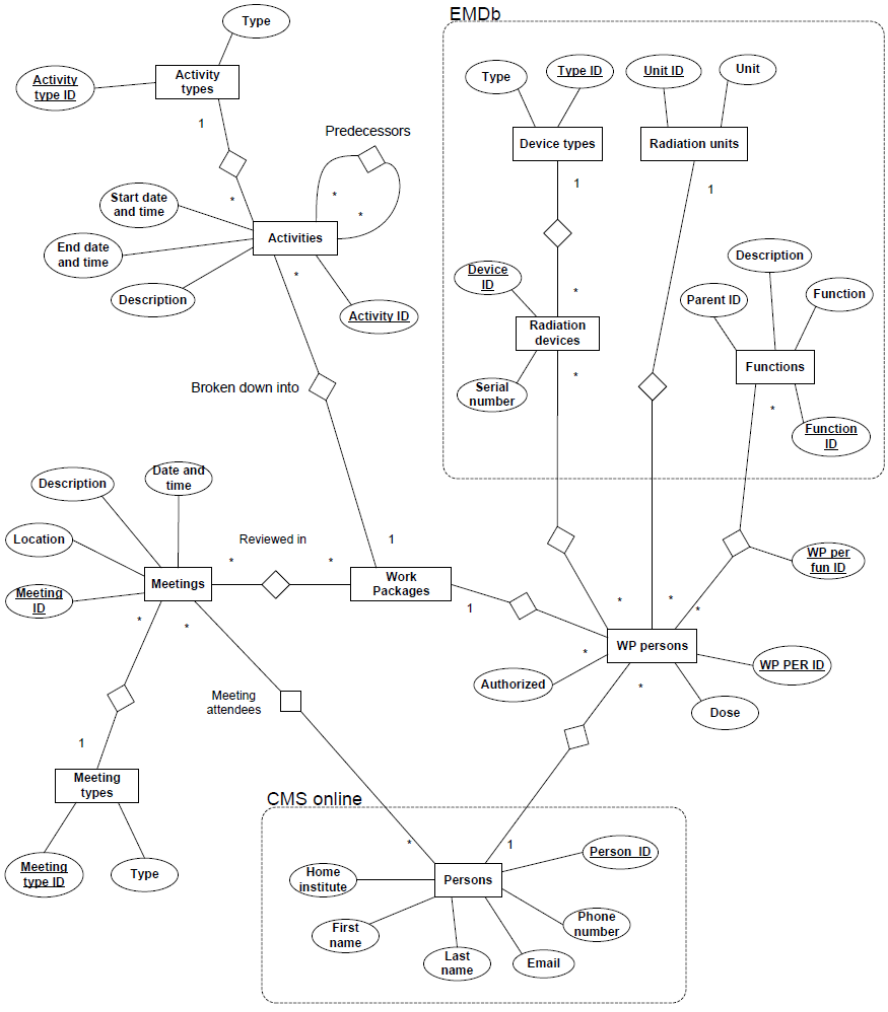

Figure B.2: Part 2/5 of the ER diagrams for ACT.

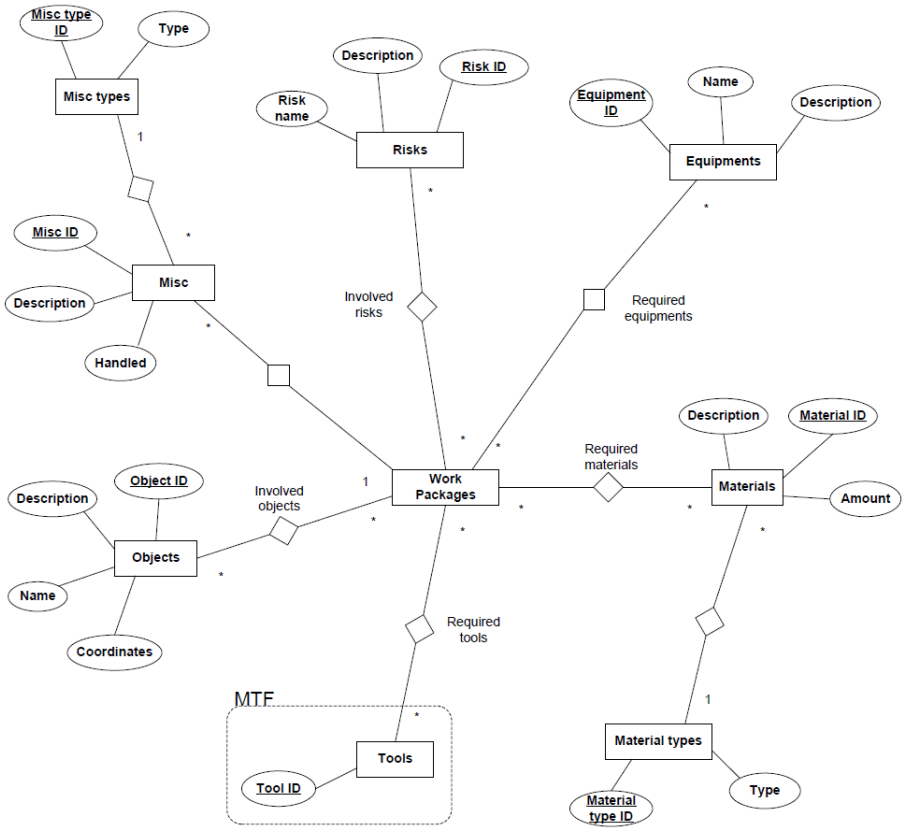

Figure B.3: Part 3/5 of the ER diagrams for ACT.

