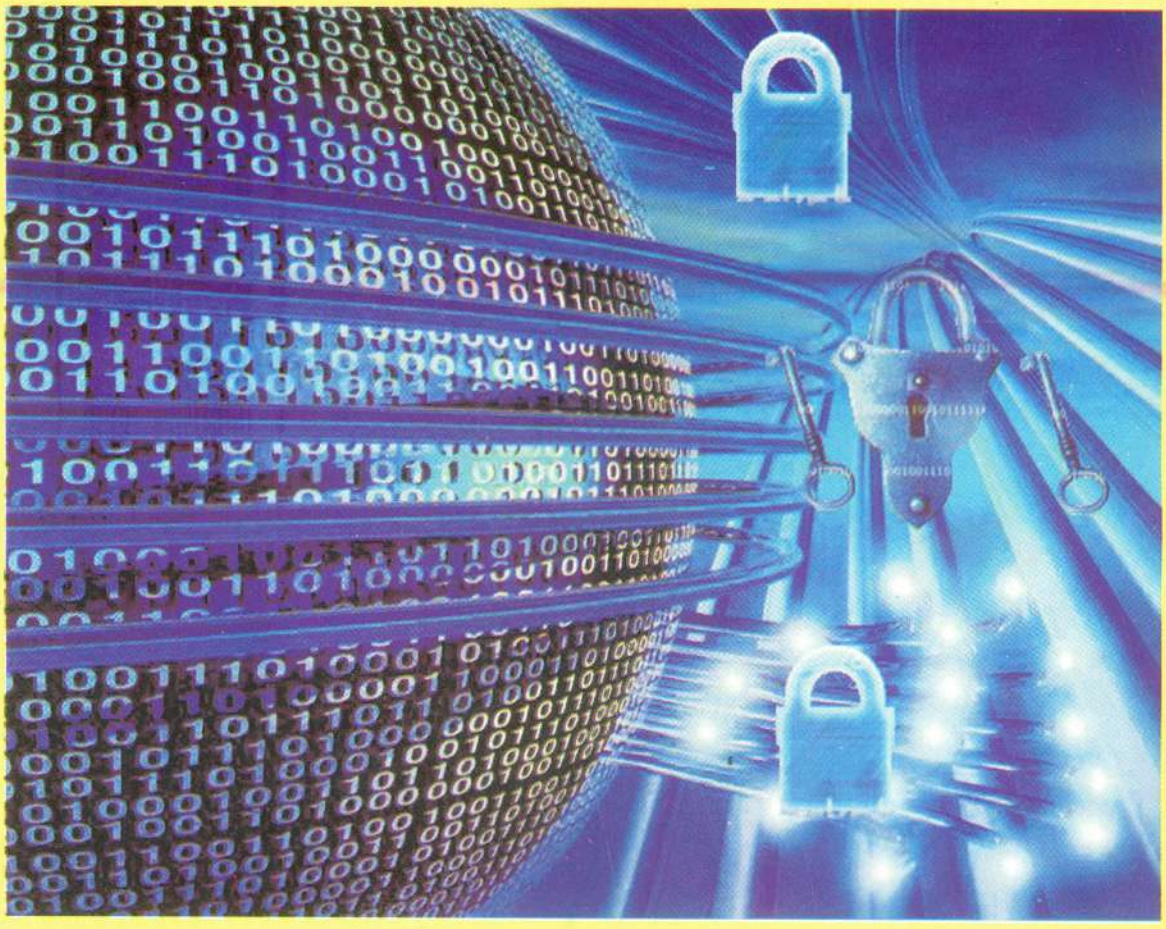
ignou
THE PEOPLE'S
UNIVERSITY

Indira Gandhi National Open University
School of Vocational Education and Training

MSEI-023
CYBER SECURITY



Database Security

2

"शिक्षा मानव को बन्धनों से मुक्त करती है और आज के युग में तो यह लोकतंत्रा की भावना का आधार भी है। जन्म तथा अन्य कारणों से उत्पन्न जाति एवं वर्गगत विषमताओं को दूर करते हुए मनुष्य को इन सबसे ऊपर उठाती है।"

- इन्दिरा गाँधी

"Education is a liberating force, and in our age it is also a democratising force, cutting across the barriers of caste and class, smoothing out inequalities imposed by birth and other circumstances."

- Indira Gandhi

ignou
THE PEOPLE'S
UNIVERSITY

Indira Gandhi National Open University
School of Vocational Education and Training

Block

2

DATABASE SECURITY

Programme Expert/Design Committee of Post Graduate Diploma in Information Security (PGDIS)

Prof. K.R. Srivathsan
Pro Vice-Chancellor, IGNOU

Mr. B.J. Srinath, Sr. Director & Scientist 'G',CERT-In, Department of Information Technology, Ministry of Communication and Information Technology, Govt of India

Mr. A.S.A Krishnan, Director, Department of Information Technology, Cyber-Laws and E-Security Group, Ministry of Communication and Information Technology, Govt of India

Mr. S. Balasubramony, Dy. Superintendent of Police, CBI, Cyber Crime Investigation Cell Delhi

Mr. B.V.C. Rao, Technical Director, National Informatics Centre, Ministry of Communication and Information Technology

Prof. M.N. Doja, Professor, Department of Computer Engineering, Jamia Milia Islamia New Delhi

Dr. D.K. Lobiyal, Associate Professor, School of Computer and Systems Sciences, JNU New Delhi

Mr. Omveer Singh, Scientist, CERT-In Department of Information Technology, Cyber-Laws and E-Security Group, Ministry of Communication and Information Technology Govt of India

Dr. Vivek Mudgil, Director, Eninov Systems Noida

Mr. V.V.Subrahmanyam, Assistant Professor School of Computer and Information Science IGNOU

Mr. Anup Girdhar, CEO, Sedulity Solutions & Technologies, New Delhi

Prof. A.K. Saini, Professor, University School of Management Studies, Guru Gobind Singh Indraprastha University, Delhi

Mr. C.S. Rao, Technical Director in Cyber Security Division, National Informatics Centre Ministry of Communication and Information Technology

Prof. C.G. Naidu, Director, School of Vocational Education & Training, IGNOU

Prof. Manohar Lal, Director, School of Computer and Information Science, IGNOU

Prof. K. Subramanian, Director, ACIIL, IGNOU Former Deputy Director General, National Informatics Centre, Ministry of Communication and Information Technology, Govt of India

Prof. K. Elumalai, Director, School of Law IGNOU

Dr. A. Murali M Rao, Joint Director, Computer Division, IGNOU

Mr. P.V. Suresh, Sr. Assistant Professor School of Computer and Information Science IGNOU

Ms. Mansi Sharma, Assistant Professor, School of Law, IGNOU

Ms. Urshla Kant
Assistant Professor, School of Vocational Education & Training, IGNOU
Programme Coordinator

Block Preparation

Unit Writer

Ms. Manka Vasti
B.E (Computer Science)
University of Pune
Faculty Associate (IT)
Institute of Apparel Management, Gurgaon
(Unit 1, 2, 3 & 4)

Block Editors

Prof. K.R. Srivathsan
Pro Vice-Chancellor
IGNOU

Ms. Urshla Kant
Assistant Professor, School of Vocational Education & Training, IGNOU

Proof Reading

Ms. Urshla Kant
Assistant Professor
School of Vocational Education & Training
IGNOU

Production

Mr. B. Natrajan
Dy. Registrar (Pub.)
MPDD, IGNOU, New Delhi

Mr. Jitender Sethi
Asstt. Registrar (Pub.)
MPDD, IGNOU, New Delhi

Mr. Hemant Parida
Proof Reader
MPDD, IGNOU, New Delhi

August, 2011

ISBN: 978-81-266-5616-5

Further information about the School of Vocational Education and Training and the Indira Gandhi National Open University courses may be obtained from the University's office at Maidan Garhi, New Delhi-110068. or the website of IGNOU www.ignou.ac.in

Printed and published on behalf of the Indira Gandhi National Open University, New Delhi, by the Registrar, MPDD

Laser typeset by Mctronics Printographics, 27/3 Ward No. 1, Opp. Mother Dairy, Mehrauli, New Delhi-30

Printed by : A-One Offset Printers, 5/34, Kirti Nagar Indl. Area, New Delhi-110015

BLOCK INTRODUCTION

Database security is a growing concern nowadays evidenced by an increase in the number of reported incidents of loss of or unauthorized exposure to sensitive data. As the amount of data collected, retained and shared electronically expands, so does the need to understand database security. The Defense Information Systems Agency of the US Department of Defense (2004), in its Database Security Technical Implementation Guide, states that database security should provide controlled, protected access to the contents of a database as well as preserve the integrity, consistency, and overall quality of the data. It is very important to develop an understanding of the issues and challenges related to database security and must be able to identify possible solutions. This block comprises of four units and is designed in the following way;

The **Unit one** deals with the introduction to database concepts, database management system, relational database management system. The concepts of Relational Algebra were undertaken. This unit also explains about the advantages of databases and Relational database management system. The E-R Model is also covered to explain about the entities, their properties and their relationships with other entities. The concept of database abstraction is also explained for the knowledge of the reader.

The **Unit two** describes the hands-on experience on the database. It starts with the concepts of oracle architecture. The methods to open oracle and SQL*PLUS which is the command line interpreter. It handles different types of queries that can be handled on Oracle database. It gives information about all the commands that are most commonly used in Oracle Sql database. The different types of hacking attempts are also a raised issue. It also talks about data dictionary and database objects that exist in the database.

