



UNIVERSITY
OF
JOHANNESBURG

COPYRIGHT AND CITATION CONSIDERATIONS FOR THIS THESIS/ DISSERTATION



- Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- NonCommercial — You may not use the material for commercial purposes.
- ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

How to cite this thesis

Surname, Initial(s). (2012) Title of the thesis or dissertation. PhD. (Chemistry)/ M.Sc. (Physics)/ M.A. (Philosophy)/M.Com. (Finance) etc. [Unpublished]: University of Johannesburg. Retrieved from: <https://ujdigispace.uj.ac.za> (Accessed: Date).

ERAO
WIID

AUDIT RISKS IN A DATABASE ENVIRONMENT WITH SPECIFIC REFERENCE TO
ORACLE7

BY

LINÉ CORNETTE WIID

SHORT DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE

MASTER OF COMMERCE

IN



IN THE

FACULTY OF ECONOMIC AND MANAGEMENT SCIENCES

AT THE

RAND AFRIKAANS UNIVERSITY

STUDY LEADER : PROF. A. DU TOIT

JOHANNESBURG

OCTOBER 1995

DECLARATION

I declare that except to the extent acknowledged in the text, this short dissertation hereby submitted to the Rand Afrikaans University for the degree of Master of Commerce is my own unaided work and has not been submitted previously for any degree to any other university.

LINÉ C WIID



INDEX

Chapter	Page
OPSOMMING IN AFRIKAANS	i
SYNOPSIS	vi
1. INTRODUCTION	1
2. RISKS AND CONTROLS IN A GENERAL DATABASE ENVIRONMENT	10
3. RISKS AND CONTROLS IN AN ORACLE7 (CLIENT/SERVER) DATABASE ENVIRONMENT	32
4. SUMMARY	68
5. CONCLUSION	76
BIBLIOGRAPHY	79

LOUDITRISIKO'S IN 'N DATABASISOMGEWING MET SPESIFIEKE VERWYSING NA
ORACLE7

Deur

LINÉ CORNETTE WIID

OPSOMMING VAN VERHANDELING INGEDIEN VIR DIE GRAAD MAGISTER
COMMERCII IN REKENAARLOUDITERING IN DIE FAKULTEIT EKONOMIESE EN
BESTUURSWETENSKAPPE AAN DIE RANDSE AFRIKAANSE UNIVERSITEIT



JOHANNESBURG

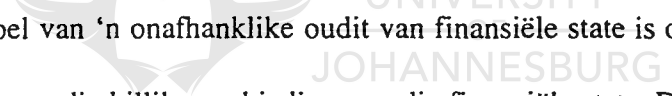
OKTOBER 1995

OPSOMMING IN AFRIKAANS

Die doel van die opsomming is om die agtergrond, metodiek en gevolgtrekkings van die navorsing weer te gee. Hierdie opsomming is onder die volgende hoofde uiteengesit:

1. PROBLEEMOMSKRYWING EN DOEL VAN HIERDIE NAVORSING
2. NAVORSINGSONTWERP EN -METODIEK
3. RESULTATE EN GEVOLGTREKKINGS

1. PROBLEEMOMSKRYWING EN DOEL VAN HIERDIE NAVORSING



Die doel van 'n onafhanklike audit van finansiële state is om 'n mening uit te spreek oor die billike aanbieding van die finansiële state. Die ouditeur behoort toereikende audit bewyse kry wat hom in staat stel om gevolgtrekkings te maak ter ondersteuning van die inhoud van sy verslag. Die ouditeur behoort 'n begrip van die entiteit se rekeningkundige stelsel en verbandhoudende interne beheermaatreëls te verkry te einde die geskiktheid daarvan as grondslag vir die opstel van finansiële inligting te beoordeel en om te help in die ontwerp van sy auditprosedures. As die ouditeur van voorneme is om op enige interne beheermaatreëls te vertrou, behoort hy daardie beheermaatreëls te bestudeer en te evalueer.

Indien 'n databasisstelsel in gebruik is, is dit logies dat alle finansiële data op die

databasis sal wees. Ten einde 'n mening uit te spreek oor die finansiële state van die entiteit moet die ouditeur bepaal of hy op die integriteit van die finansiële data in die databasis kan vertrou.

Die doel van die studie was om die risikos en beheermaatreëls in 'n algemene databasisomgewing sowel as die Oracle7-databasisbestuurstelsel te identifiseer. Hierdie omgewings is in tabelvorm met mekaar vergelyk en 'n interne beheervraelys vir Oracle7 is ontwikkel.

2. NAVORSINGSONTWERP EN -METHODIEK

Die benadering van die navorsing was om 'n vergelyking te tref tussen die algemene databasisomgewing en die Oracle7-databasisbestuurstelsel sowel as om 'n interne beheervraelys van toepassing op Oracle7 te ontwikkel.

METHODIEK

- (1) 'n Literatuurstudie is uitgevoer van bestaande gesaghebbende literatuur oor databasisse met die doel om alle moontlike risiko's en beheermaatreëls teenwoordig in 'n algemene databasisomgewing te help identifiseer. Die risiko's en beheermaatreëls is as 'n riglyn gebruik vir die bepaling van die risiko's en beheermaatreëls wat in 'n databasisomgewing teenwoordig is.

- (2) 'n Studie van Oracle7-handleidings is uitgevoer met die doel om alle moontlike risiko's en beheermaatreëls in die Oracle7-databasisbestuurstelsel te help identifiseer.
- (3) 'n Tabel wat 'n vergelyking tref tussen die beheermaatreëls teenwoordig in 'n algemene omgewing en in die Oracle7-databasisbestuurstelsel is opgestel en 'n interne beheervraelys is ontwikkel.

Ten einde die studieveld af te baken is 'n aantal beperkings en uitsluitings gespesifiseer wat soos volg opgesom word:

- (1) Risiko's en beheermaatreëls met betrekking tot die ontwikkeling van stelsels.
- (2) Risiko's en beheermaatreëls met betrekking to fisiese sekuriteit.
- (3) Administratiewe beheermaatreëls en risiko's.
- (4) Gebruikerskontroles wat benodig word deur stelselprogramme anders as Oracle7 of deur ander toepassingsprogramme.
- (5) Apparaat en ander klient-/bedienertoerusting.
- (6) Vroeëre weergawes van Oracle.
- (7) Ander databasisbestuurstelsels as Oracle7.
- (8) Nutsprogramme anders as dié wat ingesluit is in Oracle7, soos Oracle CASE Tools.
- (9) Oracle7-parallelbediener.
- (10) Trusted Oracle7-bediener.

3. RESULTATE EN GEVOLGTREKKINGS

In Hoofstuk 4 is 'n vergelykingstabel (tabel 4.1) en 'n Oracle7-interne beheervraelys (tabel 4.2) ontwikkel.

GEVOLGTREKKING

Alhoewel Oracle7 oor die meeste kontrole-eienskappe beskik wat in 'n databasis-omgewing teenwoordig is, is dit belangrik om daarop te let dat die beheermaatreëls spesifiek in werking gestel moet word. Dit is eweneens belangrik om daarop te let dat Oracle7 alleen nie 'n veilige omgewing bied nie, maar dat daar kontroles oor die volgende behoort te bestaan: bedryfstelsel, netwerksagteware, toepassingsprogramme, gebruikerskontroles en kontroles met betrekking to die hardeware.

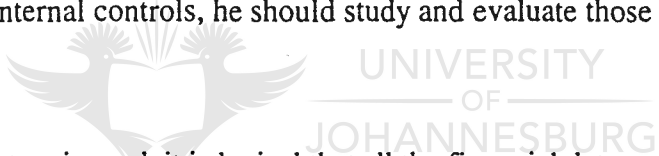
Die interne beheervraelys moet geensins gesien word as 'n vervanging van die oudit van 'n Oracle7-databasisbestuurstelsel nie. Dit is slegs ontwikkel om die ouditeur te help om 'n begrip van Oracle7 te verkry, om die ouditproses te beheer en om die risiko's wat mag bestaan te identifiseer.

Die Trusted Oracle7-bediener en Oracle7-parallelbediener omgewings is nie bestudeer nie, dit open deure vir verder navorsing met betrekking tot beheermaatreëls en risikos teenwoordig in daardie omgewings.

SYNOPSIS

1. PROBLEM DESCRIPTION AND RESEARCH OBJECTIVE

The objective of an independent audit of financial statements is to express an opinion on the fair presentation of the financial statements. The auditor should obtain sufficient audit evidence to enable him to draw conclusions to support the content of his report. The auditor should obtain an understanding of the entity's accounting system and related internal controls to assess their adequacy as a basis for the preparation of financial information and to assist in the designing of his audit procedures. If the auditor intends to rely on any internal controls, he should study and evaluate those controls.



If a database system is used, it is logical that all the financial data reside in the database. In order for an auditor to express an opinion on the financial statements, he has to determine to what extent he can rely on the integrity of the financial data that resides in the database.

The objective of this research was to identify the risks and controls present in a general database environment as well as those present in the Oracle7 database management system environment, to develop a comparison table between these environments and to develop an Oracle7 internal control questionnaire.

2. RESEARCH APPROACH

The approach was to compare the risks and controls present in a general database environment and those present in an Oracle7 database management system environment and to develop an internal control questionnaire to be used when auditing Oracle7.

3. RESEARCH METHODOLOGY

The following methodology was followed.

- (1) A literature survey was carried out on existing authoritative published works on databases in order to help identify all audit risks and controls present in a general database environment. These controls and risks were used to establish a benchmark of the controls and risks present in a database environment.
- (2) Oracle7 manuals as well as an authoritative published works on Oracle7 in a client/server environment were studied to identify risks and controls in an Oracle7 database management system environment.
- (3) A comparison table was developed in which the controls identified in an Oracle7 database management system environment were compared with the abovementioned benchmark. An Oracle7 internal control questionnaire was developed.

- (4) Based on the abovementioned comparison table and internal control questionnaire, conclusions were drawn with regard to audit risks and controls in an Oracle7 database management system.