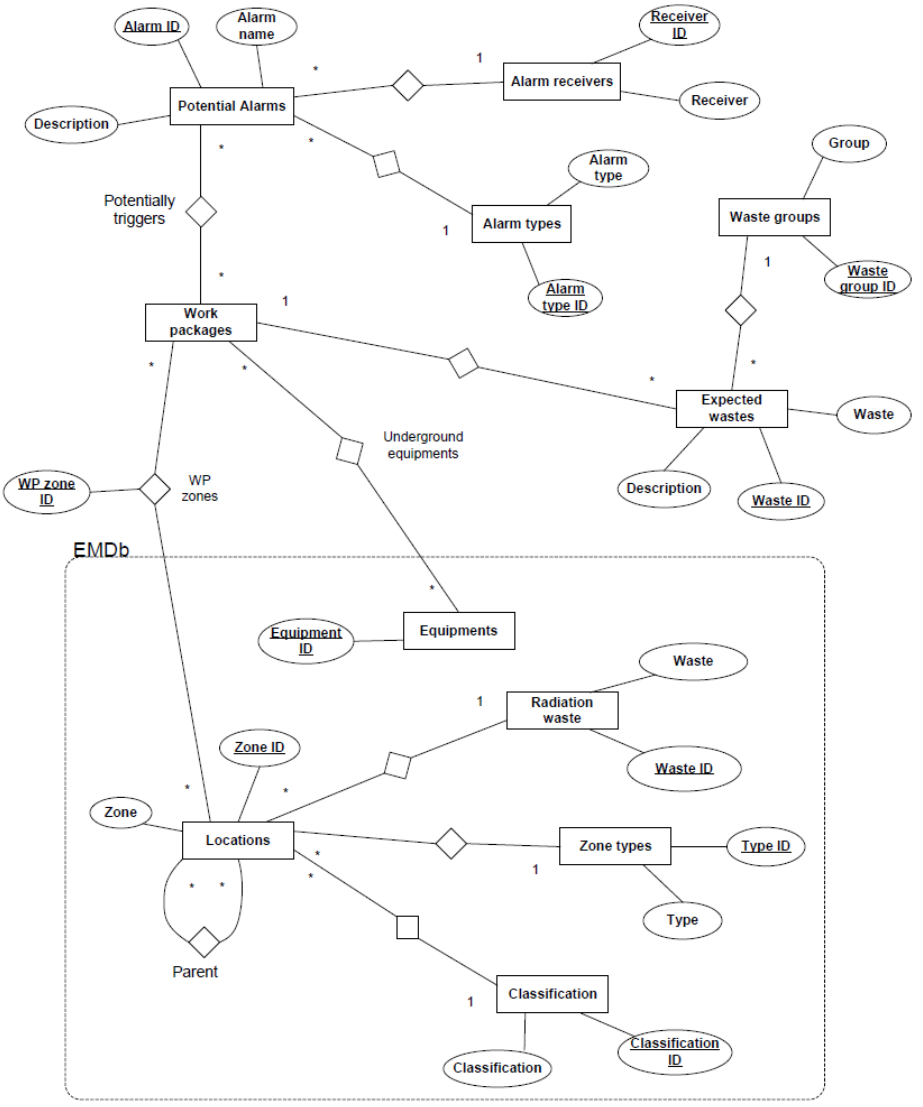

Figure B.4: Part 4/5 of the ER diagrams for ACT.

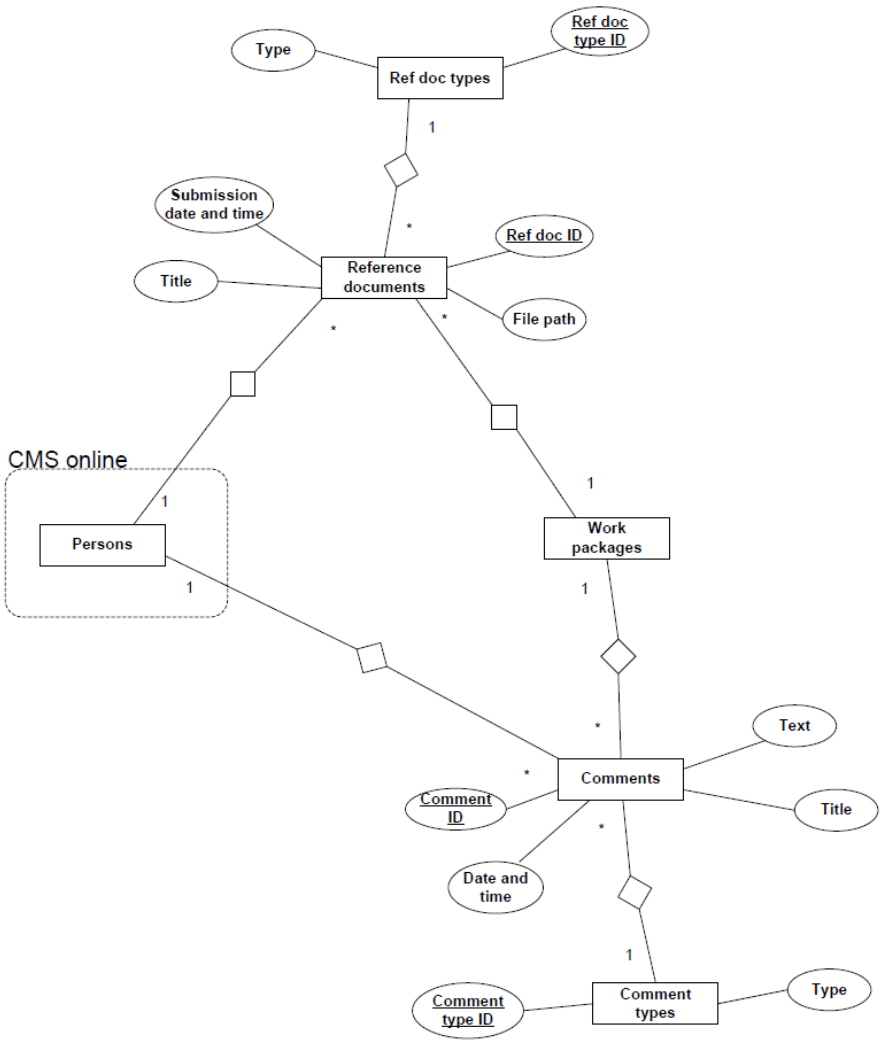

Figure B.5: Part 5/5 of the ER diagrams for ACT.