The **Unit three** covers about distributed databases, their advantages and disadvantages, distributed database design. The concept of centralised databases is also highlighted. How distributed databases are advantageous over centralised database is mentioned too. Overall it is a unit that covers security to the database by considering the two mechanisms of storing data.

Unit four explains database transaction, its definition, database concurrency- a problem and its solution. There are some properties called ACID properties that need to be adhered to by the transactions. Various database concurrency control measures were mentioned for the database to function properly. Different operations and states of the transaction were also mentioned and the various security measures to be taken to prevent database from failure.

Hope you benefit from this block.

ACKNOWLEDGEMENT

The material we have used is purely for educational purposes. Every effort has been made to trace the copyright holders of material reproduced in this book. Should any infringement have occurred, the publishers and editors apologize and will be pleased to make the necessary corrections in future editions of this book.

UNIT 1 INTRODUCTION TO DATABASE CONCEPTS

Structure

1.0 INTRODUCTION

A database is structured collection of data. It contains information about enterprise which is actually useful for the decision making processes by the officials of the organization. The day to day examples of databases include telephone directories, catalogues, forms etc. However, a computerized database is a repository of data stored electronically. It is a collection of related information stored so that it is available to many users for different purposes. The organization of data in a database system is done by Database Management System (DBMS). One of the most powerful types of the database is the 'relational' model and programs which use this model are known as relational database management systems (RDBMS).

Database management system is one solution to all the problems and threats related to maintenance of critical data such as theft of data, loss of files, and wear and tear of physical media to store data, data inconsistency so on and so forth.

Relational Database Management Systems (RDBMS) are usually organized into one or more tables which consist of rows and columns.

Database: It is a structured collection of related data.

Database Management System: It is a software suite that is responsible for organization of database on the computer.

1.1 OBJECTIVES

After studying this unit, you should be able to:

- identify the significance of database and Database Management System (DBMS);

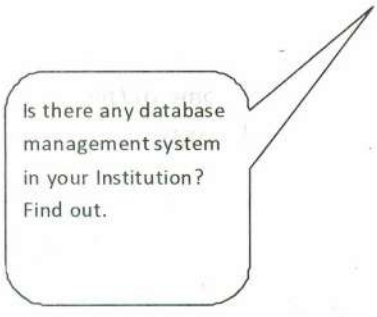
- advantages of using databases;
- traditional File based;
- concept of relations and thus, relational database management system;
- how to create/delete tables and databases, insert/update/delete and query tables in the databases; and
- elucidate the purpose of Entity relationship model.

1.2 ADVANTAGES OF DATABASE

The following are the advantages of using database:

a) **Reduces Data Redundancy**

Databases provide a mechanism wherein all the data is stored centrally or distributed at many locations. This way it reduces duplication of data and maintains latest copy of the data which is accessed by multiple users. Eg. In an Institute Management System where all the departments are computerized. In this type of system where the personal and academic details of the student may be accessed by different department's heads such as admission department head, Librarian, Teachers etc. Thus, change in information by one department will be seen by other departments as well. Thus, it reduces redundancy or duplication of unnecessary data which exists otherwise when the work is done manually.



b) **Controls data Inconsistency**

As mentioned in the example above, since all departments will then have the latest copy of the data, no inconsistency will exist.

c) **Data sharing**

It facilitates sharing of data amongst several users.

d) **Data Security**

It enforces security to the data by giving it protection from accidental loss, inaccessibility to unauthorized users, access only through username and password etc.

e) **Data Integrity**

It maintains data integrity by maintaining correct data and associations between data. It also provides constraints i.e. check to ensure that the data values confirm to certain specified rules.

f) **Enforces Standards**

It ensures that all the data follow the standards laid by the organization using the database or otherwise. This helps in data migration or interchange between platforms.

1.3 TRADITIONAL FILE ORIENTED APPROACH

Traditional file oriented applications have a Master File and a set of personal files to work upon. Eg. Cobol makes use of such approach. Such an arrangement is generally used in systems such as Payroll Management, Inventory and Financial and accounting system etc. In modern day to day organizations require the

intercommunication amongst the above said modules and sharing of data which the traditional approach is very poor at. Also, in such a system there is tight coupling between the files and the programs using data in the file. i.e. they both are dependent on each other. There are many disadvantages to traditional approach:

a) The data redundancy is high. i.e. the data is stored in multiple copies as are the number of modules requiring the data.

b) Any change in the field of the data in the master file requires changes in the programs too.

c) There is lack of flexibility as the program and its data are tightly coupled.

d) Concurrent users of the file may cause a lot of problems.

e) Data integrity can be made applicable only through programming code and not in the file itself.

f) Transactions such as Insert / Update / Delete are not possible through directly.

1.4 DATABASE ABSTRACTION

Now, we are already aware that database is a structured collection of data and database management system is a set of programs and interrelated files that provide access to the users of the data. It is highly essential to control the visibility of the underlying rules, procedures, functions and methodologies etc. from the users of the database. Therefore, providing only that much information as desired by the user of the system and hiding the rest part of it such as how the data is stored and maintained. This is called as data abstraction. Since, there are different types of users that exist in the system, the purpose to access the data is also different. Eg. The end user will only query the database for the purpose of extraction of the information, the application system analyst is more concerned about all the data that constitute the database, the relationships that exist between different data entities etc., the system analyst is more concerned about the factors that are related to the physical storage of the database. Thus, according to the above three mentioned levels the following diagram explain the types of levels present in the database management system.

Try finding out names of some of the DBMS's available.

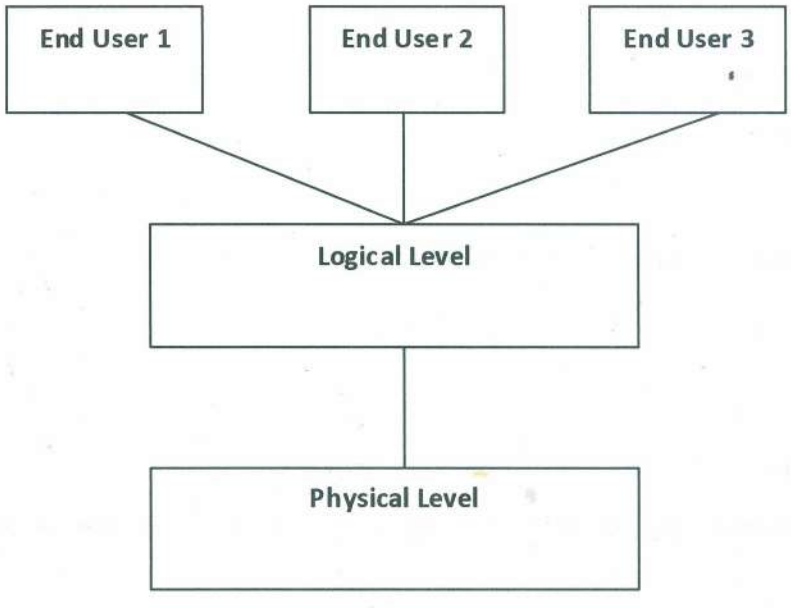


Fig. 1

Various Levels of Database Implementation

a) **Physical Level:** It is the lowest / internal level that concerns with the storage of database on the physical storage medium.

b) **Logical Level:** It is also called as the conceptual level which concerns what data is stored in the database. It basically deals with different data structures and their relations and association between them.

c) **External Level:** At this level, the end users of the system are present. This levels concerns with the way in which data is viewed by the end users.

1.5 RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)

Think and create some more tables with tuples and attributes eg. Employee details, Customer details, food-Items details etc.

In this type of Data Model, the data is represented in the form of tables i.e. rows and columns. These tables are called as relations. Each row in the table is collection of data values which represents some relation among the set of values. Thus, a table is a collection of such relationships. This data model was brought into picture by E.F.Codd, IBM and since then it is considered as one revolution in the field of database technology. Most of the available databases, such as Oracle, SQL etc., are based on Relation Data Model technology.

1.5.1 Some Important Terminologies

a) **Relation**

A relation is a table i.e. data arranged in the form of rows and columns. In a relation, within each column the set of values are similar and atomic (indivisible). Each row is distinct, i.e. no two rows are totally identical in terms of data values.

b) **Domain**

The set of values from which the values in each column are drawn. Eg. in an employee table the columns salary is numerical, name is alphabetic, address is alphanumerical, employee number can be given only odd values between 1 and 100 etc.

c) **Tuple**

The row of the table is called as tuple (pronounced as tupple). It is also called as record of the table. It is actually the horizontal collection of all the data values of the relation

d) **Attribute**

The column of the table is called as attribute of the table. It is also called as the field of the column.

e) **Degree**

The total number of attributes of the table is the degree of the relation

f) **Cardinality**

The total number of columns in the table consititute the cardinality of the relation.

Consider the following relation named STUDENT and relate the above concepts with it.

Table 1: Student

STUDENT NO.	NAME	PHYSICS	CHEMISTRY	BIOLOGY	MATHS	ENGLISH
IAM-1112-0001	TANYA SINHA	50	67	76	80	67
IAM-1112-0002	GAUTAM SHARMA	30	45	72	86	78
IAM-1112-0003	SAARANSH RASTOGI	55	66	65	87	65
IAM-1112-0004	VARUN KHANNA	56	61	74	83	76
IAM-1112-0005	GUNJAN KUKREJA	75	54	71	79	73
IAM-1112-0006	SMRIDHI PATIL	34	76	44	70	69
IAM-1112-0007	AKANKSHA SHARMA	54	82	54	55	68
IAM-1112-0008	ROBIN SINGH	76	43	70	79	72
IAM-1112-0009	ARUN DIWEDI	69	57	59	70	71
IAM-1112-0010	AMRIT KAUR	73	85	70	95	70

1.5.2 Types of Keys

a) **Primary Key**

It is the attribute(s) that uniquely distinguishes each row in the table i.e. no two values are same in that column. Each relation must have a primary key. Eg. Supplier#, Buyer# are the primary keys in Supplier and Buyer table respectively.