4. CONSTRAINTS AND EXCLUSIONS

The following are not covered in this research :

- (1) Risks and controls concerning the systems development life cycle.
- (2) Risks and controls concerning physical security.
- (3) Administrative controls and risks.
- (4) User controls required by certain system or application programs.
- (5) Hardware and client/server equipment.
- (6) Earlier versions of Oracle, eg. version 6.
- (7) Database management systems other than Oracle7.
- (8) Utilities other than those included in Oracle7, such as Oracle CASE Tools.
- (9) Oracle7 Parallel Server.
- (10) Trusted Oracle7 Server.

5. RESULTS AND CONCLUSION

A comparison table (table 4.1) and an Oracle7 internal control questionnaire (table 4.2) was developed in chapter 4.

CONCLUSION

Although Oracle7 supports the majority of features available in a database environment, it is important to realise that Oracle7 features have to be enabled by the database administrator and that manual database administration control functions still have to be performed. Furthermore, Oracle7 alone does not provide a secure environment, controls over the operating system, network software, application software, user controls and controls over the hardware are necessary to provide a secure environment.

The Internal control questionnaire is not to be seen as a substitute for the audit of an Oracle7 database management system. It was developed to help the auditor to understand the Oracle7 database management system, control the audit process and identify the possible risks and controls.

The Trusted Oracle7 Server and Oracle7 Parallel Server environments were not studied, this opens doors for more research regarding risks and controls present in the abovementioned environments.

CHAPTER 1

INTRODUCTION

CONTENTS	PAGE
1.1 Background	1
1.2 Problem description and definitions	1
1.3 Objective of this research	4
1.4 Scope, limitations and exclusions	4
1.5 Methodology	6
1.6 Research approach	7
1.7 Summary of results	8
1.8 Conclusion	8



INTRODUCTION

1.1 Background

The increasing complexity of modern data processing installations is the result of a combination of the following circumstances: the lack of a visible audit trail, the number of users, the volume of data processed and stored, on-line real-time processing, user needs, distributed processing etc. Owing to the complexity of these installations, their management and control have become more difficult (Damianides, 1991:1).

"As a result of increased expectations and needs from a computer system, the trend toward on-line real-time database processing is today the norm" (Boshoff, 1990; as quoted by Damianides, 1991:1).

The increase in complexity has resulted in an increase in the inherent risks of a computer system. It is therefore inappropriate to audit "around" the computer.

1.2 Problem description and definitions

In order to understand the problems concerning database systems, one must have a definition of a database and the database management system.

1.2.1 Database

Fernandez, Summers & Wood (1981:25), define a database as -

" a collection of interrelated data items that represent the information an enterprise needs in order to carry out certain functions".

A database supports a number of applications running concurrently either on-line, in batches or both (Fernandez, et al. 1981:25).

Owing to the concentration of data and the fact that data is shared by many users, the maintenance of data integrity becomes more critical as data corruption affects many users and systems (Weber, 1982; as quoted by Johnston, 1987:35/36).

1.2.2 Database management system

Database management systems are system software that is used in the control and use of data needed by the application programs (Halper, 1985; as quoted by Johnston, 1987:5).

A database management system is software that manages the data. It interfaces with the operating system and it allows application programs to access the data controlled by the database management system (Johnston, 1987:5/6).

Formal procedures for maintaining database integrity are necessary to convince the management of an entity to support and use database systems (Everest, 1986; as quoted by Johnston, 1987:7).

1.2.3 Problem description

The objective of an independent audit of financial statements is to express an opinion on the fair presentation of the financial statements (SAICA, 1993:AU001). The auditor should obtain audit evidence that is sufficient to enable him to draw conclusions to support the content of his report (SAICA, 1986:AU204). The auditor should obtain an understanding of the entity's accounting system and related internal controls in order to assess their adequacy as a basis for the preparation of financial information and to assist in the designing of his audit procedures. If the auditor intends to place reliance on any internal controls, he should study and evaluate those controls (SAICA, 1986:AU230).

If a database system is used, it is logical that all the financial data will reside in the database. In order for an auditor to express an opinion on the financial statements, he has to determine to what extent he can rely on the integrity of the financial data that resides in the database.

1.3 Objective of this research

The objective of this research is to identify the risks and controls present in a general database environment as well as those present in the Oracle7 database management system environment, to develop a comparison table between the abovementioned environments and to develop an Oracle7 internal control questionnaire.

1.4. Scope, limitations and exclusions

1.4.1 Components

A database system consists of the following components :

The data itself

The hardware on which the data resides

The software : The database management system (Johnston, 1987:4/5).

This research will concentrate on the software issues as well as the data itself.

1.4.2 Control objectives

Control objectives have been identified by various authoritative institutions such as the South African Institute of Chartered Accountants (SAICA), the American Institute of Certified Public Accountants (AICPA), and the Canadian Institute of Chartered

Accountants (CICA) , and are internationally accepted.

The control objectives are :

Accuracy : To ensure that transactions are accurately recorded, input to the computer and processed through the accounting records (Jenkins, Cooke & Quest, 1992:26).

Integrity (maintenance) : To ensure that the integrity of data, both in transient (being manipulated) and static (having been updated) state (Damianides, 1991:5).

Validity (Authority) : To ensure that only valid transactions are processed (Jenkins, et al. 1992:26).

Continuity : To ensure that the product is operating and will continue to operate in accordance with business practice and management expectations (Damianides, 1991:5).

1.4.3 Scope

Aspects that have possible impacts on the integrity, completeness, accuracy, validity and continuity of data residing in a database will be included in this short dissertation. This includes weaknesses in the control features of the Oracle7 database management system.

1.4.4 Limitations and exclusions

The following are not covered in this research :

- * Risks and controls concerning the systems development life cycle;
- * Risks and controls concerning physical security;
- * Administration controls and risks;
- * User controls required by certain system or application programs;
- * Hardware and client/server equipment;
- * Earlier versions of Oracle;
- * Database management systems other than Oracle7;
- * Utilities other than those included in Oracle7, such as Oracle CASE Tools;
- * Oracle7 Parallel Server; and
- * Trusted Oracle7 Server.

1.5. Methodology

The following methodology will be followed.

- a) In chapter 2, a literature survey will be carried out on existing authoritative published works on databases in order to help identify all audit risks and controls present in a general database environment. The identified controls and risks will be used to establish a benchmark of the controls and risks which are present in a database environment.

- b) In chapter 3, Oracle7 manuals as well as an authoritative published work on Oracle7 in a client/server environment will be studied to identify risks and controls in an Oracle7 database management system environment.
- c) In chapter 4, a comparison table will be developed where the controls identified present in an Oracle7 database management system environment will be compared with the abovementioned benchmark. An Oracle7 internal control questionnaire will be developed.
- d) In chapter 5, based on the abovementioned comparison table and internal control questionnaire, conclusions will be drawn with regard to audit risks and controls in an Oracle7 database management system.

1.6 Research approach



The approach is to compare the risks and controls present in a general database environment and those present in an Oracle7 database management system environment and to develop an internal control questionnaire to be used when auditing Oracle7.

It is acknowledged that there may be a risk that the contents of this short dissertation may contain confidential or unpublished material. To date, no such material has been discovered which has not been acknowledged in this study.

1.7 Summary

Risks and controls present in a general environment were identified during the literature survey. I am of the opinion that the objectives of the literature survey were met and that no additional risks and controls will be identified through further literature studies.

The risks and controls present in an Oracle7 database management system as well as the items for the internal control questionnaire were identified. All information was gathered from two reliable sources : Oracle7 server manuals and the textbook Mastering Oracle7 and Client/Server Computing by Steven M Bobrowski, one of the authors of the Oracle7 server manuals.

1.8 Conclusion



The objectives set for this short dissertation have been met. A general database environment and the Oracle7 database management system were compared and an Oracle7 internal control questionnaire was developed.

Although Oracle7 supports the majority of features available in a database environment, it must be realised that Oracle7 features have to be enabled by the database administrator and that manual database administration control functions still have to be performed. Furthermore, Oracle7 alone does not provide a secure environment; controls over the operating system, network software, application software, user controls and controls over

the hardware are necessary to provide a secure environment.

The internal control questionnaire is not to be regarded as a substitute for the audit of an Oracle7 database management system. It was developed to help the auditor to understand the Oracle7 database management system, control the audit process and to identify the possible risks and controls.

The Trusted Oracle7 Server and Oracle7 Parallel Server environments were not studied, this opens doors for more research regarding risks and controls present in the abovementioned environments.



CHAPTER 2

RISKS AND CONTROLS IN A GENERAL DATABASE ENVIRONMENT

CONTENTS	PAGE
2.1 Objective of the literature survey	11
2.2 Nature of the literature survey	11
2.3 Scope, limitations and exclusions of the literature survey	12
2.4 Background	13
2.5 Analysis of references	14
2.6 Conclusion	31



RISKS AND CONTROLS IN A GENERAL DATABASE ENVIRONMENT

2.1 Objective of the literature survey

In this chapter a literature survey is conducted in order to identify and study the risks and controls present in a general database environment.

The objective of this literature survey is to obtain authoritative views on risks and controls present in a database environment, in order to establish a benchmark of the controls and risks present in a database environment.

2.2 Nature of the literature survey

In order to perform an unbiased literature survey and to ensure credibility and acceptance of findings in this study, the views examined should be authoritative and generally accepted. An individual's views may be biased as a result of personal experience, the absence of formal research or knowledge of a particular computer environment. In order to prevent such bias and the effects of specific environments, the references were restricted to those published by auditors, auditor firms, professional auditing bodies, authoritative technical books and books used by other students in their research for their master's degree in computer auditing. The main categories of authors/publishers are:



UNIVERSITY
OF
JOHANNESBURG

- (1) The American Institute of Certified Public Accountants;
- (2) The Canadian Institute of Chartered Accountants;
- (3) The McGraw-Hill series on EDP auditing; and
- (4) The Heyden series on Advances in Data Base Management.

The reasons for choosing these authors/publishers are:

- (1) they represent amongst them most of the internationally accepted views on databases; and
- (2) the emphasis on audit-related references provides better background for identifying risks and controls relevant to the auditor.

2.3 **Scope, limitations and exclusion of the literature survey**



Existing literature on databases had to be surveyed in order to develop a benchmark against which an Oracle7 database management system could be measured. In order to avoid induced exclusion of risks and controls, existing benchmarks or models were not examined.

Literature was surveyed for references to risks, exposures, threats, dangers, problems, etc. However, as very few references to risks, exposures, etc. could be found, the literature had to be surveyed for controls in a database environment on the logical assumption that a control measure implies the existence of a corresponding risk.