Appendix C

Relational DB Schemas

This chapter presents the relational DB schemas created for ACT.

These schemas were last updated 22. September 2010.

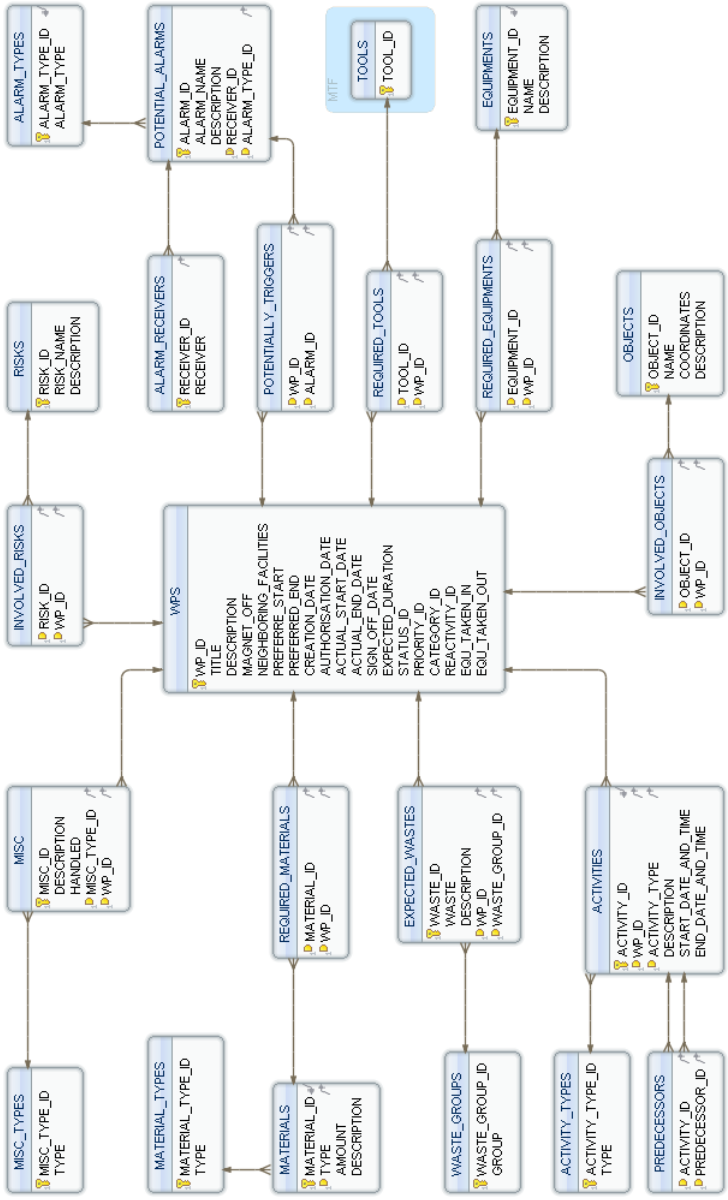

Figure C.1: Part 1/2 of the relational DB schema for ACT.