Table 2: Supplier

Supplier #	Supplier Name	Product Name	Qty-Ordered
SUP-001	JAI CHAND HOUSE	GENTS T-SHIRTS	300
SUP-002	KIRTI EXPORT HOUSE	LADIES SPORTS WEAR	350
SUP-003	EXPO GARMENTS	SOCKS	100
SUP-004	BRIJ DESIGN HOUSE	BOXER SHORTS	350
SUP-005	ORIENT CRAFT	WRIST BAND	400

Table 3: Buyer

Buyer #	Buyer Name	Product Name	Qty-Ordered	Supplier #
BUY-001	ADIDAS	GENTS T-SHIRTS	300	SUP-001
BUY-002	NIKE	LADIES SPORTS WEAR	350	SUP-002
BUY-003	REEBOK	SOCKS	100	SUP-001
BUY-004	LOTTO	BOXER SHORTS	350	SUP-003
BUY-005	LEVIS	WRIST BAND	400	SUP-005

b) **Foreign Key**

It is used to relate two or more tables. It is a non key attribute whose value is derived from the primary key of another table. Eg Supplier # is foreign key in Buyer table and primary key in Supplier table.

c) **Candidate Key**

The attribute(s) that can serve as primary key attribute of the relation is called candidate key. Alternate key is a candidate key that is not serving as the primary key for the relation.

Note: By joining the above two tables using Supplier# as the common column in between them, Supplier Table becomes the Foreign table and Buyer is the Primary or Master table.

1.5.3 Referential Integrity

It is an integrity methodology consisting of some rules to ensure the relationships between records of the related tables. This is required so as to prevent accidental manipulation/deletion of data of the related tables.

The following are certain set of rules to abide by when referential integrity is enforced:

- No foreign key field can be assigned a value in the primary table if the corresponding value doesn't exist in the primary key field of the foreign table. However, Null can be entered to show no relation between records of two tables.
- No record can be deleted from the primary table if the related record exists in the related table.
- No change in primary key field is allowed in the primary table if it has related records.

Consider the following tables Emp and Dept and refer the points that follow:

Table 4: Emp

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7689	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-81	3000		NULL
7369	SMITH	CLERK	7902	17-DEC-80	800		NULL
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		NULL

Table 5: Dept

DEPT NO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

According to referential integrity set of rules are:

- No value of deptno can be added in the Emp table if the corresponding value doesn't exist in dept table
- No value of deptno can be deleted from dept table if there exists related records in Emp table
- The change in value in the deptno in dept table is not allowed if there exists corresponding records in Emp table.

Check Your Progress 1

Notes: a) Space is given below for writing your answers.

b) Compare your answers with the one given at the end of this Unit.

1) Define database and Database Management system.

.....

.....

.....

.....

2) Write the advantages of using database.

.....

.....

.....

.....

3) What is RDBMS?

.....

.....

.....

.....

4) Define:

a) Primary Key

.....

.....

.....

.....

b) Foreign Key

.....

.....

.....

.....

c) Alternate Key

.....

.....

.....

.....

d) Candidate Key

.....

.....

.....

.....

e) Referential Integrity

.....

.....

.....

.....

5) Create two tables Student personal details and Student Academic details and relate them using a common field Stud_Id.

6) Consider the table given below and answer the questions given below :

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7689	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-81	3000		NULL
7369	SMITH	CLERK	7902	17-DEC-80	800		NULL
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		NULL

a) Name the primary key of the table.

b) What is the degree of the table above?

c) What is the cardinality of the table above?

d) Which attribute can serve as alternate key specifically for the set of values given above in the relation?

1.6 RELATIONAL ALGEBRA

It consists of some set of operations that can be performed on relations. Amongst different operations include Select, Project, Cartesian product, Union, Set Difference, Set Intersection and Join. Each operation requires operand(s). The relations in the database are the operands and the above mentioned operations are performed on them.

1.6.1 Select

- It selects rows /tuples / records from the relation based on some condition i.e. only those rows are selected that satisfy a given condition.
- It is denoted by sigma (σ)
- Eg σsal > 3000 in Table 4 : Emp will select only one record as only employee has salary more than 3000. Thus, the output will be

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10

1.6.2 Project

- It selects columns/attributes from the relation
- It is denoted by the greek letter pi (π)
- Eg π empno, ename, sal in Table 4 : Emp will output

EMPNO	ENAME	SAL
7839	KING	5000
7689	BLAKE	2850
7782	CLARK	2450
7566	JONES	2975
7654	MARTIN	1250
7499	ALLEN	1600
7844	TURNER	1500
7900	JAMES	950
7521	WARD	1250
7902	FORD	3000
7369	SMITH	800
7788	SCOTT	3000
7876	ADAMS	1100
7934	MILLER	1300

1.6.3 Cartesian Product

- It is a binary operation. i.e. it requires minimum of two operand relations to perform Cartesian product. Consider the following two relations namely Supplier and Buyer.

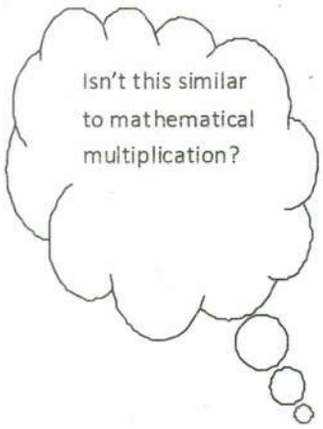
Supplier #	Supplier Name	Product Name	Qty-Ordered
SUP-001	JAI CHAND HOUSE	GENTS T-SHIRTS	300
SUP-002	KIRTI EXPORT HOUSE	LADIES SPORTS WEAR	350

And

Buyer #	Buyer Name
BUY-001	ADIDAS
BUY-002	NIKE

The Cartesian product of these relations will be

Supplier #	Supplier Name	Product Name	Qty-Ordered	Buyer #	Buyer Name
SUP-001	JAI CHAND HOUSE	GENTS T-SHIRTS	300	BUY-001	ADIDAS
SUP-001	JAI CHAND HOUSE	GENTS T-SHIRTS	300	BUY-002	NIKE
SUP-002	KIRTI EXPORT HOUSE	LADIES SPORTS WEAR	350	BUY-001	ADIDAS
SUP-002	KIRTI EXPORT HOUSE	LADIES SPORTS WEAR	350	BUY-002	NIKE



- It is denoted by the symbol cross (X). Thus catesian product of two symbols is denoted as Relation1 X Relation2. Thus, for the example above, it will be written as Supplier X Buyer.

1.6.4 Union Operator

- It is a binary operation that operates on two relations and produces a third relation that contains records from both relations.
- Eg. In a lucky draw a set of people are nominated to travel Europe whereas the other set of people are nominated to travel asia as given below:

Lucky Draw No.	Name	Location
9867	abc	U.K.
9944	def	France
9001	ghi	Switzerland

Travel Europe

Lucky Draw No.	Name	Location
1004	uvw	India
1010	xyz	Pakistan
2007	jkl	Sri Lanka

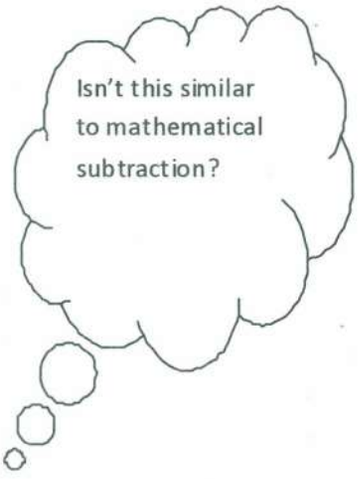
The output will be:

Lucky Draw No.	Name	Location
9867	abc	U.K.
9944	def	France
9001	ghi	Switzerland
1004	uvw	India
1010	xyz	Pakistan
2007	jkl	Sri Lanka

- It is denoted by union operator (∪).
- Two conditions that need to be taken care of before applying the Union operator on the relations is – Both the tables should be with the same degree and the domain for corresponding attributes of both the relations must be same.