The following restrictions were placed on the scope of the literature survey:

- (1) Only issues which dealt with databases were included. References to risks and controls of application programs, networks and specific databases management systems were not included.
- (2) Checklists and control questionnaires were excluded from the survey because they are rarely generic - they deal with specific applications or environments. However, relevant points which were not found in the primary sources were included.
- (3) Inferior references and individual opinions were excluded as they do not represent authoritative references.
- (4) Only risks and controls relevant to the auditor were included in the survey.

2.4 Background

“ The ultimate purpose of internal controls is to reduce exposure (or increase the level of integrity) to a level acceptable to management, as determined by cost-effectiveness considerations” (Kuong, 1983:3-1).

The basic control considerations in a database environment do not change. However, the control procedures in a database environment may differ from the control procedures in a non-database environment due to data independence and the sharing of data among a large number of users (AICPA, 1983:17).

A database management system affects an auditor's means of acquiring evidence and ability to study and evaluate the internal control features (Thorne, 1980:98).

2.5 Analysis of references

The analysis is done in the following sections:

2.5.1 Control considerations

2.5.2 Control techniques

2.5.1 Control considerations



UNIVERSITY
OF
JOHANNESBURG

2.5.1.1 Access and updating

Access control features that enhance security beyond that provided in the operating system are provided in most database management systems. This security feature is optional and must therefore be activated. A risk exists that access will not be controlled if this feature is not activated (AICPA, 1983:5).

Access and updating controls should ensure that a user can only access, add, change, delete or change relationships between data elements which he has been authorized to access (AICPA, 1983:18).

In a database, the data element only exists once. This eliminates the problem of ensuring synchronized updating of independent data elements on separate files and consequently improves accuracy and timeliness. Examples in this regard include sales and stock on hand (AICPA, 1983:8).

Data sharing between user programs which are executed at the same time could result in two programs requesting updating/retrieval functions of the same data at the same time. If this is not properly controlled, it could have a negative affect on the integrity of the data. The database management system control procedures should therefore lock the data until the update processing is complete before allowing the retrieval function to access the data (AICPA, 1983:18). The lockout of data can cause the system to a halt because of the deadlock problem (Weber, 1988:523).


The overall integrity of a database could be affected to an undefinable extent if incorrect data were accepted and subsequent validation and computer generated transactions depended on that data. This will create the same conditions in a traditional system, but the error would be more difficult to isolate and correct because of the technical complexity of a database, the integration of data and the number of users that could have used the incorrect data element (AICPA, 1983:18/19).

Controls should be in place to co-ordinate cut-offs to ensure that all user requirements for period-end and historical position information are satisfied, for example, accounting versus marketing periods (AICPA, 1983:19).

2.5.1.2 Data ownership

In traditional systems, data was owned by the user of that data. A database environment is based on a data sharing concept. This means that data are available to many users. Data sharing creates concern about data integrity, security and ownership (AICPA, 1983:8).

2.5.1.3 Data redundancy and consistency of data elements



In traditional systems, the same data was stored several times in different ways in order to enable application programs to manipulate the data. The database environment facilitates a decrease in the number of times the same data is stored and also ensures that each data element in all instances is represented by the same specifications and attributes. As reconciliations to independent files are not possible, quality control of data and transaction editing before a database update are important (AICPA, 1983:8).

Because data is shared by several users, there should be control procedures to ensure consistent and accurate understanding of the correct definition of each data element and the purpose for which the data element should be used, for example, whether the price of an item is regarded as the cost or the selling price (AICPA, 1983:19).

2.5.1.4 Database structure

As data is stored through multiple logical views from a single physical representation, it is necessary to use data relationships. Data relationships necessitate the use of indexes and pointer structures to access data. Controls should be in place to ensure that the multiple logical views are protected and maintained and that they are accessible only to authorised users (AICPA, 1983:20).

Many users use and rely on a single physical representation. As physical loss in the event of fire or the like would affect many users. There should therefore be controls to protect the physical structure against improper alteration and destruction (AICPA, 1983:19/20).

2.5.1.5 Migration of control



In traditional systems, the control features are applied within the application programs and are therefore different for every system. In the database environment, certain controls may migrate from the application programs to the database management system. This has the advantage of having only one set of controls for all the application programs accessing the same data and ensures that the data from all the application programs is subject to the same controls (AICPA, 1983:22).

2.5.1.6 Reliance on the database management system and the database administrator

Failure of the database management system software will have a significant effect on the availability of data and the protection of the database in general (AICPA, 1983:20).

The database administrator is a centralised control function. This control function is responsible for the following (Thorne, 1980:99/100):

- * control of data element definitions;
- * data use identifiers;
- * data dictionaries;
- * data and storage structure definitions;
- * defining, creating, initial loading and reorganising of the database;
- * certification, recovery, restarting, protection and validation of the database;
- * monitoring the database performance;
- * correcting inefficiencies by reorganising, restructuring or removing unused data;
- * co-ordination of all database activities, training and standards; and
- * documentation of the database.

2.5.1.7 Data dictionary/ Directory system (DD/DS)

By separating the definition of data from the data itself and the programs that use the data, certain changes can be made to the definition, for example by changing the user's "view", without changing the stored data, and changes to the definition of data can be made without changing the program's source code.

The data element used to define a database is called metadata, which means data about data. The data definition language enables the database administrator to create and modify a data definition. The data definition language can also perform validation tests on the data definition to ensure that the integrity of the metadata is preserved, and also prevents unauthorised access and manipulation of the metadata. As some DD/DSs facilitate the use by several database management systems, the impact is reduced when changing from one database management system to another. On the other hand, this introduces control problems in that two separate definitions are maintained, one by the DD/DS and the other by the database management system.

The DD/DS can be either passive or active. A passive DD/DS is not used by the database management system to gain access and manipulate data, therefore the data definition maintained by the DD/DS and the data definition maintained by the database management system are separate. A control problem arises in this case because two different data definitions can occur. An active DD/DS is one that is used by the database management system, and it is therefore more likely that a consistent data definition will exist.

Controls such as security, backup and recovery controls have to be established over the data definition and the DD/DS. If the data definition is lost or destroyed, the operations of the entity will be severely affected, and if the DD/DS is corrupted the database definition can also be corrupted, which will result in the corruption of the database (Weber, 1988:224-228).

The data dictionary is a documentation tool and provides the following features (Thorne, 1980:100) :

- * standard definitions for data elements, segments and databases;
- * narrative and technical descriptions of the data;
- * information about security, edit considerations, structure and application usage; and
- * description of controls related to each data element.

5.1.2.8 Reorganisation of the database

When the database is reorganised, the data becomes extremely vulnerable because the data can be modified, deleted or lost through ineffective reorganisation procedures (Perry, 1983:335).

Reorganisation is required in the following cases:

- * to accommodate a new application (Perry, 1983:201);
- * response times (Weber, 1988:534);
- * resource consumption (Weber, 1988:534); and
- * changing needs of users (Perry, 1983:346).

Reorganisation involves establishing new access paths via indexes, pointers clearing out overflow areas and assigning data to faster storage devices (Weber, 1988:534) and adding new capabilities and sub-schemas (Perry, 1983:354).

2.5.2 Control techniques



2.5.2.1 Access and update controls

Access to the database can be restricted by the use of passwords. The passwords can restrict individuals, organisational units, terminals, programs, transactions, and the use of functions such as read, update, modify, add or delete. If passwords are logically related to terminals, programs, and sub-schemas, only authorized users will be able to enter, amend or delete data. For passwords to be effective, procedures should be in place for changing passwords, maintaining secrecy and investigating security violations (AICPA, 1983:27).

The person assigning user access should not be the same person as the person responsible for the implementation of access control (Weber, 1988:217).

The password file should be well protected (Perry, 1983:330).

The sub-schema can be used to restrict access to the database and prevent unauthorised transactions access by describing an application programs's logical view in the sub-schema and limiting the programs's access to and operations on the data. This technique is less effective when the sub-schema is embedded in the data manipulation language (AICPA, 1983:28).

Sub-schema controls are the application controls implemented by the user over the integrity of the sub-schema and include controls such as segment counts, control totals etc. (Perry, 1983:358).

Utility programs should be controlled by the database administrator, as these programs can be used to gain unauthorised access to the database and database management system library. The programs can execute functions such as listing the database descriptions and testing the data manipulation language (Thorne, 1980:103).

As utilities and commands are provided by the database management system, these utilities and commands provide capabilities to modify data and to modify improper operations of the database management system. These utilities and commands should be protected through the use of passwords (Perry, 1983:335).

All changes to the database management system should be logged to ensure that only required data items are accessed by the application programs (Thorne, 1980:103).

A log of programs executed and program specifications used should be maintained to ensure that only authorized program specification were used to access the data. This will also reveal temporary changes to the database management system library for one execution of a program (Thorne, 1980:103).

Passwords should be used to control the reorganisation process. The password should not be used until management authorizes the reorganisation (Perry, 1983:335). There should be control over the utilities used for the reorganisation process (Perry, 1983:346). Statistics will provide sufficient information to permit the integrity of the database to be verified after reorganisation (Perry, 1983:354).

As the master terminal is used to execute privileged database management system commands, it is the only terminal from which these commands can be issued (Perry, 1983:336).

A terminal should be automatically shut down after a predetermined number of unsuccessful access attempts or when an invalid process has occurred (Perry, 1983:337).

Quality control ensures the accuracy, completeness and consistency of data maintained in the database. This control includes validation of input data, batch control over data in transit, check digit etc. (Weber, 1988:218).

Input validation checks should be used and these checks should be documented in the data dictionary. These checks include checks such as limit checks, reasonableness checks, range checks, comparisons to values in tables etc. (Perry, 1983:359).

2.5.2.2 Database management controls

The ownership of a data element should be assigned to a single user. That user is responsible for defining the logical meaning, access and security rules such as who can use the data element and what functions can be performed (AICPA, 1983:33).

Administration, amendment, creation of schemas and sub-schemas should be centrally controlled; this will help to prevent unauthorised changes, sharing, and creation of schemas and sub-schemas (AICPA, 1983:34).


Adequate segregation of duties should be maintained for design, implementation and operation of the database (AICPA, 1983:34/35).