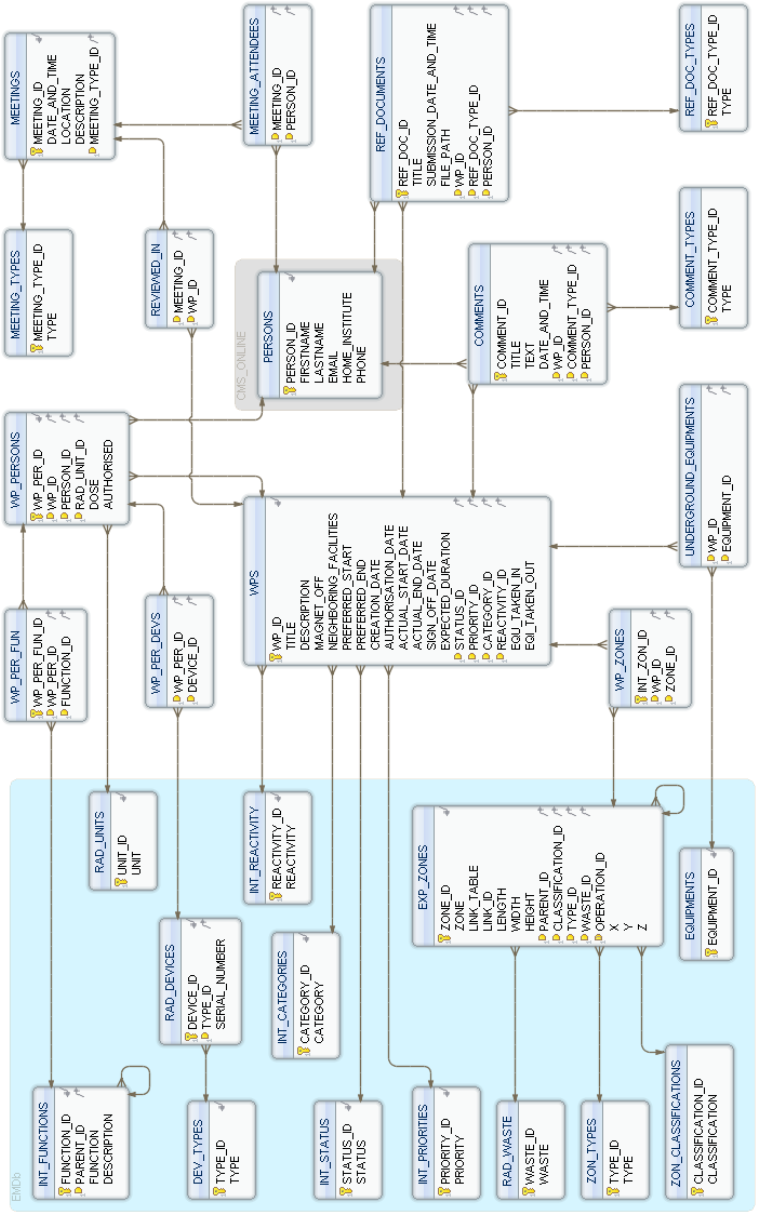

Figure C.2: Part 2/2 of the relational DB schema for ACT.

Appendix D

Access to CMS Online

Persons have access to the CMS online network depending on what CERN group or Experiment they belong to. This chapter provides a short description of the CERN groups and Experiments which have access to the CMS online network.

D.1 Experiments

D.1.1 CMS

The CMS (Compact Muon Solenoid) experiment uses a general-purpose detector to investigate a wide range of physics, including the search for the Higgs boson, extra dimensions, and particles that could make up dark matter. Although it has the same scientific goals as the ATLAS experiment, it uses different technical solutions and design of its detector magnet system to achieve these [57].

D.1.2 TOTEM

The TOTEM (TOTal Elastic and diffractive cross section Measurement) experiment studies forward particles to focus on physics that is not accessible. To do this TOTEM must be able to detect particles produced very close to the LHC beams [57].

D.2 Groups

D.2.1 EN-CV

The EN-CV (Engineering Department – Cooling and Ventilation) group concerns the operation and maintenance of the cooling systems, pumping stations, air conditioning installations and fluid distribution systems for the PS, SPS and LHC including their experimental areas and special cooling systems of LHC sub-detectors. It also provides service to the Computer Centre and some miscellaneous installations [58].

D.2.2 EN-EL

The EN-EL (Engineering Department – Electrical Engineering) group operates, maintain, renovate and extend the CERN electrical distribution network from 400kV to 400/230V. They analyze and make projections for CERN electrical energy consumption, make modifications and/or extensions to the network, and they also provide all cabling and fiber optics installations for accelerators and experiments [58].

D.2.3 EN-MEF

The EN-MEF (Engineering Department – Machines & Experimental Facilities) group is responsible for planning and coordinating all maintenance and installation activities in the CERN Accelerators Complex. They provide support to the Experiments using the CERN accelerators, during their preparation, integration, installation and operation. They also design, installation and maintenance of the secondary beam zones, and

provide the integrated layouts of the machines, including the optimization of the layout of the regions around the LHC experiments [58].

D.2.4 EN-MME

The EN-MME (Engineering Department – Mechanical & Materials Engineering) group is responsible for providing the CERN community specific engineering solutions combining mechanical design, production facilities and material sciences [58].

D.2.5 EN-HE

The EN-HE (Engineering Department – Handling Engineering) group is responsible for providing transport and handling services for the technical infrastructure of CERN, accelerators and experiments. This includes the design, the tendering/procurement, the installation, the commissioning, the operation, the maintenance and decommissioning of standard industrial and custom built transport and handling equipment [58].

D.2.6 GS-SEM

The GS-SEM (General Infrastructure Services Department – Site Engineering and Management) group provides and maintains the sites, buildings and underground civil engineering works of the organization as well as the necessary logistics and stores services [59].

D.2.7 GS-ASE

The GS-ASE (General Infrastructure Services Department – Access, Safety and Engineering tools) group provides the organization's systems for personnel safety and access control. The group procures and supports the organization's engineering, equipment data management tools and mechanical CAD systems [59].

D.2.8 IT-CS

The IT-CS (IT Department – Communication Systems) group is responsible for all communication services in use at the laboratory. This includes networks and communication for data, voice and video. In order to support the different services, the group operates a large campus IP/Ethernet network installation, which reaches out to almost every building on the CERN sites. The campus network is connected to the Internet, and it also interconnects with high capacity links to collaborating institutes. The group is also operating the fixed-wire telephone system, the GSM mobile phone system and the radio system for the emergency services [60].

D.2.9 TE-VSC

The TE-VSC (Technology Department – Vacuum, Surface and Coatings) group is responsible for design, construction, operation, maintenance and upgrade of high & ultra-high vacuum systems for Accelerators and Detectors. Coatings, surfaces treatments, surface and chemical analysis for Accelerators and Detectors are also under their responsibility [61].

D.2.10 BE-ABP

The BE-ABP (Beams Department – Accelerators & Beams Physics) group is in charge of beam physics issues and alignment over the complete CERN accelerator chain from the sources to the LHC. The ABP group is responsible for the organization of machine developments and contributes to operation supervision and the machine and detector alignment metrology. It is also in charge of operation and developments of hadron sources and the supervision and coordination of the hadron linacs [62].

Appendix E

CERN Databases

This chapter provides a short explanation about what information is found in the various databases which ACT somehow involve.

E.1 Foundation

The Foundation database provides information about CERN personnel. Name, phone number, e-mail address, ID number, etc. can be obtained and verified here. Foundation is a database that's being copied into the CMS online database, and is therefore locally available for the ACT schema.