1.6.5 Set Difference Operator

- It is a subtraction operator that finds the tuples that are in first relation but not in the second.
- It is denoted by minus (–) sign. Thus, considering the following two relations:



Lucky Draw No.	Name
9867	abc
9944	def
9001	ghi

First Draw

Lucky Draw No.	Name
9867	abc
9944	def
2007	jkl

Second Draw

The set difference operation Travel Europe – Travel Asia will output:

Lucky Draw No.	Name
9001	ghi

1.6.6 Set Intersection

- This binary operator finds the tuples that are common to both the relations.
- It is denoted by the symbol of intersection (∩).
- Thus for the following operands

Lucky Draw No.	Name
9867	abc
9944	def
9001	ghi

Travel Europe

Lucky Draw No.	Name
9867	abc
9944	def
2007	jkl

Travel Asia

- The set intersection operation Travel Europe – Travel Asia will output:

Lucky Draw No.	Name
9867	abc
9944	def

1.6.7 Join

- This operation joins two or more relations based on one common column.
- It is represented by the join symbol (∞)
- Consider the following two relations namely Emp and Dept:

Table 6: Emp

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7689	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-81	3000		NULL
7369	SMITH	CLERK	7902	17-DEC-80	800		NULL
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		NULL

Table 6: Dept

DEPT NO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

On applying join operation on the two relations based on the attribute deptno, the output produced will be:

EMPNO	ENAME	JOB	MGR NO	HIREDATE NO	SAL	COMM	DEPT	DEPT	DNAME	LOC
7839	KING	PRESIDENT		17-NOV-81	5000		10	10	ACCOUNTING	NEW YORK
7689	BLAKE	MANAGER	7839	01-MAY-81	2850		30	30	SALES	CHICAGO
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	02-APR-81	2975		20	20	RESEARCH	DALLAS
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	30	SALES	CHICAGO
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30	30	SALES	CHICAGO
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	30	SALES	CHICAGO
7900	JAMES	CLERK	7698	03-DEC-81	950		30	30	SALES	CHICAGO
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	30	SALES	CHICAGO
7902	FORD	ANALYST	7566	03-DEC-81	3000		NULL			
7369	SMITH	CLERK	7902	17-DEC-80	800		NULL			
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20	20	RESEARCH	DALLAS
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20	20	RESEARCH	DALLAS
7934	MILLER	CLERK	7782	23-JAN-82	1300		NULL			

Here the two of the columns are named deptno (deptno. of emp and dept respectively). This type of join where the combining of two tables is based on equality condition is called as equijoin. On removing one of the two repeated columns converts this join to natural join as shown below:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DNAME	LOC
7839	KING	PRESIDENT		17-NOV-81	5000		10	ACCOUNTING	NEW YORK
7689	BLAKE	MANAGER	7839	01-MAY-81	2850		30	SALES	CHICAGO
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	02-APR-81	2975		20	RESEARCH	DALLAS
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	SALES	CHICAGO
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30	SALES	CHICAGO
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	SALES	CHICAGO
7900	JAMES	CLERK	7698	03-DEC-81	950		30	SALES	CHICAGO
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	SALES	CHICAGO
7902	FORD	ANALYST	7566	03-DEC-81	3000		NULL		
7369	SMITH	CLERK	7902	17-DEC-80	800		NULL		
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20	RESEARCH	DALLAS
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20	RESEARCH	DALLAS
7934	MILLER	CLERK	7782	23-JAN-82	1300		NULL		

1.7 ER MODEL

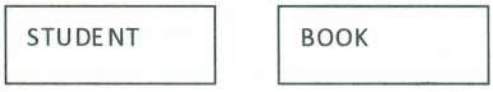
The ER Model stands for Entity Relationship Model which is based on the concept of real world comprising of entities and their relationships. It was brought as a concept by P.P. Chen. It is a high level, conceptual model that circulates around entitites and their relationships.

Some Basic concepts

a) **Entity**

It is an object that has a name and that exists with some of its properties. It is denoted by a rectangle.

Eg.



b) **Entity Set/Type**

A set of similar entities i.e. having common properties.

c) **Entity Instance**

An instance of the entity type.

d) **Types of entities**

Some entities are dependent while others are independent. The dependent entity depends on the other entity for its existence. It is also called as weak entities. The independent doesn't depend upon any entity for its existence. They are also termed as strong or regular entities.

e) **Relationships**

It defines the association between entities. Eg. Relationship between a father and his son, a teacher and the student etc. It is denoted by a diamond shaped box.

Eg:

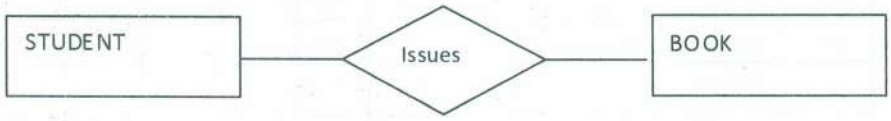
f) **Types of relationships**

i) **One to One**

There exists only one to one relationship of entity X with entity Y.

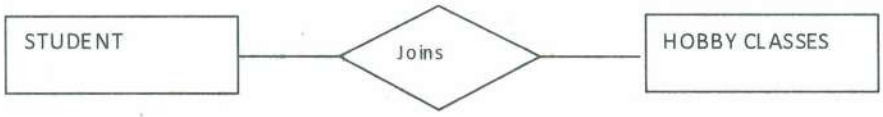
ii) **One to Many**

Entity X can have minimum 1 or maximum many relationships with entity Y.



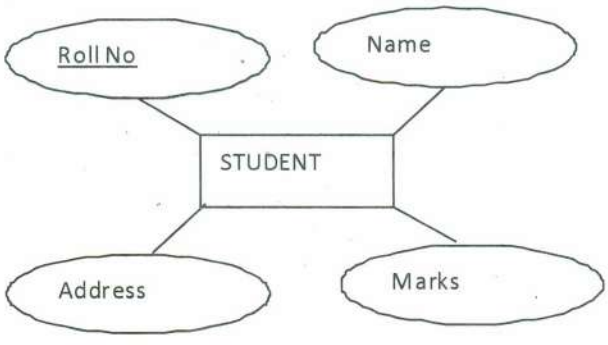
iii **Many to Many**

There exist many to many relationships between entities X and Y.



g) **Attributes**

It represents the property of an entity. Eg for the entity student the attributes are Student Roll No, Student Name, Address, Marks etc. It is represented by an oval.



The underlined attribute is the key field attribute.

Check Your Progress 2

Notes: a) Space is given below for writing your answers.

b) Compare your answer with the one given at the end of the Unit.

1) Define Relational Algebra.

.....

.....

.....

.....

2) Explain any four operations that can be performed on relations.

.....

.....

.....

.....

3) Take two table such as Student and Marks and perform the following operations:

a) Select operation

.....

.....

.....

.....

b) Product operation

.....

.....

.....

.....

c) Cartesian Product

.....

.....

.....

.....

d) Join

.....

.....

.....

.....

1.8 LET US SUM UP

This unit is based on the introductory concept of database, database management system, relational database management system. The concepts of Relational Algebra were undertaken. This unit also explains about the advantages of databases and Relational database management system. The E-R Model is also covered to explain about the entities, their properties and their relationships with other entities. The concept of database abstraction is also explained for the knowledge of the reader.

1.9 CHECK YOUR PROGRESS: THE KEY

Check Your Progress 1

1) **Database:** It is a structured collection of related data.

   Database Management System: It is a software suite that is responsible for organization of database on the computer.

2) The following are the advantages of using database:

   i) **Reduces Data Redundancy:** databases provide a mechanism wherein all the data is stored centrally or distributed at many locations. This way it reduces duplication of data and maintains latest copy of the data which is accessed by multiple users. Eg. In an Institute Management System where all the departments are computerized. In this type of system where the personal and academic details of the student may be accessed by different department's heads such as admission department head, Librarian, Teachers etc. Thus, change in information by one department will be seen

by other departments as well. Thus, it reduçes redundancy or duplication of unnecessary data which exists otherwise when the work is done manually.

ii) **Controls data Inconsistency:** as mentioned in the example above, since all departments will then have the latest copy of the data, no inconsistency will exist.

iii) **Data sharing:** It facilitates sharing of data amongst several users.

iv) **Data Security:** It enforces security to the data by giving it protection from accidental loss, inaccessibility to unauthorized users, access only through username and password etc.

v) **Data Integrity:** It maintains data integrity by maintaining correct data and associations between data. It also provides constraints i.e. check to ensure that the data values confirm to certain specified rules.

vi) **Enforces Standards:** It ensures that all the data follow the standards laid by the organization using the database or otherwise. This helps in data migration or interchange between platforms.

3) **RDBMS–** In this type of Data Model, the data is represented in the form of tables i.e. rows and columns. These tables are called as relations. Each row in the table is collection of data values which represents some relation among the set of values.

4) a) **Primary Key:** It is the attribute(s) that uniquely distinguishes each row in the table i.e. no two values are same in that column. Each relation must have a primary key. Eg. Supplier#, Buyer# are the primary keys in Supplier and Buyer table respectively.

b) **Foreign Key:** It is used to relate two or more tables. It is a non key attribute whose value is derived from the primary key of another table. Eg Supplier # is foreign key in Buyer table and primary key in Supplier table.

c) **Alternate key** is a candidate key that is not serving as the primary key for the relation.

d) **Candidate Key:** The attribute(s) that can serve as primary key attribute of the relation is called candidate key.

e) **Referential Integrity:** It is an integrity methodology consisting of some rules to ensure the relationships between records of the related tables. This is required so as to prevent accidental manipulation / deletion of data of the related tables.

The following are certain set of rules to abide by when referential:

Integrity is enforced

- No foreign key field can be assigned a value in the primary table if the corresponding value doesn't exist in the primary key field of the foreign table. However, Null can be entered to show no relation between records of two tables.
- No record can be deleted from the primary table if the related record exists in the related table.
- No change in primary key field is allowed in the primary table if it has related records.

5) **Student Personal Details**

Stud_Id	Name of the Student	Father's Name	Address	Phone

Student Academic Details

Stud_Id	Name of the Student	Marks1	Marks2	Marks3

The above two tables are related usinf Stud_Id as the primary key to Student Personal Details and Foreign key to Student Academic details.