The database administrator should review the database management system log and the operating system log in order to determine whether a database, while under the control of the database management system, was accessed by authorized programs only (Thorne, 1980:102).

Changes to the database management system can have an impact on the control and integrity of data. Any changes to a database management system should therefore be approved by the database administrator, as he is the only person with complete knowledge of the database system (Thorne, 1980:103).

Maintenance-related utility programs should be controlled by the database administrator, since these programs are extremely powerful (Thorne, 1980: 102/103).

As the database administrator has a complete knowledge of the database management system library and all the application programs, he can easily gain access to and manipulate data items. Therefore the database administrator should not have access to initiate transactions or operate equipment (Thorne, 1980:103).



Manual logs should be maintained of activities such as requests for access to the database or the use of servicing tools by the database administrator, as well as the database administrator or operations manager's requests for object codes and requests for source code listings by the programming manager. Machine logs should record similar activities as above-mentioned manual logs to provide an independent check on manual logs (Weber, 1988:231).

Data elements are used by more than one user; therefore the removal of a data element will affect many users. To reduce the risk of removing data elements which are still being used, a formal request form for removal should be circulated and every party should approve the suggested removal (Perry, 1983:321).

Database malfunctions or database errors should be reported. The report should identify the malfunction and its cause and recommend a course of action to be taken (Perry, 1983:330).

In order to maintain an accounting audit trail in an application system, the database subsystem has to perform certain functions:

- (1) A unique time stamp should be attached to all transactions applied against the database definition. Such a time stamp confirms that the transactions reached the database definition or the database, and identifies the position of a transaction in the time series of events that has occurred to a data item in the database definition or the database.
- (2) It must attach before-and-after images of a data item against which a transaction is applied to the audit trail entry for the transaction. The purpose of the before-and-after-images is that they facilitate enquiries about the effect of a transaction on the database definition and provide redundancy for the time stamp in that a fraudulent deletion of an audit trail entry or an alteration of a time stamp can be detected by the mismatch between the after-image and the before-image of the subsequent transaction (Weber, 1988:531/532).


The operations audit trail maintains the chronology of resources consumption for each event that occurs to the database definition or the database (Weber, 1988:533).

2.5.2.3 Controls over application programs

Data manipulation language call verbs, that are not necessary for certain application programs to function should be restricted. These call verbs include commands such as insert, replace or delete (Thorne, 1980:104).

Standards should be specified for the program action that should be taken for each database management system return code. Programs should also provide for disposition of all database management system return codes to ensure that processing errors are detected (Thorne, 1980:104).

2.5.2.4 Definition and DD/DS control

The logo of the University of Johannesburg, featuring two stylized birds facing each other with a book between them, and the text 'UNIVERSITY OF JOHANNESBURG' in a serif font.

Definition control ensures correspondence between the database and its definition at all times. This implies that neither a program nor a procedure should be able to destroy this correspondence; for example, application programs and end-user development programs are not allowed to manipulate pointers in the database in case an access path is destroyed (Weber, 1988:217).

Duplicate copies of the DD/DS and the database definition should be stored off-site, and a log of all changes to the database definition should be kept (Weber, 1988:229).

2.5.2.5 Concurrency control

A problem arises when two programs try to update the same data at the same time. If this happens the database may end up in an inconsistent state, with the result that the database integrity is impaired (Weber, 1988:218).

Two-phase locking handles a transaction as follows. First, before a transaction can read a data item it has to own a “Read lock”, and before a transaction can write to a data item it has to own a “Write lock”. This rule provides a partial strategy to prevent deadlock. Second, different transactions are not allowed to own conflicting locks simultaneously, which means that two transactions are not allowed to own a “read lock” and a “write lock” or two “write locks” on the same data item. Third, when a transaction releases ownership of a lock it cannot obtain additional locks. A transaction should commit the database changes before it releases the lock (Weber, 1988:527).

Two-phase schedulers are constructed to process and to enforce locking protocols in a distributed database. According to Weber (1988:528), at least three strategies are available to ensure control:

- * A scheduler for each data item is placed at the location where the data item is stored;
- * One version of the data item and its associated scheduler is designated as the primary copy; and
- * Schedulers are located at a single centralised site.

2.5.2.6 Existence controls

These controls must ensure the capability of restoring the database to its state at the point of failure in as little time as possible. This implies that a recovery and backup strategy must be in place.

Backup strategies include maintaining a prior version of the database and a log of subsequent changes to the database.

Recovery strategies are divided into two forms. First, the current state of a database must be restored after the failure of a physical device. This will result in applying rollforward procedures using a prior version of a dump of the database and a log of subsequent transactions. Second, a prior state of the database must be restored because of the invalid state of the current database resulting from an erroneous program update. This will involve rollback procedures using the current state of the database and a log of transactions (Weber, 1988:534-536).

As dumping involves copying the whole or a portion of the database to a backup medium, a physical or logical dump can be executed. Dumping is only a partial backup strategy and as it only restores the database to a valid state prior to the time of failure, it will still be necessary to maintain logs of transactions or changed images of records on the database (Weber, 1988:540/541).

Logging is the recording of changes to the database caused by transactions, an image of the record changed by an update or the change parameters resulting from an update. The four types of logging strategies are :

- * logging input transactions, which is a strategy involving the reprocessing of update transactions from the time of the last dump up until the time the database was damaged. Selective recovery is possible with this strategy;
- * logging before-images, which is a strategy used to facilitate rollback of the database. Before-images can also be used for rollforward;
- * logging after-images, which is a strategy used to facilitate rollforward of the database. The use of after-images for rollback should be avoided; and
- * logging change parameters.



The strategy chosen would depend on the requirements of the application. It does not matter which strategy is chosen, but it is important that the log file must never be buffered or blocked because the log file may not be current if the contents of the buffer have not been written to the log in case of a system crash (Weber, 1988:541-546).

Although database recovery can be achieved in many different ways, it is important that recovery lockout control should prevent files from being updated while the recovery is in progress. However, inquiries during that period may be allowed (Perry, 1983:334).

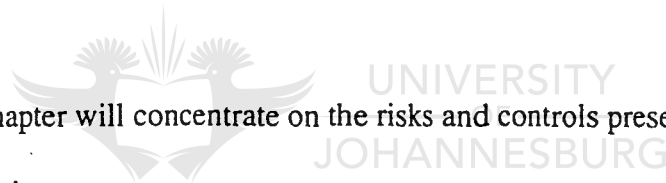
A recovery audit trail should be used in pinpointing accountability for improper acts, discovering potential weaknesses and offering solutions (Perry, 1983:358).

Backup and recovery procedures should be tested (Thorne, 1980:104).

2.6 Conclusion

Risks and controls present in a general environment were identified during the literature survey. I am of the opinion that the objectives of the literature survey in this chapter were met and that no additional risks and controls will be identified through further literature studies.

The following chapter will concentrate on the risks and controls present in the Oracle7 client / server environment.



CHAPTER 3

RISKS AND CONTROLS IN AN ORACLE7 (CLIENT / SERVER) ENVIRONMENT

CONTENTS	PAGE
3.1 Introduction	33
3.2 Background	34
3.3 Database structure	34
3.4 Oracle system architecture	38
3.5 Management of data	40
3.6 Access controls	52
3.7 Database backup	58
3.8 Recovery	63
3.9 Distributed databases	65
3.10 Conclusion	67



RISKS AND CONTROLS IN A ORACLE7 (CLIENT / SERVER) ENVIRONMENT

3.1 Introduction

A client/server system is a mix of independently developed and manufactured hardware and software and is therefore less reliable than a centrally managed mainframe or micro-computer. It is important to choose the correct applications to run on the client/server system. The client/ server system is an important part of the corporate strategy, but it is not the right choice for every application (Bobrowski, 1994:15-17).

The Oracle7 database management system will be studied in this chapter, using the Oracle7 server manuals as well as the textbook Mastering Oracle7 and Client/Server Computing by Steven M Bobrowski, one of the writers of the Oracle7 server manuals as a source.

The objectives of this study are to gather information to compare the risks and controls present in Oracle7 with general risks and controls present in a database environment, and to identify items to be included in the Oracle7 internal control questionnaire for the auditor to use when auditing an Oracle7 database management system.

3.2 Background

Oracle7 can be plugged into a client/server equation. Oracle7 is not only a database management system, but also a database server for client/server database computing. It supports the major operating systems for both clients and servers, including MS DOS, NetWare, UnixWare and OS/2. Oracle7 networking software, SQL*NET, supports all major communication protocols such as TCP/IP, SPX/IPX, Named Pipes and DECNet. Oracle7 is based on a relational database model (Bobrowski, 1994:17).

3.3 Database structure

The database structure consists of a logical as well as a physical structure. The physical structure can therefore be managed without affecting the logical structure (Armstrong, Bobrowski, Closkey, Linden & Pratt, 1993:Chapter 1).

3.3.1 Physical database structure

The physical structure consists of the following types of files : data files, redo log files and control files (Armstrong, et al. 1993:Chapter 1). The Oracle7 physical structure is independent and hidden from the end user's logical view (Bobrowski, 1994:27).

3.3.2 Logical database structure

The logical structure is determined by : a) tablespaces and b) database schema objects such as tables, views, sequences, indexes, stored procedures, synonyms and database links. No relationship exists between a tablespace and a schema (Armstrong, et al. 1993: Chapter 1).

3.3.3 Tablespaces

A tablespace is a partition or logical area of storage in a database that directly corresponds to one or more physical data files (Bobrowski, 1994:30).

Oracle7 also makes provision for read-only tablespaces that cannot be modified. It is therefore not necessary to make repeated backups of these tablespaces, and they need not be recovered. After a tablespace has been made read-only, it should be backed up. After a read-only tablespace has been changed to read-write, backups should be resumed. Data files cannot be added to read-only tablespaces; they should first be changed to writable (Frazzini, Hiltner & Pratt, 1994:Chapter 3).

Every Oracle7 database will have at least one SYSTEM tablespace. When a database is created, the initial data files for the database are specified, these files are the physical storage for the system table space. The SYSTEM tablespace is used to hold the data dictionary (Bobrowski, 1994:30). The data dictionary must always be available to Oracle7, therefore the SYSTEM tablespace must always be kept online (Armstrong, et al. 1993:Chapter 1).