The copy of Foundation is made every 24 hours .This copy frequency can create some problems. Consider a WP which involves a couple of workers. For these workers to be allowed to perform any work at CERN, they need to be registered in the CERN system (for insurance issues, etc.). Sometimes these workers arrive the day that the WP is supposed to start, and they perform the registration in the morning. In the afternoon they would then be in the CERN system. The ACT application, however, will not be able to see this new information before a new copy is taken of the Foundation database. In other words, the information might not be available for ACT before the

next day. Despite this potential lag in the information, the local copy is used instead of the Foundation database itself.

E.2 D7i

The D7i database is used by, among others, persons at CCC to get information about what's going on at different locations at CERN. Initially the plan was to copy a limited amount of WP information from ACT to D7i. This would make the WP information accessible for a wider group of persons. After a new meeting with persons at CCC, a view (see 3.2.2 on page 22) of the ACT database has been created instead of copying the information. The reason for still having D7i as a database in the overview is because MTF is located in this database.

E.3 MTF

MTF is part of the same database as D7i, but is designed to store information about tooling. This will, in a later version of the application, be used to confirm that a tool is CERN certified. Only certified tools are allowed to be used in a work activity, and all tools should be found here.

E.4 EMDb

EMDb is a database which is responsibility of the traceability radioactive material at CMS. All equipment, which has become radioactive during operation, are registered and made traceable according to French law. In addition to tracing radioactive material, it also keeps track of values collected during radiation measurement.

EMDb contain a list of all CERN certified equipment which is used underground. This information will be used by ACT in order to associate equipment with WPs. EMDb also contain a table structure for locations, which are used by the ACT application.

E.5 CATIA

CATIA is a system which will provide 3D images of the work area. When a location or object is selected in the application, images of these will be displayed if they are available in CATIA. This functionality is not planned to be implemented in the first versions of the ACT application.

E.6 ADaMS

ADaMS use information contained in a variety of databases to govern the access control at all CERN facilities.

ADaMS provides the functionality to predict what access rights a person currently has. There is also functionality for predicting a person's access rights in the future. Note that the functionality for predicting a person's access rights in the future will not be 100% trustworthy. The calculation is based on a lot of criteria's like:

- Safety and CTA course(s).
- If there are exceptions for this person (e.g. if a person has misbehaved or he's the president of some country).
- Access card status.

Most these criteria's are easy to foresee due to the "validity end" date they contain. However, the calculations also involve dosimeter[1] data. This information is kept in another database and it's not shared with other database systems at CERN. The only information that ADaMS can retrieve from this database is a dosimeters state (valid or invalid). If it's invalid, the person will looses access rights for some facilities.

The biggest uncertainty with the future access right calculations is associated with the dosimeter data. If an access right test is done a couple of weeks before the access is going to happen, one might get "ok" as an answer. However, at the day when the access

[1] A dosimeter is a device which registers how much radiation a person has been exposed to.

is needed, the person might not have access anymore. This can happen if the person gets to much radiation in this short period of time and his dosimeter state has changed to invalid, or he could have misbehaved and has therefore lost he's access rights to the radioactive areas. Despite this, he can still access the non-radioactive areas which haven't been blocked.

These functionalities, provided by ADaMS, is planned to be used in the first version of the ACT application.

E.7 AET

AET is a database which interacts with ADaMS to provide a service for creating slots of access rights. These time slots are currently fixed and cannot be expanded after they are granted.

The functionality of expanding an access slot is wanted by the ACT application, since a WP might need more time at the end. This functionality is a bit hard to implement due to changing environments, like the physical configuration of the detector, or an end of a technical stop. Another problem with AET is the dosimeter issue which is also experienced by ADaMS.

AET is being developed at the time of writing, and is not implementation in ACT yet.

E.8 EDMS

EDMS is a file server which facilitates storage of files. EMDb does have a similar functionality, but this is implemented using Binary Large Objects (BLOB). BLOB works just fine but its performance and user friendliness isn't as good as with a file server. Picture and any type of documents added to a WP will be stored in EDMS. This functionality is not implemented in the first release of ACT.

Appendix F

Basic ER Diagram Terminology

This chapter provides a short explanation of basic ER diagram symbols. All ER diagram concepts used in the ACT schema will be covered here.

F.1 Entity

An entity can be defined as the general name for the information that's going to be stored within a single table [27, 32].

ER diagrams represent entities with rectangles.

F.2 Attribute

An attribute is a piece of data that is stored about each entity [25].

Attributes are represented as ovals linked to an entity.

F.3 Primary Key

A primary key is an attribute, or a combination of attributes, that uniquely identify a row of data found in a table. When multiple attributes are used to derive a primary key, this key is known as a concatenated, or composite, primary key [32].

A surrogate primary key is an artificial primary key used when a natural primary key is either unavailable or impractical. From a performance point of view, an integer used as a surrogate primary key can often provide better performance in a join operation than a composite primary key [29].

Primary key(s) are represented as an attribute which has its name underlined.

F.4 Foreign Key

A foreign key is a primary key of one entity put into another entity. The foreign key will create a reference between the tables which is used when retrieving related information from multiple tables [25, 28].

F.5 Relationships

Relationships between entities are represented by lines, connecting the entities, with a diamond/square in the middle. Each line describes the relationship in both directions. If a relationship involves the same entity type more than once, it's called a recursive relationship [24, 27].

It's possible to model relationships involving more than two entities. If there are three entities in one relationship, it's called a ternary relationship.

It's also possible for relationships to have attributes.

F.6 Cardinality Constraint

Cardinality constraints specify how many instances of an entity relate to an instance of another entity [27].