6) a) EmpNo

b) 8

c) 14

d) Ename

Check Your Progress 2

1) In this type of Data Model, the data is represented in the form of tables i.e. rows and columns. These tables are called as relations. Each row in the table is collection of data values which represents some relation among the set of values. Thus, a table is a collection of such relationships. This data model was brought into picture by E.F.Codd, IBM and since then it is considered as one revolution in the field of database technology. Most of the available databases, such as Oracle, SQL etc. are based on Relation Data Model technology

2) and 3) The relations in the database are the operands and the below mentioned operations are performed on them.

a) **Select**

- It selects rows/tuples/records from the relation based on some condition i.e. only those rows are selected that satisfy a given condition.
- It is denoted by sigma (σ)
- Eg σ sal > 3000 in Table 4: Emp will select only one record as only employee has salary more than 3000. Thus, the output will be

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT NO
7839	KING	PRESIDENT		17-NOV-81	5000		10

b) **Project**

- It selects columns/attributes from the relation
- It is denoted by the greek letter pi(π)
- Eg (π) empno, ename, sal in Table 4: Emp will output

EMPNO	ENAME	SAL
7839	KING	5000
7689	BLAKE	2850
7782	CLARK	2450
7566	JONES	2975
7654	MARTIN	1250
7499	ALLEN	1600
7844	TURNER	1500
7900	JAMES	950
7521	WARD	1250
7902	FORD	3000
7369	SMITH	800
7788	SCOTT	3000
7876	ADAMS	1100
7934	MILLER	1300

c) **Cartesian Product**

- It is a binary operation. i.e. it requires minimum of two operand relations to perform Cartesian product. Consider the following two relations namely Supplier and Buyer

Supplier #	Supplier Name	Product Name	Qty-Ordered
SUP-001	JAI CHAND HOUSE	GENTS T-SHIRTS	300
SUP-002	KIRTI EXPORT HOUSE	LADIES SPORTS WEAR	350

And

Buyer #	Buyer Name
BUY-001	ADIDAS
BUY-002	NIKE

The Cartesian product of these relations will be

Supplier #	Supplier Name	Product Name	Qty-Ordered	Buyer #	Buyer Name
SUP-001	JAI CHAND HOUSE	GENTS T-SHIRTS	300	BUY-001	ADIDAS
SUP-001	JAI CHAND HOUSE	GENTS T-SHIRTS	300	BUY-002	NIKE
SUP-002	KIRTI EXPORT HOUSE	LADIES SPORTS WEAR	350	BUY-001	ADIDAS
SUP-002	KIRTI EXPORT HOUSE	LADIES SPORTS WEAR	350	BUY-002	NIKE

- It is denoted by the symbol cross (X). Thus catesian product of two symbols is denoted as Relation1 X Relation2. Thus, for the example above, it will be written as Supplier X Buyer.

d) **Union Operator**

- It is a binary operation that operates on two relations and produces a third relation that contains records from both relations.
- Eg. In a lucky draw a set of people are nominated to travel Europe whereas the other set of people are nominated to travel asia as given below:

Lucky Draw No.	Name	Location
9867	abc	U.K.
9944	def	France
9001	ghi	Switzerland

Travel Europe

Lucky Draw No.	Name	Location
1004	uvw	India
1010	xyz	Pakistan
2007	jkl	Sri Lanka

Travel Asia

The output will be:

Lucky Draw No.	Name	Location
9867	abc	U.K.
9944	def	France
9001	ghi	Switzerland
1004	uvw	India
1010	xyz	Pakistan
2007	jkl	Sri Lanka

- It is denoted by union operator ($\cup$).
- Two conditions that need to be taken care of before applying the Union operator on the relations is – Both the tables should be with the same degree and the domain for corresponding attributes of both the relations must be same.

UNIT 2 HANDS-ON DATABASE USAGE AND HACKING ATTEMPT

Structure

2.0 INTRODUCTION TO ORACLE

Oracle is one of the most popular DBMS that is based on Relational Database Management System Model. This is one of the most commonly used DBMS used across various organizations. This unit will be totally taking care of hands-on Oracle database. Oracle was developed by Relational Software Incorporated (RSI), in the year 1977 by Larry Ellison, Bob Minar and Ed Dates.

Oracle Architecture

The Oracle Database Management System consists of the following components:

The Oracle Server and Oracle Instance

The Oracle Server is a relational database management system that consists of an Oracle Database and an Oracle Instance. The Oracle database is actually the physical storage of Information whereas the term Oracle Instance refers to the set of programs on the server that provides the information stored in the Oracle Database. Refer the Fig. 1 to get information about Oracle's Architecture.

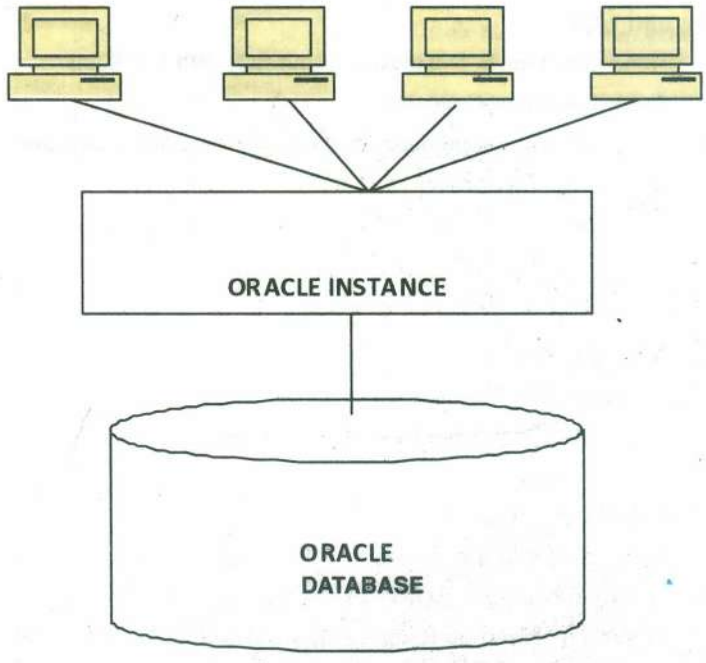


Fig. 1 : Oracle Architecture

2.1 OBJECTIVES

After studying this unit, you should be able to:

- understand oracle and Hands-on experience;
- explain the architecture of Oracle;
- explain concept of SQL;
- explain DDL and DML;
- explain transaction control statements; and
- explain hacking of databases – a concept.

2.2 ORACLE AND SQL

2.2.1 Hands-on Oracle Database – using SQL *PLUS shell

a) SQL stands for Structured Query Language. It has a proper structured syntax which is used to query /insert/update the database. Oracle SQL * PLUS is the shell to query Oracle database. To start SQL *Plus session on Oracle, consider Fig. 2 shown below:

2.2.2 Some Important Concepts

Object – An object is an identifiable entity which has characteristics and behavior. In a database, objects refer to entitites that exist within the database such as Relations, Views, Stored Procedures and Functions, Synonyms so on and so forth. In Oracle database management system, the different database objects that exist in the database are as follows:

a) **Relation:** a table that stores data in the form of rows and columns.

b) **Views:** it is a virtual table which selects data from one or more underlying base tables, but can be queried as if it were one table.

c) **Stored Procedures/Functions:** It is a procedure/function stored in compile form in the database.

d) **Synonyms:** These are generally alternate names given to the database objects.

e) **Indexes:** It is a database object which is used to keep track of rows and columns of the table.

Data Dictionary – It is a repository of data about data i.e. metadata.

Upon creation of the database objects, the details are stored in Data Dictionary. For any further modification in the definition of database objects, Data Dictionary is consulted.

Data Dictionary gives the inside view and the structure of the Oracle database. It contains information and details about database objects, their logical structure, their relationships amongst themselves etc. The views of data dictionary are divided into three general categories:

- User
- All
- DBA

2.2.3 Data Definition Language vs. Data Manipulation Language

Data Definition Language

It is a type of language which is responsible for creating and setting the database schema consisting of relations, views, stored procedures, functions etc. Some of the DDL Commands include:

- Create/Alter/Drop Schema objects
- Grant and Revoke Commands for granting and revoking privileges

Data Manipulation Language

Data Manipulation Language (DML) includes retrieval of information/inserting records/deletion of records/modification of data etc. Some of the commands from the command set include:

- Insert/Update/Delete commands
- Select and its variations etc.

2.2.4 Processing Capabilities of SQL

The following are some of the capabilities of SQL:

- It provides commands to define or alter the database objects through Data Definition Language (DDL)
- It allows users of the database to insert/update/delete/query data from the database through simple Data Manipulation Language (DML).
- It provides security to the system through specifying access rights to database objects.
- It includes data integrity checking.
- It provides control over transaction processing through transaction Control Language (TCL).

2.2.5 Datatypes in Oracle

Table 1: Types of datatypes

DATATYPE	DESCRIPTION
CHAR(SIZE)	FIXED LENGTH CHARACTER STRING WITH SIZE BYTES
VARCHAR2 (SIZE)	TO STORE VARIABLE LENGTH CHARACTER STRING WITH MAXIMUM SIZE SPECIFIED WITH SIZE ATTRIBUTE
NUMBER (P,S)	TO STORE NUMERIC VALUES WITH PRECISION P RANGING FROM 1-38 AND S SCALE IN BETWEEN RANGE -84 TILL 127
DATE	VALID DATE IN DD-MON-YYYY FORMAT
LONG	VARIABLE LENGTH CHARACTER DATA UPTO FEW GIGA BYTES
RAW(SIZE)	IT STORES BINARY DATA OF LENGTH SIZE BYTES
LONG RAW	VARIABLE LENGTH BINARY DATA UPTO FEW GIGA BYTES

Check Your Progress 1

Notes: a) Space is given below for writing your answers.

b) Compare your answers with the one given at the end of the Unit.

1) Define SQL.

.....

.....

.....

.....

.....

2) Differentiate between DDL and DML.

.....

.....

.....

.....

.....

3) What is data dictionary?

.....

.....

.....

.....

.....

4) Name any four datatypes in Oracle.

.....

.....

.....

.....

.....

5) What are the processing capabilities of SQL?

.....

.....

.....

.....

.....

6) Write steps to open SQL *PLUS Shell.

.....

.....

.....

.....

.....