As it is likely that there will be a number client applications that access the database, it is best to create one or more tablespaces to hold each application's data separate from the data dictionary and the data of other applications (Bobrowski, 1994:30). Using tablespaces allows -

- * administrators to control the availability of a database's data on a tablespace-by-tablespace basis. An application can be made inaccessible by taking the application's tablespace offline (Bobrowski, 1994:30). A tablespace that contains active rollback segments cannot be taken offline until those rollback segments are inactive (Armstrong, et al. 1993:Chapter 1);
- * administrators to back up a database at the tablespace level (Bobrowski, 1994:30); and
- * each application's tablespace to be placed on different disks of a database, so that they will not contend with each other for disk access and space (Bobrowski, 1994:30).

3.3.4 Schema objects

- * **Tables:** The tables hold the user-accessible data. The data is stored in rows (records) and columns (fields). Each table has its unique table name. Each column has a column name, data type and width (Armstrong, et al. 1993:Chapter 1);
- * **View :** A view is a virtual table, deriving its data from base tables (Bobrowski, 1994:33). A query is used to define a view (Armstrong, et al. 1993:Chapter 1). When using views, users see the same data that is in the database tables, but with a different angle. Views can be used to increase security by limiting access to specific sets of rows and columns of a table. They can also be used to hide complicated queries from users. Users will only issue simple queries against the view, and the view will take care of the complicated query. Oracle7 does not allow manipulation of views that have a defining query with a group operator or a join (Bobrowski, 1994:33/34). Data is presented in a different perspective from that of the base table (Armstrong, et al. 1993:Chapter 1);
- * **Sequences:** These generate unique numbers for numeric columns. Sequence numbers are independent of the tables, therefore the same sequence can be used for different tables (Armstrong, et al. 1993: Chapter 1);

- * **Synonyms:** A synonym is an direct reference to an object. Synonyms can be used as security measures by masking the name and owner of an object and providing location transparency for remote objects of a distributed database. Synonyms can either be public or private: a public synonym is accessible to every user in the database, whereas a private synonym is contained in the schema of a specific user and of the user's grantees (Armstrong, et al. 1993:Chapter1); and
- * **Database links:**Database links are the paths from one database to another (Armstrong, et al. 1993:Chapter 1).

3.4 Oracle system architecture



The System Global Area (SGA)

This is an allocated shared memory region that contains data and control information for an instance. An SGA and the Oracle background processes constitute an instance (Armstrong, et al. 1993:Chapter 1).

Database Writer (DBWR)

This background process writes modified blocks from the database buffer cache to the data files (Armstrong, et al. 1993:Chapter 1).

The Log Writer (LGWR)

This background process writes redo log entries to disk (Armstrong, et al. 1993:Chapter1). The Log Writer stamps the headers in all the data files with the most recent checkpoint's internal information (Bobrowski, 1994:109).

Checkpoint (CKPT)

The Database Writer (DBWR) performs checkpoints to ensure that all modified information in memory is physically stored on disk. The checkpoint indicates how much of a transaction log needs to be applied in case of a crash (Bobrowski, 1994:109).

System Monitor (SMON)



UNIVERSITY
OF
JOHANNESBURG

This background process performs instance recovery at instance startup (Armstrong, et al. 1993:Chapter 1).

Process Monitor (PMON)

This background process is responsible for cleaning up a front-end server process after a crash. The PMON rolls back a user's dead transaction and releases the transaction's locks to enable other transactions to get access (Bobrowski, 1994:109).

Archiver (ARCH)

This background process copies the online redo log files to archive storage (Armstrong, et al. 1993:Chapter 1).

Recoverer (RECO)

This background process is used to resolve distributed transactions that are not committed due to a network or system failure (Armstrong, et al. 1993:Chapter 1).

3.5 Management of data

3.5.1 Data Dictionary



The data dictionary is a read-only set of tables. The following information is stored in the data dictionary:

- * names of Oracle users;
- * privileges and roles granted;
- * names of schema objects;
- * integrity constraints;
- * default values for columns;
- * space allocated and usage by objects;

- * auditing; and
- * other general information.

Oracle reads the data dictionary during database operation to ensure that objects exist and that users have access to them. The dictionary is also continuously updated to reflect changes in the database structure, auditing, grants and data. The data in the data dictionary should not be deleted or altered.

Base tables store the information about the database. Only Oracle writes to and reads these tables; most of the data is stored in a cryptic format.

User accessible views summarize and display information in the base tables. Users should only be given access to views and not to base tables. Access to certain types of views can be limited. The following types of views exist in Oracle : USER, ALL and DBA. Access to the DBA view should only be granted to the database administrator. A user with a SELECT ANY TABLE privilege can query the DBA views. Synonyms cannot be created for DBA views.

All base tables as well as the user views are owned by the Oracle SYS user, therefore no user should have access to the SYS schema (Armstrong, et al. 1993:Chapter 8).

3.5.2 Data integrity

3.5.2.1 General

If a relational database has data integrity it means that all of the data in the database is valid according to a set of rules (Bobrowski, 1994:22). The relational database model describes the following set of rules to guarantee data integrity (Bobrowski, 1994:34-37):

- * Domain integrity : ensures that a value in a column is a member of the column's domain. A row cannot be in a table unless each of the column values of that row is a member of the domains of the corresponding column;
- * Entity integrity : every row in a table must be unique (no duplicating). In order to ensure entity integrity, a column in the table must be designated as the primary key. Each row in a table must contain a unique primary key. A table can have only one primary key. Another non-primary key column can be designated as a unique key, but a table cannot have duplicate unique keys; and
- * Referential integrity : defines the relationships among different columns and tables. The values in one column or set of columns must refer to or match the values in a related column or set of columns. The dependent column is called a foreign key and the referenced column is called the parent key. The parent key must be a primary or unique key. When the parent and the foreign key are in the same table, it is called self-referential integrity.

3.5.2.2 Data integrity with Oracle7

One can define standard data integrity rules of the relational model within Oracle7 by using integrity constraints. When creating a table in Oracle7, one can specify integrity rules with command options (Bobrowski, 1994:22).

The following features are used to enforce data integrity at the database server:

- * Data types are used to enforce domain integrity. A record cannot be in a table unless the data for each column is the correct type. There are some data types that allow further limitations of the domain of a column, for example:
 - NUMBER data types can define the precision (total number of digits) and scale (number of digits to left or right of a decimal place). Oracle7 does not allow a row in a table if the row has a number in a column that exceeds the precision of the column; and
 - CHARACTER data types allow domain integrity control by specifying the maximum length of text strings for the column. Oracle7 does not allow a row in a table if the row has a text string in a column that exceeds the maximum length of a text string in that column (Bobrowski, 1994:39/40).

* Integrity constraints. An integrity constraint is a statement about a table's data that is always true. If an integrity constraint is created and the existing data does not conform to the constraint, the constraint cannot be enforced. When a DML statement violates the constraint after it has been defined, the statement will be rolled back and an error will be returned. The integrity constraint is part of the definition of a table and is stored in the data dictionary ; therefore any data entered by an application must adhere to the same integrity constraints associated with a table. Default column values are subjected to all integrity constraint checks (Armstrong, et al. 1993:Chapter 1&7). Oracle7 includes different kinds of integrity constraints:

- absent values (NULL) can be eliminated from the domain of a column by declaring a NOT NULL integrity constraint with the column's definition (Bobrowski, 1994:40-42);
- a CHECK integrity constraint can be used to enforce a custom domain integrity expression, for example valid provincial codes (Bobrowski, 1994:40-42);
- a PRIMARY KEY integrity constraint can be used to define a table's primary key and enforce entity integrity (Bobrowski, 1994:40-42);
- a UNIQUE integrity constraint can be used to enforce uniqueness of customer names (Bobrowski, 1994:40-42); and
- a FOREIGN KEY integrity constraint can be used to define a foreign key in a table and enforce referential integrity. Referential integrity defines referential action when operating on the parent key; for example, delete cascade referential action, restrict referential action, set to Null, and set

to default (Bobrowski, 1994:40-42). Referential integrity cannot be enforced when the parent key and the child key are on different nodes of a distributed database. Triggers will have to be used to enforce referential integrity in a distributed database (Armstrong, et al. 1993:Chapter 1).

* Integrity constraints can be used to enforce a large number of data integrity rules, but there will always be custom business rules that cannot be enforced with integrity constraints. Business rules can be enforced by using stored procedures and triggers;

- Stored Procedures is a compiled collection of SQL statements, flow-of-control statements, variable declarations and assignment operators that are created and stored in a database. When a user performs an operation using a procedure, Oracle7 forces the user to touch data in a prescribed manner; for example, when a user deletes a customer record from the CUSTOMER table, the user should log the customer ID and name in a history table; and

- A trigger is a stored procedure that Oracle7 automatically fires under the appropriate conditions, eg. when INSERT, UPDATE, or DELETE statements are issued (Bobrowski, 1994:43-46). Triggers can be used to automate data generation, audit data modifications, enforce complex integrity constraints and customize complex security authorizations. When a trigger is created it does not check the existing data already loaded onto a table (Armstrong, et al. 1993:Chapter 1). A trigger only enforces a constraint at the time that the data changes. Triggers are stored separately from their associated tables (Armstrong, et al.

1993:Chapter15). The firing of triggers can be restricted by using Boolean conditions to specify an optional WHEN clause of the 'create trigger' command (Bobrowski, 1994:147). The Boolean expression must be TRUE for the trigger to fire (Armstrong, et al. 1993:Chapter 15).

3.5.3 Managing Data Concurrency

Oracle7 uses two levels of locking exclusive and shared locks to prevent destructive interference. All the necessary locks are obtained automatically. When a user attempts to operate on some data, Oracle7 always acquires a share lock if there is no possibility of destructive interference. A share lock allows other transaction to acquire share locks on the same data. When a share lock leaves open the possibility of destructive interference, an exclusive lock must be acquired.

When different transactions want to update two different rows in the same table, shared locks will be acquired. When two different transactions want to update the same row in a table, an exclusive lock will be acquired on a first-come-first-serve basis in a serial fashion.

Oracle7 therefore locks data on a row-for-row basis; other systems use page-level locking (Bobrowski, 1994:47-49).

Different locks can be obtained depending on the resource being locked and the operation being performed. The different kinds of locks are (Armstrong, et al. 1993:Chapter 10):

- * Data locks;
- * Dictionary locks;
- * Internal locks - protects database structures; and
- * Distributed locks.