Cardinality constraints are represented by a number/asterisk on each end of a relationship. The cardinality constraint belonging to an entity is the number/asterisk that's closest to the entity on the opposite side of the relationship. The cardinality constraints are:

- A 1:1, or one-to-one, relationship from entity type S to entity type T is one in which an entity from S is related to at most one entity from T and vice versa [24, 32].
- A N:1, or many-to-one, relationship from entity type S to entity type T is one in which an entity from T can be related to two or more entities from S, and an entity from S can be related at most one entity from T [24].
- A 1:N, or one-to-many, relationship from entity type S to entity type T is one in which an entity from S can be related to two or more entities from T, and an entity from T can be related at most one entity from S [24].
- A N:M, or many-to-many, relationship from entity type S to entity type T is one in which an entity from S can be related to two or more entities from T, and an entity from T can be related to two or more entities from S [24, 32].

Appendix G

Oracle Terms and Concepts

This chapter gives an introduction to some technical terms and concepts used in Oracle database systems. This chapter starts off with some basic information in order to gradually go deeper into details. Overviews of how things relate will also be presented.

G.1 Instance and Database

The terms instance and database are often, since they're so closely connected[1], used interchangeably. In Oracle, a database is a set of files which is located on a disk - or in other words, the database represents the physical storage of information. An instance, on the other hand, refers to a set of memory structures and background processes. When a user requests information from a database, it's the instance that receives the request, not the database itself [1, 2, 6].

Files in a database can exist independently of an instance, and an instance can exist independently of a database. An instance can associate with one, and only one, database at any given time [1, 2, 6]. The relationship between instance and database is illustrated in figure G.1.

[1] The term *Oracle database* is sometimes used to refer to both instance and database [6].

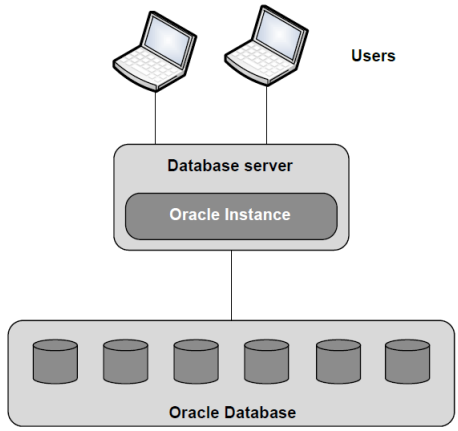

Figure G.1: A simple illustration of the relationship between a database and an instance [1].

G.2 Datafiles

Every Oracle database has one or more *physical* datafiles containing all data stored in the database [6]. A datafile is composed of Oracle database blocks that, in turn, are composed of operating system blocks on a disk.

The first block of each datafile, called *datafile header*, contains information to help track the current state of the datafile, and is critical for maintaining the overall integrity of the database. One of the most critical pieces of information contained in the header is the checkpoint structure. This is a logical timestamp that indicates the last point at which changes were written to the datafile. This timestamp is an important piece of information during an Oracle recovery process as the timestamp in the header determines which redo logs (these concepts will be explained shortly) to apply when recovering the datafile to a point in time [1, 2].

From a physical point of view, a datafile is stored as operating system blocks. From a logical point of view, datafiles have three intermediate organizational levels, which will be explained next, namely: data blocks, extents and segments [1, 2]. Figure G.2 gives an illustration of how segments, extents and blocks relate.

G.3 Data Blocks

Data blocks are part of the *logical* storage structure and represent the smallest unit of storage in Oracle. One data block corresponds to a specific number of bytes on disk [6].

G.4 Extents

An extent is a specific number of *logically* contiguous data blocks, obtained in a single allocation, used to store a specific type of information [6].

G.5 Segments

Segments are, like extents and datablocks, part of the *logical* storage structure. Segments are the storage objects within the Oracle database, which can be a table or an index. In most cases a single segment cannot reside in more than one tablespace, and they will consist of one or more extents [3, 6].

A segment will represent a *single* instance of e.g. a table or an index. For example, a table with two indexes is implemented as three segments in the schema [5].

G.6 Tablespace

All data stored in an Oracle database must reside in a *logical* structure called a tablespace. Each tablespace is composed of one or more datafiles, and act like a *logical* container for a segment. Tablespaces are the link between the physical and logical world of Oracle [1, 2, 6].

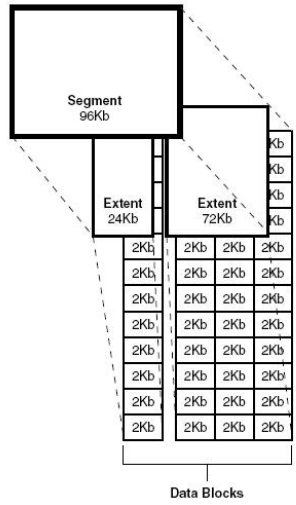

Figure G.2: Relationship between segments, extents and blocks in an Oracle database [18].

G.7 Schemas

Schemas are *logical* structures containing objects like segments, views, procedures, functions, triggers, sequences, synonyms, database links, and so on. A schema is a user-created structure that directly refers to the data in the database. A schema can contain many segments and many segment types [5, 6].

Figure G.3 illustrate how database, tablespace, datafiles, segments, extents, block, and schemas are intertwined.