2.3 SIMPLE SQL QUERIES USING ORACLE DATABASE

There are many data objects already built in Oracle, amongst those the relations EMP and DEPT will be used now onwards for dealing with different queries. Each statement is ended with Statement Delimiter (;).

2.3.1 Select Command and Its Variations

- Selecting all columns from the table

SYNTAX : SELECT * FROM TABLENAME;

Eg.

SQL > SELECT * FROM EMP;

Table 2: Emp Table

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7689	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-81	3000		NULL
7369	SMITH	CLERK	7902	17-DEC-80	800		NULL
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		NULL

- **Selecting few columns from the table**

SYNTAX : SELECT COLUMN NAME1 , COLUMN NAME2,... FROM TABLENAME;

Eg.

SQL > SELECT EMPNO, ENAME, SAL, DEPTNO FROM EMP;

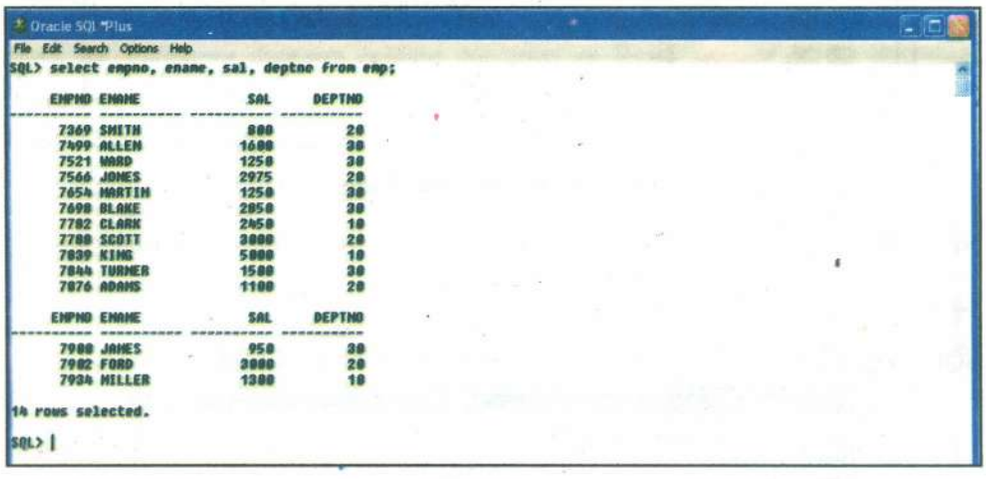


Fig. 5: Only few columns from Emp table are selected

- **Selecting rows based on condition**

SYNTAX: SELECT COLUMN NAME1 , COLUMN NAME2,... FROM TABLENAME WHERE <CONDITION>;

Eg.

SELECT EMPNO, ENAME, HIREDATE,SAL, DEPTNO FROM EMP WHERE HIREDATE >'01-JAN-1982'

Table 3: Records based on condition on Emp Table

EMPNO	ENAME	HIREDATE	SAL	DEPTNO
7788	SCOTT	09-DEC-82	3000	20
7876	ADAMS	12-JAN-83	1100	20
7934	MILLER	23-JAN-82	1300	NULL

2.3.2 Adding Comments

To add comments to SQL queries, the following two methodologies are used:

– Single Line Comment – where the comment begins with two hyphens i.e. (--). It will comment the part of the statement starting with - till the end of the statement.

– Multiline comments – Where the comment begins with/* and ends with */. Any part of the query can be commented using multiline comment.

2.3.3 Handling Null Values

The Null values are handled using Null Value Function in Oracle.

SYNTAX: NVL (attribute name, value to be substituted)

Eg.

SQL> SELECT EMPNO, ENAME, NVL (COMM, 0) FROM EMP;

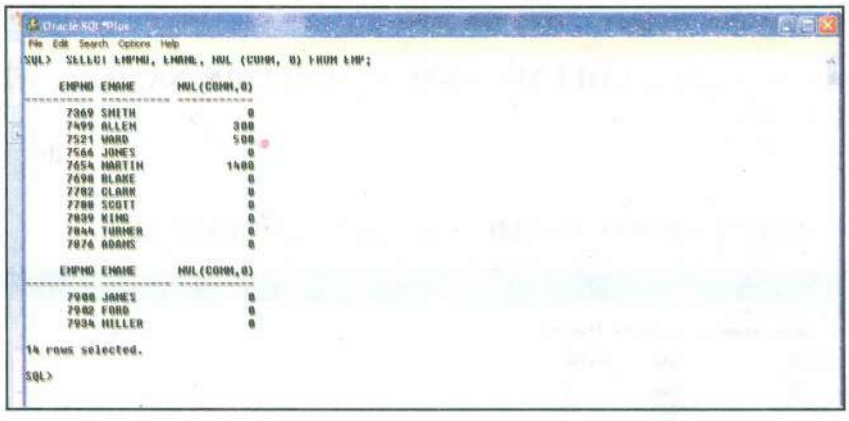


Fig. 6: Substituting Null Values

Note - To check for a NULL value in the field, IS NULL clause is used

Eg:

SQL> SELECT ENAME, COMM FROM EMP WHERE COMM IS NULL;

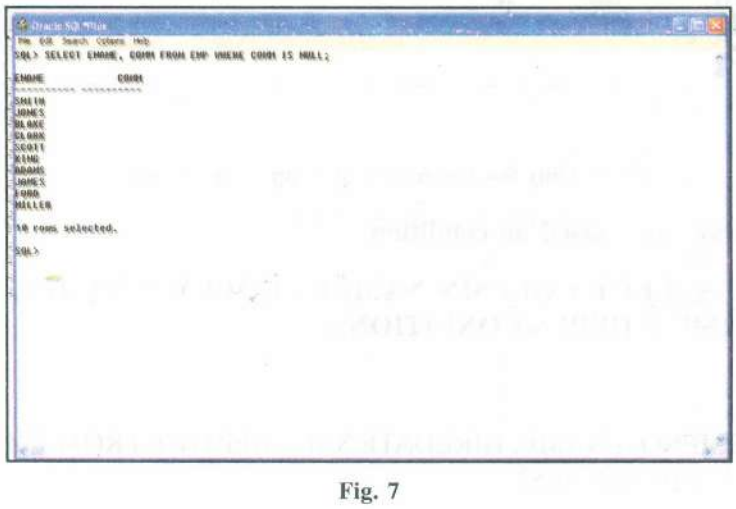


Fig. 7

2.3.4 Selecting ALL vs. DISTINCT Values from the Table

By default all values (including duplicastes) are selected from each column.

Eg:

SQL > SELECT ALL EMPNO FROM EMP;

is same as

SQL> SELECT EMPNO FROM EMP;

To have distinct values i.e. to eliminate repeated values, add DISTINCT keyword before the column name as shown below:

Eg.

SQL> SELECT DISTINCT COMM FROM EMP;

COMM
1400
300
0
500

2.3.5 Column Alias

It is a temporary name given to the column only for the purpose of displaying output from it.

Eg. Select EMPNO "Employee Number" FROM EMP;

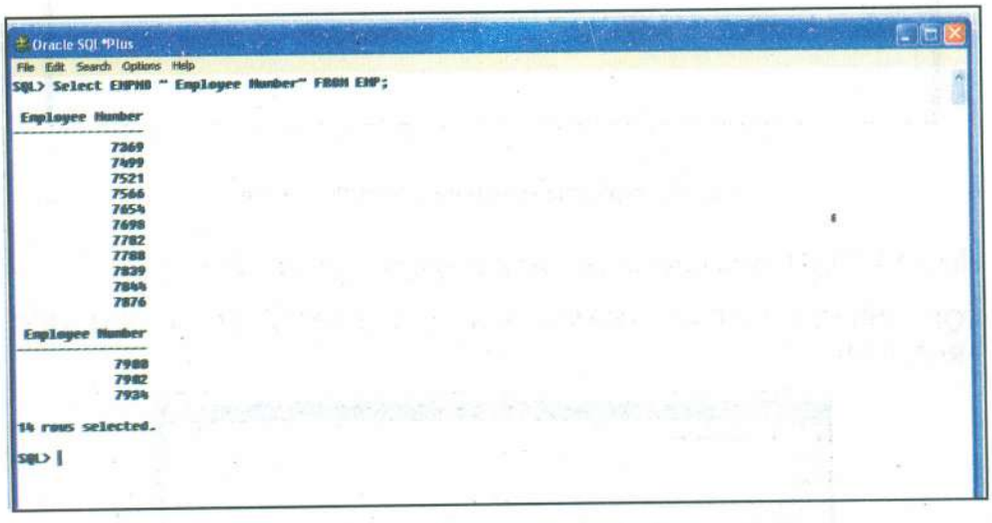


Fig. 8: Column Alias

2.3.6 Calculations in Query

Simple calculations such as multiplication, addition, subtractions, division, mod etc. can be done in the SQL query itself. There exists a dummy table in Oracle with the table name DUAL. To see the table structure of any table the syntax is as follows:

SYNTAX: DESC tablename;

Or

DESCRIBE tablename;

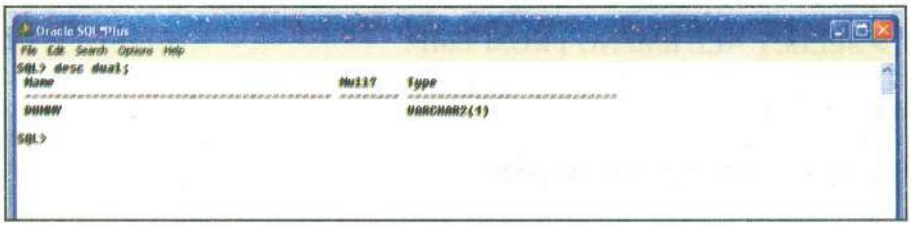


Fig. 9: Structure of the structure of the table

All calculations can be performed on this table such as

SQL > SELECT 12 * 4 "ANSWER" FROM DUAL;