The option to manually lock data also exists (Armstrong, et al. 1993:Chapter 10).

A transaction does not acquire any locks for any type of query; therefore two transactions can query the same data at the same time in the same way. A query can never block an update, and vice versa.

Although Oracle7 does not use locks for queries, accurate results are obtained because Oracle7 has a multi-versioning mechanism: for every query a timepoint-base version of the data is returned. Oracle7 uses rollback segments for its multi-version mechanism (Bobrowski, 1994:49-51).

Deadlock situations are automatically detected and are resolved by rolling back one of the statements involved (Armstrong, et al. 1993:Chapter 10).

3.5.4 Auditing

Auditing can be performed at three different levels (Armstrong, et al. 1993:Chapter 1):

- * statement auditing - auditing of specific SQL statements;
- * privilege auditing - auditing the use of system privileges; and
- * object auditing - auditing access to specific schema objects.

The results of the audits are recorded and stored in a data dictionary table named the audit trail or in an operating system audit trail.

No audit records will be generated by SYS user sessions. Auditing is site autonomous, and the local database can therefore not audit actions that take place in a remote database.

Oracle will always audit certain database actions, whether auditing is enabled or not. These actions are instance startup, instance shutdown and connections to the database as INTERNAL (Armstrong, et al. 1993:Chapter 19).

3.5.5 Deleting a database

Oracle 7 does not provide a “delete database” command. You have to identify the files that make up the database and delete them by using the commands of the operating system (Bobrowski, 1994:157).

3.5.6 Administration Accounts

Oracle 7 creates a SYS user when a database is created. The SYS user is the owner of the data dictionary tables.

When a user connects to the database as the SYS user, the potential exist of adversely affecting the data dictionary. The default password for the SYSTEM and SYS users should be changed immediately after the creation of a database (Bobrowski, 1994:162/163).

Users connecting as INTERNAL can execute database startup and shutdown. Such users can be required to connect with a unique user name and password. This will allow auditing connections by user ID (Frazzini, et al. 1994:Chapter 5).

Privileged users connecting to the SYS schema to perform database administration tasks can be required to use a unique username and password. Privileged connections are always audited, regardless whether auditing is enabled or not (Frazzini, et al. 1994: Chapter 5).

Passwords are stored in a external encrypted password file. The use of secure password files permits remote database administration over non-secure networks (Frazzini, et al. 1994:Chapter 5).

The REMOTE_LOGIN_PASSWORDFILE can be set to: NONE, EXCLUSIVE, or SHARED

NONE - The user is authenticated by the operating system; only authentication for local and secure connections is permitted.

EXCLUSIVE - The password/ user ID combination is checked against an external password file. The connection is noted in the audit trail. The use of secure password files permits connection to succeed over non-secure remote connections.

SHARED - All privileged users must connect as the SYS user with the appropriate password. Remote database administration over non-secure connection is allowed.

Only users connected as SYSDBA or INTERNAL can grant SYSDBA and SYSOPER system privileges. These privileges cannot be granted as roles (Frazzini, et al. 1994: Chapter 5).

Password files can be created by using the ORAPWD utility or the operating system. It is of critical importance to protect the password file as well as where it is located (Frazzini, et al. 1994:Chapter 5).

3.5.7 User Accounts

If a default tablespace and a temporary tablespace for a user are not specified, the default tablespace will be the SYSTEM tablespace (Bobrowski, 1994:266/267).

When a schema object is created by a user without specifying a tablespace, the object is placed in the user's default tablespace (Armstrong, et al. 1993:Chapter 17).

The temporary tablespace is used when a user executes a SQL statement that requires the creation of a temporary segment (Armstrong, et al. 1993:Chapter 17).

3.5.8 SQL*Loader and SQL*DBA

Oracle 7 includes the SQL*LOADER that can be used to load data onto an Oracle database from a non-Oracle source (Bobrowski, 1994:178).

When the SQL*Loader is used, it automatically disables CHECK and referential integrity constraints and INSERT triggers. Disabled integrity constraints must be enabled manually. Oracle7 cannot enable a constraint in a table if a row violates the integrity rule. However, when a trigger is re-enabled Oracle7 does not check the rule against all table rows before enabling the trigger (Bobrowski, 1994:184-186).

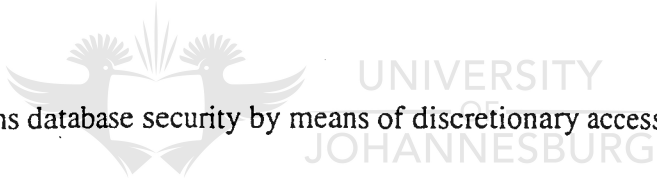
SQL*DBA is the primary utility used to administer an Oracle7 database system. When Oracle7 is installed, SQL*DBA is automatically installed too. As this utility has extremely powerful functions, access to it should be limited to the database administrator (Bobrowski, 1994:164-176).

3.5.9 Messages and Codes

Oracle7 includes messages as warnings and indications of errors that have occurred. The probable cause of each error is explained in the Messages and Codes manual. If the message is a warning or indicates an error, the message listing usually indicates a corrective action. The types of messages are classified and listed in Armstrong, Frazzini, Portfolio, Quigley & Smith, 1993: Server Messages and Codes Manual (R).

3.6 Access Controls

3.6.1 Access Controls (excluding SYS and SYSTEM users)



Oracle7 maintains database security by means of discretionary access controls based on privileges. In order to give a user general access to the database, a user name must be created to register the user. The username must also have a corresponding password and both must be given to connect to the database (Bobrowski, 1994:53).

For each user created, a schema with the same name is created containing database objects. The user has access to all objects contained in his schema. Access rights are controlled by a user's security domain. A user's security domain determines the user's privileges and roles, tablespace quotas and system resource limits (Armstrong, et al. 1993:Chapter 1&17).

Oracle7 permits either password or operating system authentication. Using operating system authentication while running Oracle7 in a client/server environment makes security violations possible, because it makes it easy for a person on a client workstation to impersonate a real database user. It is therefore much more secure to use password authentication (Bobrowski, 1994:265/266).

The user passwords are stored in the data dictionary in an encrypted format (Armstrong, et al. 1993:Chapter 17). An Oracle7 client can be configured to encrypt a password before sending it across the network (Bobrowski, 1994:271). The encryption method used encrypts to a different value with every connection attempt. Oracle encrypts a password whenever a connection is attempted. If auditing is enabled and a connection fails, the failure is noted in the audit log. To prevent a user forcing Oracle from reattempting a connection with an unencrypted password, the following values must be set to TRUE: ORA_ENCRYPT_LOGIN and DBLINK_ENCRYPT_LOGIN. If set to FALSE, the second connection attempt will attempt to use an unencrypted password (Frazzini, et al. 1994:Chapter 5).

Oracle7 includes the feature of terminating idle user sessions after a certain timespan between calls for a session. The current transaction will be rolled back and the session terminated, and the next call will receive an error (Bobrowski, 1994:279).

A user's profile is a set of system resource limits. A new database user cannot connect to the database until the user's privileges have been granted (Bobrowski, 1994:266/267).

Oracle is usually licenced for use by a maximum number of users (Armstrong, et al. 1993:Chapter 17).

Operations of users not assigned a specific profile will be automatically limited to the Oracle 7 DEFAULT profile (Bobrowski, 1994:282).

The availability of data in an open database can be controlled on a tablespace-by-tablespace basis. If a tablespace is online, the tables in the tablespace can be queried and modified by privileged users; when it is offline, no one can use the data in that tablespace (Bobrowski, 1994:60).

Procedures, functions, packages and triggers are all stored in the Oracle7 SYSTEM tablespace. There is no option to physically locate them elsewhere. It is therefore important that there should be enough space on the SYSTEM tablespace as well as restrictions on access to the SYSTEM tablespace (Bobrowski, 1994:293).


The Oracle 7 data dictionary includes a number of views that can be used to reveal the availability and structural information about the database. Access to the dictionary and these views should therefore be restricted (Bobrowski, 1994:216).

3.6.2 Privileges

The right to execute a particular type of SQL statement is called a privilege (Armstrong, et al. 1993:Chapter 1).

A user's access is controlled by the user's database system privileges and database object privileges. The user's privileges should correspond to the application he is currently running (Bobrowski, 1994:271).

By the granting and revoking of different privileges, all database operations and access can be controlled, eg. which users can create tables and views, which users can create and modify tablespaces and which users can read and modify the various tables and views in the database (Bobrowski, 1994:55). Users granted a system privilege with the ADMIN OPTION or GRANT ANY PRIVILEGE can grant or revoke a system privilege. The owner of an object can grant any object privilege to any other user or role, and the grantee can further grant object privileges if the grant includes the GRANT OPTION (Armstrong, et al. 1993:Chapter 18). There are two broad types of privileges:

- 
- (1) System privileges : The database system privilege controls access to and the performing of administrative functions on a database-wide basis. This is a very powerful privilege. The granting of these privileges should be considered carefully, and they should only be granted to administrators. Examples of system privileges are (Bobrowski, 1994:55+272):

ALTER DATABASE - alters physical structure.

DROP TABLESPACE - can drop any tablespace in the database except the SYSTEM tablespace.

SELECT ANY TABLE - can query any table in the tablespace.

- (2) Database object privileges : allows a user to perform a particular action on a specific object (Armstrong, et al. 1993:Chapter 1). An object is a table, view, role, procedure or user. There are INSERT, UPDATE, and DELETE object privileges for each table (Bobrowski, 1994:55).

Application users normally only need object privileges to access objects associated with an application and should never have more than a selected few system privileges, such as CREATE SESSION.

The CONNECT role is intended for end users, but this role includes some privileges that should not be included in a end user role, such as the CREATE TABLE system privilege (Bobrowski, 1994:279).



Unlimited access to all tablespaces in a database can be granted by using the UNLIMITED TABLESPACE system privilege. This privilege should only be granted to application developers in a test database. Oracle7 does not allow this system privilege to be granted as a role - it can only be granted directly (Bobrowski, 1994:281).

Privileges can be granted to users explicitly or to roles (Armstrong, et al. 1993: Chapter1).