G.8 SGA

The SGA is a group of shared memory structures that contain data and control information for one database instance. Among the things found in SGA is database buffer

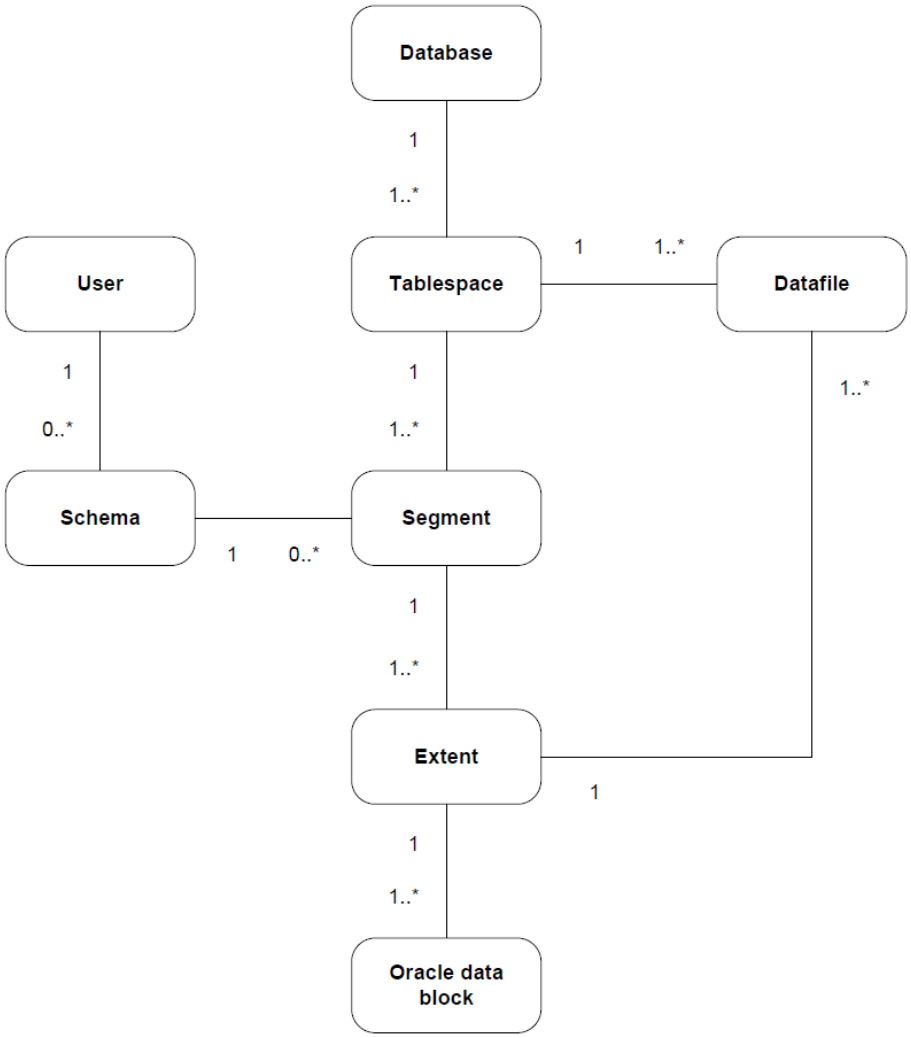

Figure G.3: Relationship between database, tablespace, datafiles, segments, extents, block, and schemas. This figure is based on [19].

cache[2], redo log buffer[3], and the library cache [4] [6]. The SGA also contains background processes, and the most important of these processes are [1, 2, 4]:

- The Database Writer (DBWR) process writes database blocks from the database buffer cache to the datafiles on disk. Blocks of the database buffer cache are written to disk when Oracle performs a checkpoint, when Oracle needs space in the database buffer cache, or when the database is shut down.
- The Log Writer (LGWR) process is responsible for redo log buffer management and writes redo information from the redo log buffer to all copies of the current online redo log file. LGWR is signaled to do these writes when a user session is committed, when the log buffer is nearly full, and other times as required.
- The Archiver (ARCH) process reads the redo log files once Oracle has filled them and writes a copy of the used redo log files to the specified archive log destination(s).
- The Checkpoint (CKPT) process updates datafile headers whenever a checkpoint is performed.
- The System Monitor (SMON) process maintains the health and safety for an Oracle instance.
- The Process Monitor (PMON) process watches over the user processes that access the database.

Figure G.4 gives an overview of the datafiles which these processes interact with.

G.9 PGA

PGA is a memory region that contains data and control information for a server or background process. Access to the PGA is exclusive to the server processes [3, 6].

[2]With few exceptions, any data coming in or going out of the database will pass through the database buffer cache. When a user makes changes to a block, those changes are made on a block residing in the cache [1, 3].

[3]The Redo Log buffer contains user data that's not yet committed [3].

[4]The Oracle's library cache is responsible for collecting, parsing, interpreting, and executing all of the SQL statements that's going to the Oracle database [3].

G.10 System Change Number

SCN is a counter that represents the current state of the database. Each SCN represents a point in the life of the database and is bumped up every time a transaction is committed.

The SCN is important for recovery operations. The number will be used to define where a recovery will start and when recovery may end [2, 4].

G.11 Redo Log Files and Online Redo Logs

Redo log files is one of the fundamental physical files that make up an Oracle database (the other ones are datafiles and control files). Redo log files contain a recording of changes made to the database as a result of committed transactions and internal Oracle activities. Information contained in the redo log files should enable an Oracle database to recover these transactions or internal activities [4].

A redo contains all information needed to reconstruct changes made to the database. Redo information is written to disks whenever a commit is received. Writing changes made to the database buffer cache might however be deferred until it's more efficient for the DBWR process to flush the changes. I/O operations to disk are much slower compared to memory operations, so deferring write operations will improve the performance of the database. When a failure occur, and some changes have not yet been written to disk, the redo can be used to play back the changes that has occurred since the last write to the disks. Redo logs can, in addition to redo changes, also be used to undo[5] changes made to the database [1, 4, 5].

The redo log buffer also caches uncommitted changes to the database. By avoiding constantly writing to the redo log disks will improve the performance of the database. The redo information will be written to disk when a commit is received [3]. The changes contained in the redo log buffer will be written to an Online Redo Log on disk after a commit is received [2, 3].

Figure G.4 illustrate how the redo log buffer and the online redo logs relate.

[5] When data is changed, a "before images" of the data is created to allow a transaction to roll back [5].