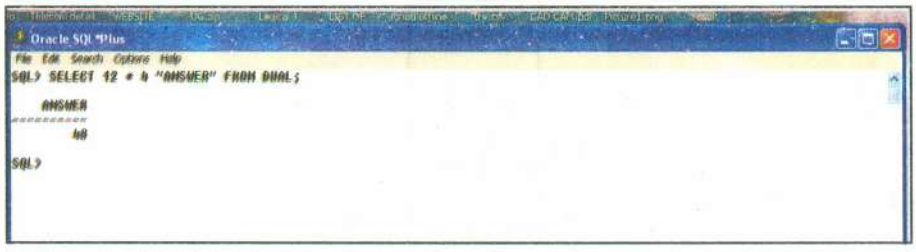


Fig. 10: Applying calculations on table

SQL > SELECT SYSDATE FROM DUAL;

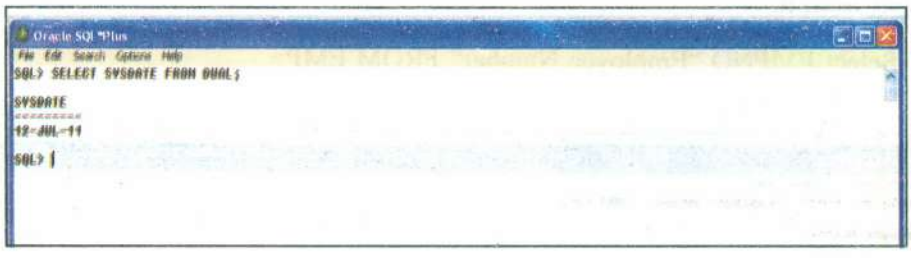


Fig. 11: Displaying system date from the system

Note: SYSDATE parameter always returns system's current date.

SQL> SELECT EMPNO, ENAME, SAL, SAL+COMM "TOTAL SALARY" FROM EMP;

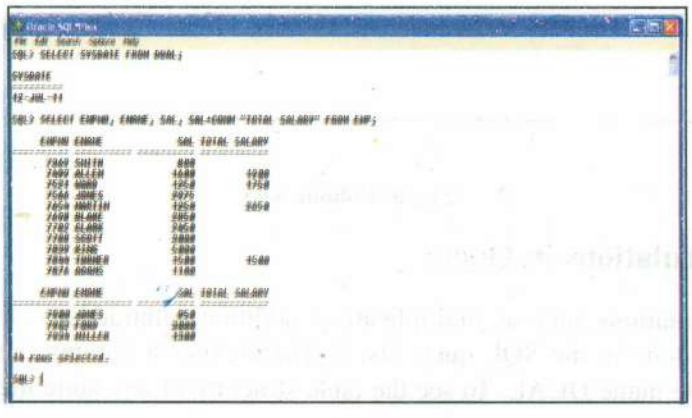


Fig. 12: Displaying calculation in the query

2.3.7 Inserting Text in the Query

To insert text in the result of the query single quotes (') are used.

SQL> SELECT EMPNO, ENAME , 'EARNS Rs.', SAL "SALARY" FROM EMP;

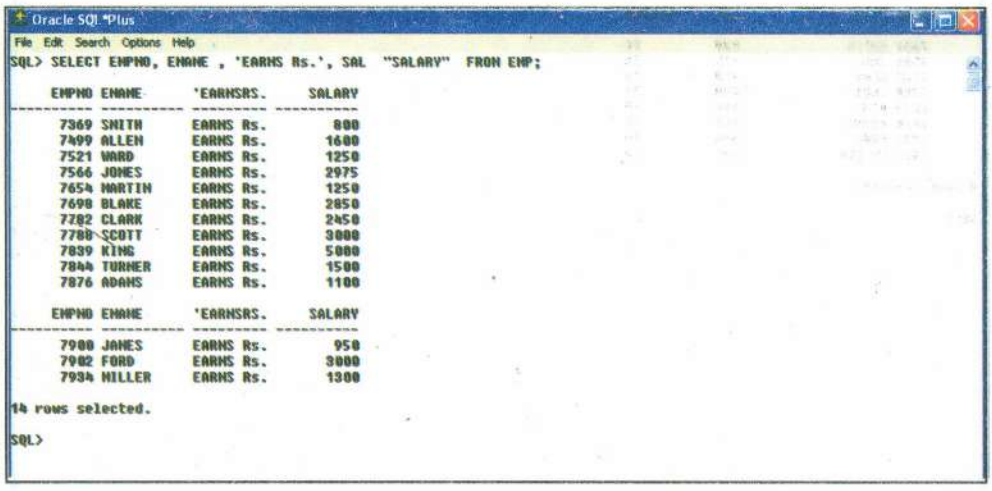


Fig. 13: Adding Text in the query

2.3.8 Sorting the Output – ORDER BY Clause

Order By : This clause is used to sort the data of the table in the output of the query.

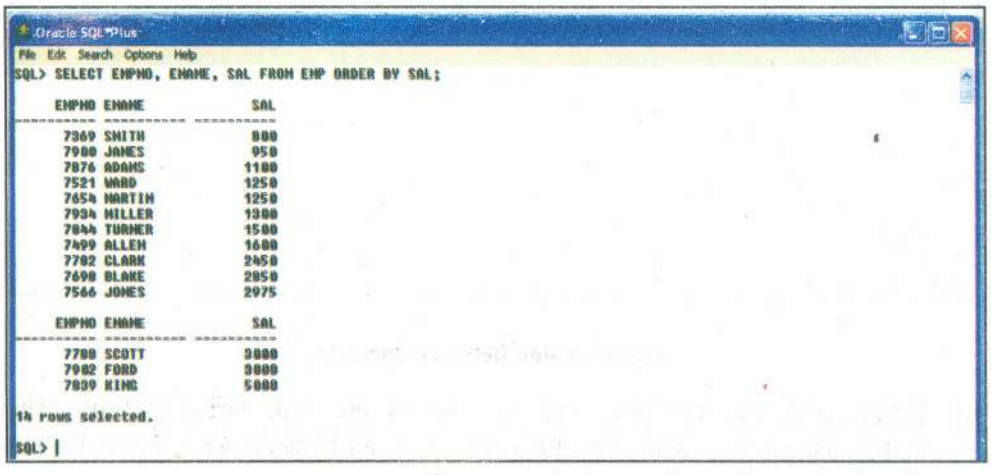
SYNTAX : SELECT COLUMN NAME1, COLUMNNAME2, ..., FROM

TABLENAME WHERE <CONDITION> ORDER BY

<COLUMNNAME>;

Eg.

SQL> SELECT EMPNO, ENAME, SAL FROM EMP ORDERBY SAL;



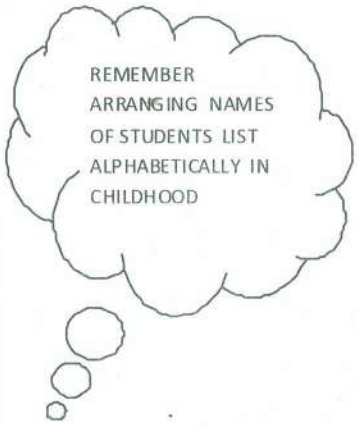


Fig. 14: Sorting the result in the query

2.3.9 Conditions based on Range, Pattern Matching and List of Items

a) Range – To define a continuous range of values, Between operator is used. The range specified in this operator is continuous inclusive of upper and lower bound values.

SQL> SELECT EMPNO, ENAME, SAL, DEPTNO FROM EMP WHERE DEPTNO BETWEEN 10 AND 20;

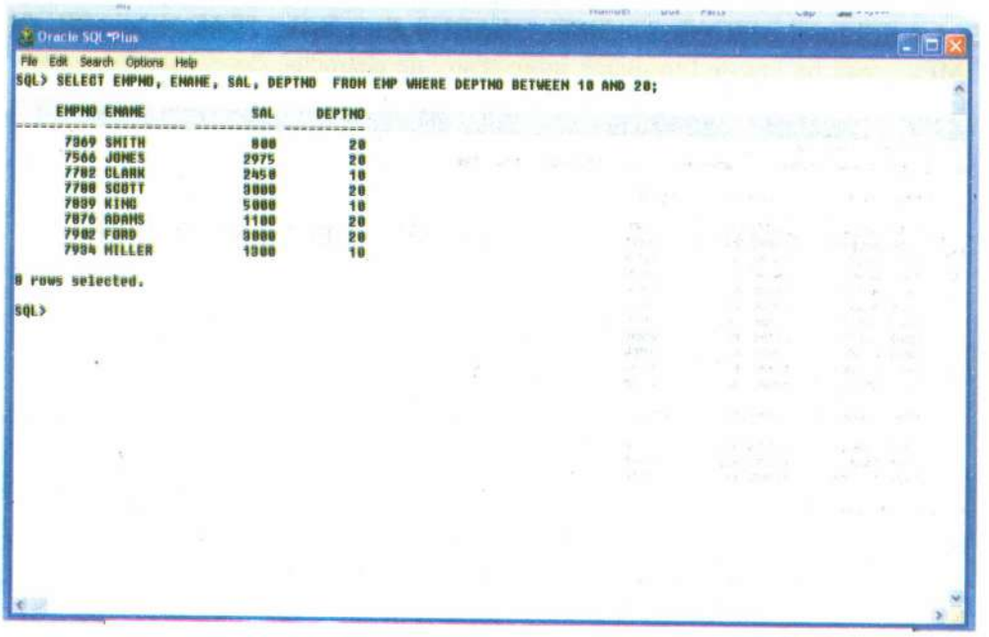


Fig. 15: Using Between Operator

SQL> SELECT EMPNO, ENAME, HIREDATE, DEPTNO FROM EMP WHERE HIREDATE BETWEEN '09-JUN-81' AND '09-DEC-82';