3.6.3 Roles

Roles can also be used to manage access control. A role is a collection of related privileges that can be granted collectively to database users and to other roles. Roles can also be used to change the privilege domain of users as they use different applications by using the SET ROLE statement (Bobrowski, 1994:55/56).

Roles can be granted system or object privileges. By enabling a role that contains other roles, all indirectly granted roles are enabled (Armstrong, et al. 1993:Chapter 18).

There is a default database administration role named DBA. This role has every system privilege, which means that a user with the DBA role can do anything within the database (Bobrowski, 1994:273).



Unauthorized use of privileges granted to roles can be prevented by using passwords to protect the role (Armstrong, et al. 1993:Chapter 1).

A user who is granted the system privilege GRANT ANY ROLE or a user granted a role with the ADMIN OPTION can grant or revoke a role (Armstrong, et al. 1993:Chapter 18).

3.7 Database backup

Documentation of each backup should be kept. This documentation should include information such as date of backup, database name and highest log sequence in the transaction log (Bobrowski, 1994:350/351).

3.7.1 Transaction (Redo) log

Oracle7 records all changes, committed as well as uncommitted, in a database to the transaction log (redo log). When a transaction is committed, Oracle7 immediately writes a record to the log stating that the transaction and its changes are now permanent. When the current database instance crashes, there might be some committed transactions that have not been written to the data files. At the startup of the next instance, Oracle7 automatically performs crash recovery to restore the database to its state just after the last transaction that was committed before the failure.

The transaction log contains two or more groups of fixed-size log files or members. After transactions have filled up the log space in the first group, Oracle7 switches to the next group and automatically starts to archive the filled group in parallel, without interfering with the continuous logging. Oracle7 increments the log sequence number of the database in order to keep track of current and archived log groups. Before overwriting a filled log group, Oracle7 ensures that the log group has been archived. The archived groups build a permanent offline sequential transaction log.

The archived log contains a sequence of backup files that correspond to the log groups as they were filled. The archived log is necessary for complete database recovery from a media failure.

Media recovery is enabled when log groups are required to be archived before they can be reused. Media recovery can be disabled; in that case, Oracle7 will not archive the transaction log. As media recovery is disabled when a new database is created, the risk exists that the media recovery is not enabled (Bobrowski, 1994:339-344).

The transaction log has its own protective features, eg mirroring the log groups of the database. Mirroring involves creating multiple members for a log group and physically placing them on different disks. When Oracle7 logs transactions to a mirrored group, it writes the changes in parallel to all members of the group (Bobrowski, 1994:61/62).

The mirrored online redo log should be symmetrical, in other words all groups should contain the same number of members in order to protect the redo log against a single point failure (Armstrong, et al. 1993:Chapter 23).

3.7.2 Rollback

“Undo” information is kept in a rollback segment. The rollback segment is used to “undo” any uncommitted changes applied from the redo log to the Data files. When a rollback segment is created, the segment is left offline and is not available to transactions; it must be brought online manually. Rollback segments can be stored in any tablespace.

All rollback segments should, however, be kept on one exclusive rollback segment tablespace (Bobrowski, 1994:318).

The rollback segments are stored in the database buffers and are therefore automatically protected by the redo log (Armstrong, et al. 1993:Chapter 1).

When a rollback segment's OPTIMAL storage parameter is increased, the likelihood of “snapshots too old” errors will be reduced (Bobrowski, 1994:323).

3.7.3 Checkpoints

Checkpoints occur when switching from one transaction log group to another. Settings of checkpoints can be changed. If the interval between checkpoints is too long, it can increase the underlying risk.

Three actions take place when a checkpoint occurs:

- (1) The DBWR process writes all modified data blocks in the SGA back to the data files of the database;
- (2) The LGWR background process updates the database control file to indicate when the last checkpoint occurred; and
- (3) The LGWR updates the headers in all the data files to indicate when the last checkpoint occurred.

In order to relieve the burden on the LGWR the optional CKPT background process can be used to update the headers of the data files rather than the LGWR (Bobrowski, 1994:384-386).

3.7.4 Full Backup

A full backup involves backup of all the Data files, online redo log files, the control file and the parameter files. A full backup cannot be made while the database is open (Armstrong, et al. 1993:Chapter 24).

3.7.5 Datafile backups

Oracle7 permits different backup methods : offline datafile backup and online datafile backup.

Offline datafile backup is the only backup option available if media recovery is disabled. If media recovery is disabled, the backups will have to include backups of the transaction log.

Online datafile backups are only available if media recovery is enabled. Online backups involves backing up one tablespace at a time. Offline tablespaces must be backed up first using the operating system commands or backup utility. When backing up the online tablespaces, the online table space must be marked for backup. After completion of the backup the online tablespace must be removed from the backup mode. If the online

tablespace is not marked for backup, the backup will be useless. If the tablespace is not removed after the backup, the database can become damaged if the server runs for several days with this condition, and the possibility of the server crashing is high (Bobrowski, 1994:345-349).

Data files and log files should be placed on different disks. A data file should never be placed on the same disk as the archive log file or an unmirrored log group, because it would be unrecoverable in the case of a single point of failure (Bobrowski, 1994:335).

3.7.6 Permanent and control file backup

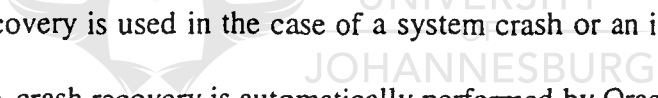
There should always be backup copies of the permanent files for a database as well as the control file (Bobrowski, 1994:63). The control file records the physical structure of the database, storing the database name, names and location of all data and redo log files and the current log sequence of the transaction log and time stamp of database creation (Armstrong, et al. 1993:Chapter 1). The control file is used during recovery to guide the application of the transaction log groups. Mirroring of the control file can be used to protect the control file against single points of failure (Bobrowski, 1994:64). If there is a change to the physical structure of the database, Oracle7 automatically modifies the control file (Armstrong, et al. 1993:Chapter 1). The control file should be backed up every time there is a change to the database structure (Bobrowski, 1994:64).

3.8 Recovery

The testing of database recoveries will ensure the soundness of a database backup strategy (Bobrowski, 1994:354).

The recovery process includes rollforward and rollback recovery. Rollforward recovery is the application of necessary transaction log groups to the backup copies of the damaged data files. Rollback recovery is the rolling back of any uncommitted transactions that are left after the rollforward recovery (Bobrowski, 1994:64).

3.8.1 Crash (Instance) recovery



This type of recovery is used in the case of a system crash or an instance failure, eg. power failure. A crash recovery is automatically performed by Oracle7 during the next instance startup. Oracle7 applies the necessary changes in the transaction log to the intact data files, thereby rolling forward and recovering all committed work (Bobrowski, 1994:357).

3.8.2 Media (Disk) failure recovery - closed recovery

Media recovery recovers the entire database to a state prior to the failure. This involves the following steps:

- (1) repair hardware;
- (2) restore any lost or damaged files using the most recent database backup; and
- (3) restore any necessary archived transaction log groups to disk. All archived transaction log groups have to be restored to the location as specified by the LOG_ARCHIVE_DEST. If the database transaction log or control file was not mirrored or media recovery was disabled, all work performed since the most recent backup will be lost (Bobrowski, 1994:357-360).

3.8.3 Online database recovery

Online database recovery is the recovery of damaged tablespaces or data files in an online database. The database will continue running uninterrupted while the damaged tablespace will be recovered offline (Bobrowski, 1994:362).

3.8.4 Parallel recovery

The time necessary to recover the database is reduced because multiple files undergo the rollforward recovery simultaneously (Bobrowski, 1994:361).

3.8.5 Cancel-Based recovery

Cancel-based recovery is used when one or more redo log groups have been damaged by media failure and are therefore not available for recovery procedures (Armstrong, et al. 1993:Chapter 25).

3.8.6 Time-based and change-based recovery

Time-based recovery and change-based recovery are used to recover the database to a specific point in the past (Armstrong, et al. 1993:Chapter 25).

3.9 Distributed Databases

3.9.1 Global object names

An object's "Global object name" must be uniquely identified throughout a distributed database. Oracle ensures uniqueness by ensuring that an object name is unique in its local database and that each database in a distributed database system has a unique data name. The unique database name has two components : the database name (8 characters) and the network domain that contains the database (Armstrong, et al. 1993:Chapter 21).

3.9.2 Two-phase commit

Oracle uses the two-phase commit mechanisms to ensure that all the nodes referenced in a distributed transaction either all commit or all rollback even if a network failure occurs during the committing of a transaction.


All DML operations performed by integrity constraints, remote procedure calls, and triggers are protected by the two-phase commit mechanism. The two-phase commit has two phases : the prepare phase and the commit phase.

Pending or in-doubt transactions are automatically recovered by the RECO background process when a network failure occurs during a two-phase commit process. Transactions resolved are then automatically removed from the Pending transaction table (Armstrong, et al. 1993:Chapter 21&22).

3.9.3 Snapshots

The snapshot feature provides asynchronous table replication. Triggers can be used to implement synchronous table replication. Replicates are read-only replications of the master table (Armstrong, et al. 1993:Chapter 21).

3.9.4 Database links



Database links are used to facilitate connections between different databases in a distributed database. A database link defines the path to a remote database. Two different kinds of links can be used : private and public. The use of private database links provides a higher level of security than public links. There is no method available to selectively restrict the use of public database links; a remote database specified by the public link can be connected to by any user.

A database link can use a central remote account or individual remote accounts. Using a central remote account creates certain security concerns :

- (1) users may be able to access more remote data than necessary and
- (2) public database links allow any user of the local database to access the remote database (Armstrong, et al. 1993:Chapter 21).

3.9.5 Passwords

A local username and password or an explicitly specified username and password can be user to connect to a remote database (Armstrong, et al. 1993:Chapter 21).

3.10 Conclusion

The objectives of this chapter were met in that the risks and controls present as well as the items for the internal control questionnaire were identified. All information were gathered from two reliable sources : Oracle7 server manuals and the textbook Mastering Oracle7 and Client/Server Computing by Steven M Bobrowski, one of the authors of the Oracle7 server manuals.

In the following chapter the controls of Oracle7 database management system will be compared with the controls of a general database environment, and an internal control questionnaire will be developed to be used by the auditor when conducting an audit of an Oracle7 database management system.