G.12 Archived Redo Logs

There is a finite number of online redo log files (Oracle requires at least two online redo log files). These files are accessed in a round robin fashion. As an online redo log file fills with redo, Oracle switches to the next online redo log file. This is often referred to as a *log switch*. When a log switch occurs, filled logs are reinitialized and overwritten. This erases the history of changes made to the database, and can lead to unrecoverable data. [1, 2, 3].

There are two modes concerning an Oracle database's action when a log switch occurs.

- If the database is in *archivelog* mode, and the *arch* process is running, at each log switch a copy of the online redo log being switched from will be made. This copy is called an archived redo log, and it's applied during media recovery operations [2]. The archived redo log in combination with the online redo log provides a complete history of all changes made to the database [1]. Without archived redo logs, the database backup and recovery options available is severely limited [7].
- *Noarchivelog* mode, as the name implies, do not make archived redo logs. This mode permits normal database operation, but does not provide the capability to perform point-in-time recovery or online (hot) backups. By default, the database is created in noarchivelog mode, and if online backups are wanted, then the database should be run in archivelog mode. One advantage of noarchivelog mode is its simplicity compared to archivelog mode [1, 2].

Figure G.4 show how information is passed from the online redo log to the archived log.

G.13 Control Files

Control files are one of the fundamental *physical* files that make up an Oracle database. The control file contain key information about the content and state of the database, the current SCN, where physical files of a database can be found, what header information each file currently contains (or should contain), datafile information, redo log

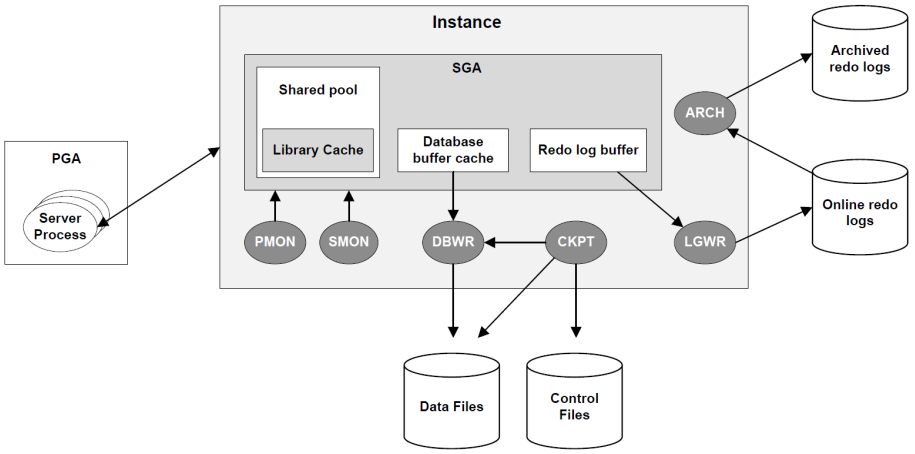

Figure G.4: Fundamental processes for memory and storage interactions between an instance and an Oracle database. This figure is based on information found in [1, 2, 20].

information, and archived log information. Since the control file contains a great deal of database information which Oracle cannot function without, there should at least be two control files on different physical disks [1, 2, 3, 5].

The control file ages information out as it needs space, but not all information can be eliminated - for instance, the list of datafiles. This information is critical for minute to minute database operation, and new space must be made available for these records. The control file separates the internal data into two types of records. Circular records are records that include information that can be aged out of the control file. Noncircular reuse records are those records that cannot be sacrificed. If the control file runs out of space for the noncircular reuse records, the file expands to make more room [2].

G.14 Checkpoints

Oracle writes redo information when a transaction is committed. Since the actual data block is written to disk at a later point there are situations when the redo log contain changes not yet reflected in the actual database [5].

When a checkpoint is performed, Oracle writes (through DBWR) all the changed database blocks to disk and update the file headers. In this way the datafiles on disk "catches up" with the redo logs. These writes occur in such a way as to not hamper the database performance. There are several different kinds of checkpoints. An event that results in a checkpoint is a log switch, database shutdowns, and when a tablespace is taken in or out of the online backup mode. Completed checkpoints are recorded in the control file, datafile headers, and redo log [1, 2, 3].

Checkpoint also represents a point in time when all items in the database buffer cache and database datafiles are consistent and roll-forward recovery can be performed [3, 4, 5].

Figure G.4 show what datafiles a chechpoint process interact with.

G.15 Basic Concepts of the Database Recovery Process

Reconstructing the contents of a database, from a backup, typically involves two phases: retrieving a copy of the datafile, and reapplying changes made to the datafile since the backup was made. These changes are retrieved from the archived and online redo logs [7].

To restore a datafile from backup is to retrieve the file from a backup location, and make it available to the database server [7].

To recover a datafile (also called performing recovery on a datafile), is to take a restored copy of the datafile and apply to it the changes recorded in the redo logs [7].

G.16 Dedicated or Shared Server Process

An Oracle database server can be configured to run dedicated or shared server architecture.

In a dedicated server environment there's a dedicated server process[6] for each user process. The benefit of this is that each user process has a dedicated server process to handle all of its database requests. The problem is that each server process is often idle a large percentage of the time, consuming system resources [5]. This is, despite its potential huge resource consumption, the solution used at CERN. The amount of users connecting to the database isn't large, and the connections often involve a lot of requests to the database. The low connection frequency of new users, and the amount of requests generated on average by each user, justifies the use of dedicated server processes.

Shared server architecture offers increased scalability for large number of users. This is possible because a single server process can be shared among a number of user processes. The shared server architecture is much more scalable then a dedicated architecture as the number of users for a system increases [5].

[6]Server processes are the interface between the Oracle database server and the user processes. Database communication between a user process and a database must go through a server process [5]. Each server process uses memory in PGA for its operations [1].