CHAPTER 4

SUMMARY

CONTENTS	PAGE
4.1 Introduction	69
4.2 Comparison table	70
4.3 Oracle7 internal control questionnaire	72



CHAPTER 4

SUMMARY

4.1 Introduction

The objective of this chapter is to summarise the previous chapter. The summary will be in the form of tables: the first (table 4.1) will be a comparison between a general database environment and the Oracle7 database management system and the second table (table 4.2) will be an internal control questionnaire developed for the auditor to use when performing an audit on the Oracle7 database management system.

References are made to relevant paragraphs in chapters 2 and 3. For more detail refer to the relevant paragraphs.



UNIVERSITY
OF
JOHANNESBURG

4.2 Comparison

General Control Features	Ref	Oracle7 Control Features	Ref
<p>Access Controls</p> <ul style="list-style-type: none"> - Passwords : Individuals <li style="padding-left: 100px;">Terminals <li style="padding-left: 100px;">Applications <li style="padding-left: 100px;">Organizational units - Utilities - Logical views - Access by applications - Password file protected - Privileged DBA functions - Master terminal - Logging of programs executed and program specifications - Logging of changes to DBMS - Terminal shutdown after number of unsuccessful access attempts 	2.5.2.1	<p style="text-align: center;">X</p> <p style="text-align: center;">X</p> <p style="text-align: center;">X</p> <p style="text-align: center;">X</p> <p style="text-align: center;">X</p> <p style="text-align: center;">X</p> <p style="text-align: center;">X</p> <p style="text-align: center;">X</p> <p style="text-align: center;">X</p>	3.6

Integrity controls	2.5.2.1	X	3.5.2
- Domain integrity		X	
- Entity integrity		X	
- Referential integrity			
Separation of duties	2.5.2.2	X	3.6.3
Ownership of data elements	2.5.2.2	X	2.6.1 3.5.7
Control over administration, creation of schemas and sub-schemas	2.5.2.2	X	3.5.6
Database management system log	2.5.2.2		
Control over maintenance-related utilities	2.5.2.2	X	3.6
DBA should not have access to initiate transactions	2.5.2.2		
Database error reporting	2.5.2.2 2.5.2.3	X	3.5.9
Control over data definition	2.5.2.4	X	3.3.3 3.5.1
Concurrency controls	2.5.2.5	X	3.5.3
Two-Phase locking	2.5.2.5	X	3.9.2
Audit functions and Audit trails	2.5.2.2	X	3.5.4
Transaction logs	2.5.2.6	X	3.7
Backup facilities	2.5.2.6	X	3.7
Rollforward and rollback procedures	2.5.2.6	X	3.7

Control over pointer structures	2.5.1.4	X	3.3.2 3.5.1
Data definition separated from the data	2.5.1.7	X	3.3.3 3.5.6 3.5.1

Table 4.1 : Comparison table

4.3 Oracle7 internal control questionnaire

Item	Reason/Implication	Ref	Yes	No
1.Redo logs maintained ?	Protects redo log against single point failure.	3.7.1		
2.ARCHIVELOG set ?	Facilitates -online datafile backup -media recovery -archiving of the redo log	3.7.1		
3.Only DBA has SELECT ANY TABLE privilege ?	Facilitates querying DBA views of data dictionary	3.5.1		
4.SERIALIZABLE = FALSE and ROW_LOCKING = ALWAYS ?	Facilitates automatic locking	3.5.3		

5.Triggers are enabled and trigger restriction = TRUE ?	The trigger restriction must be TRUE for the trigger to fire.	3.5.2.2		
6.No system privileges granted to PUBLIC?	Maintains security over access rights.	3.6.2		
7.Idle time for session limited ?	Prevents unauthorized access.	3.6.1		
8.Checked V\$LICENSE VIEW ?	Shows licence limits, current number of sessions, maximum number of concurrent sessions since instance started.	3.6.1		
9.LICENSE_MAX_SESSION ?	Facilitates restriction to maximum number of concurrent sessions.	3.6.1		
10.LICENSE_SESSION_WARNING ?	Facilitates warning when approaching maximum number of sessions.	3.6.1		
11.Only DBA has GRANT ANY PRIVILEGE or system privilege with the ADMIN OPTION ?	Facilitates granting or revoking of system privileges to users or roles.	3.6.2		
12.Only DBA has the CREATE PROCEDURE or CREATE ANY PROCEDURE system privilege ?	Facilitates creating or revoking of procedures that serve as security measures.	3.6.2		
13.Only DBA has the GRANT ANY ROLE or role granted with the ADMIN OPTION ?	Facilitates granting or revoking of privileges associated with roles.	3.6.3		
14.No transactions are listed in the DBA_2PC_PENDING table ?	Contains in-doubt and pending transactions not resolved.	3.9.2		
15.DBA_2PC_PENDING.MIXED= YES?	Facilitates automatic detection and flagging of incorrect decisions.	3.9.2		

16.No transactions are listed on the DBA_2PC_PENDING table flagged as MIXED ?	Indicates incorrect decisions.	3.9.2		
17.Control file mirrored ?	Protects control file against single point failure.	3.7.6		
18.LOG_CHECKPOINT_INTERVAL or LOG_CHECKPOINT_TIMEOUT ?	Facilitates forcing checkpoints at a predetermined event.	3.7.3		
19.All the groups of the mirrored online redo log have the same number of members ?	Safeguards the redo log against a single point failure.	3.7.1		
20.LOG_ARCHIVE_START ?	Facilitates automatic archiving at instance startup.	3.7.1		
21.Are the mirrored redo log and control file stored on separate disks ?	Facilitates protection against single point failure.	3.7.1		
22.RECOVERY_PARALLELISM ?	Facilitates concurrent recovery of processes.	3.8.4		
23.ORA_ENCRYPT_LOGIN= TRUE on the client and DBLINK_ENCRYPT_LOGIN= TRUE on the server ?	Facilitates encryption of passwords used to verify a connection.	3.6.1		
24.REMOTE_LOGIN_PASSWORD FILE= EXCLUSIVE ?	Facilitates the checking of password/user ID against an external password file.	3.5.6		
25.Passwords for SYSTEM and SYS changed after database is created ?	Changes the default password.	3.5.6		

26.Only the DBA is granted the SYSDBA and SYSOPER privileges according to the V\$PWFIL USER'S view ?	Ensures no user is granted powerful databases administration privileges.	3.5.6		
27.Is password authentication rather than operating system authentication used ?	Password authentication is more secure.	3.6.1		
28.Is auditing enabled ?	Facilitates auditing functions.	3.5.4		
29.Is access to SQL*DBA restricted to the DBA ?	Facilitates powerful database administrative functions.	3.5.8		
30.Is access to SQL*LOADER restricted to the DBA ?	Facilitates loading of non-Oracle data on to an Oracle database.	3.5.8		

Table 4.2 : Oracle7 internal control questionnaire



CHAPTER 5

CONCLUSION

CONTENTS	PAGE
5.1 Conclusion	77



CHAPTER 5

CONCLUSION

5.1 Conclusion

The objectives set for this short dissertation have been met. A general database environment and the Oracle7 database management system were compared, and an Oracle7 internal control questionnaire was developed.

Although Oracle7 supports the majority of features available in a database environment, it is important to realise that Oracle7 features have to be enabled by the database administrator and that manual database administration control functions still have to be performed. It is also important to note that Oracle7 alone does not provide a secure environment; operating system, network software, application software, user and hardware controls are necessary to provide a secure environment.

The internal control questionnaire is not to be seen as a substitute for an audit of an Oracle7 database management system. It was developed to help the auditor to understand the Oracle7 database management system, control the audit process and to identify the possible risks and controls that exist.

The Trusted Oracle7 Server and Oracle7 Parallel Server environments were not studied, this opens doors for more research regarding risks and controls present in the abovementioned environments.



BIBLIOGRAPHY

AICPA 1983: Report of the joint Data Base Task force.

ARMSTRONG, E; BOBROWSKI, S; CLOSKEY, C; LINDEN, B & PRATT, M 1993:

Oracle7 Server concepts manual. CD/Rom: Oracle Corporation.

ARMSTRONG, E; FRAZZINI, J; PORTFOLIO, T; QUIGLEY, B & SMITH, T 1993:

Oracle7 Server messages and codes manual(R). CD/Rom: Oracle Corporation.

BOBROWSKI, SM 1994: Mastering Oracle7 & client / server computing. San Francisco:

SYBEX.



UNIVERSITY
OF
JOHANNESBURG

DAMLIANIDES, M 1991: A control model for the evaluation and analysis of control

facilities in a simple path context model in a MVS/XA environment. Johannesburg:

Rand Afrikaans University (dissertation in partial fulfilment of a master's degree
in Computer Auditing).

FERNANDEZ, EB; SUMMERS, RC & WOOD, C 1981: Database security and integrity.

Massachusetts: Addison-Wesley.

JENKINS, B; COOKE, P & QUEST, P 1992: An audit approach to computers; fourth

edition. London: Coopers & Lybrand Deloitte.

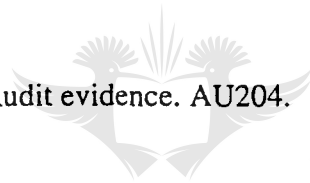
JOHNSTON, HN 1987: Auditing database integrity with special reference to Relational and Relationallike Database Management System. Johannesburg: Rand Afrikaans University (dissertation in partial fulfilment of a master's degree in Computer Auditing).

KUONG, JF 1983: Controls for advanced On-line / Data Base systems. Massachusetts: Management Advisory Publications.

PERRY, WE 1983: Ensuring Data Base integrity. New York: John Wiley.

SAICA 1993: Responsibilities and functions of the independent auditor. AU001.

SAICA 1986: Audit evidence. AU204.



UNIVERSITY
OF
JOHANNESBURG

SAICA 1986: Accounting systems and internal controls. AU230.

THORNE, JF 1980: Auditing Data Base management systems. (In: Rullo, TA ed. 1980: Advances in Data Base management. Philadelphia: Heyden, pp. 94-117.)

WEBER, R 1988: EDP Auditing conceptual foundations and practice. New York: McGraw-Hill.

ProQuest Number:28329527

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent on the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 28329527

Published by ProQuest LLC (2021). Copyright of the Dissertation is held by the Author.

All Rights Reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346