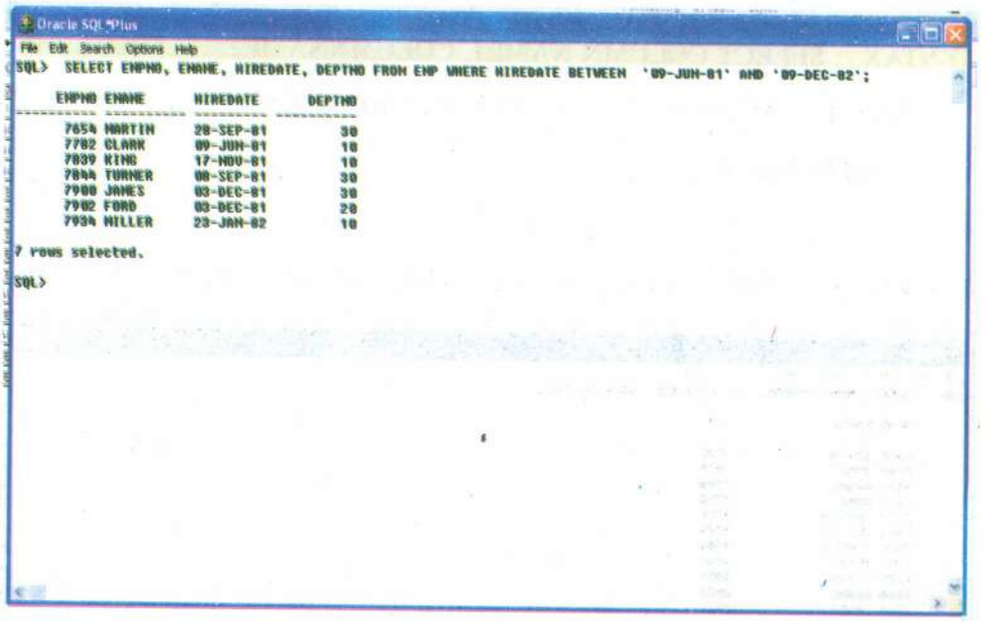


Fig. 16: Using Between Operator

b) Pattern Matching operator: This is used for matching string patterns. This makes use of two wildcard characters such as Underscore (_) and Percent (%). Consider the following strings:

- Bat
- Cat
- Mat
- Pat etc.

All these strings have same no. and type of characters with the difference only in

the intial letter. Thus all the above given strings can be mentioned as ' _at'. Therefore, only by changing the intial letter, all the above strings be produced. Similarly, in SQL the following two operators are used to apply condition on different strings with same or different patterns:

- Underscore (_): It matches exactly one character. Multiple underscores can be inserted to match more than one character.
- Percent (%): It matches with many characters.

Eg:

SQL> SELECT EMPNO, ENAME FROM EMP WHERE ENAME LIKE '_ _ _ _S';

SQL> SELECT EMPNO, ENAME, DEPTNO FROM EMP WHERE ENAME LIKE 'J%';

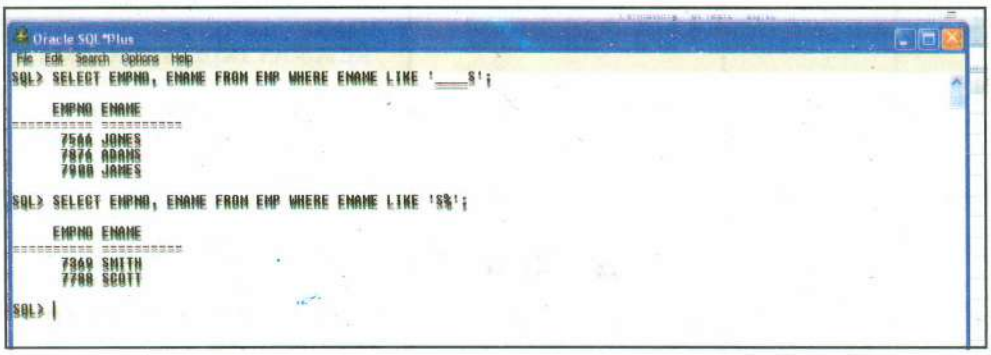


Fig. 17: using pattern matching operators

c) List of Items (IN Operator): This operator catches a value specified in the list of values. The values can be inserted randomly in the list. Consider the following lists:

- List of names of students in a class
- List of all Vegetables
- List of countries in Asia etc.

SYNTAX:

SQL> SELECT COLUMN NAME1, COLUMN NAME2, COLUMN NAME3,... WHERE COLUMNNAME IN (VALUE1, VALUE2, VALUE3,...);

Eg:

SQL > SELECT DEPTNO, DNAME FROM DEPT WHERE LOC IN ('DALLAS', 'BOSTON');

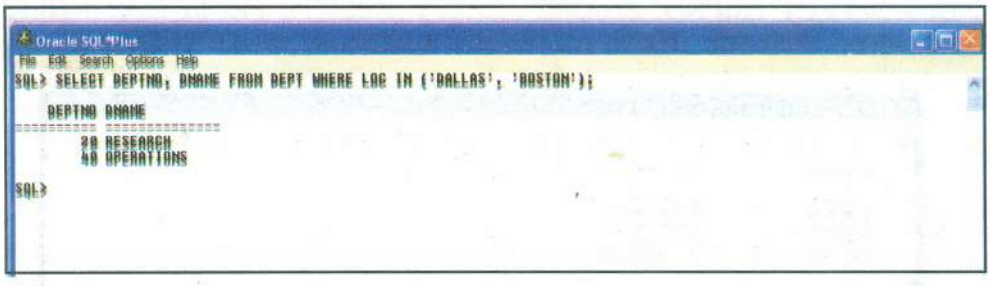


Fig. 18: using IN operator

2.3.10 Relational and Logical Operators

Relational and Logical: To compare two values, there are many relational operators available and to combine two or more conditions there are three Logical operators as shown below:

Table 4: List of Relational Operators

SNO.	OPERATOR	OPERATOR NAME	DESCRIPTION
1	>	GREATER THAN	RETURNS TRUE WHEN FIRST OPERAND IS GREATER THAT THE SECOND OPERAND, OTHERWISE RETURNS FALSE
2	<	LESS THAN	RETURNS TRUE WHEN FIRST OPERAND IS LESS THAN THE SECOND OPERAND, OTHERWISE RETURNS FALSE
3	>=	GREATER THAN AND EQUAL TO	RETURNS TRUE WHEN FIRST OPERAND IS GREATER THAN OR EQUALS TO THE SECOND OPERAND, OTHERWISE RETURNS FALSE
4	<=	LESS THAN AND EQUAL TO	RETURNS TRUE WHEN FIRST OPERAND IS LESS THAN OR EQUALS TO THE SECOND OPERAND, OTHERWISE RETURNS FALSE
5	=	EQUALS TO	RETURNS TRUE WHEN FIRST OPERAND IS EQUALS TO THE SECOND OPERAND, OTHERWISE RETURNS FALSE
6	<>	NOT EUAL TO	RETURNS TRUE WHEN BOTH THE OPERANDS ARE UNEQUAL

Table 5: List of Logical Operators

SNO.	OPERATOR	DESCRIPTION
1	AND	RETURNS TRUE WHEN BOTH THE CONDITIONS ARE TRUE ELSE RETURNS FALSE
2	OR	RETURNS TRUE WHEN EITHER ONE OR BOTH THE CONDITIONS ARE TRUE ELSE RETURNS FALSE
3	NOT	NEGATES THE OUTPUT

Eg:

SQL> SELECT EMPNO, ENAME , SAL, HIREDATE FROM EMP WHERE SAL >2000 AND HIREDATE BETWEEN '01-JAN-81' AND '31-DEC-81';

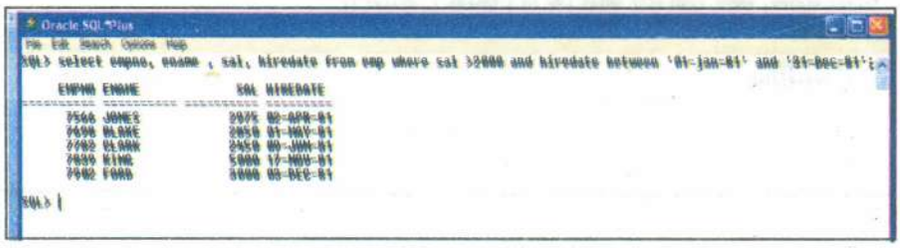


Fig. 19: Use of Relational and Logical Operators

2.4 SQL FUNCTIONS AND GROUPING

There are predefined set of functions in SQL. A function is defined as a set of statements/Instructions that perform some task and return the value. Only one value can be returned by the function. The functions in SQL that are applied on individual row is called as Single Row Functions and the functions that are applicable on group of rows are termed as Multi Row Functions. Amongst single row functions category, the functions covered in this book are:

a) Character Functions

b) Numeric Functions

Multi Row functions cover the following functions category:

a) Aggregate Functions

2.4.1 Character Functions

This is a category that works on character values or strings. In most functions, the accepted parameters are character value.

a) **LOWER**

- Syntax : Lower (char)
- Parameter : Character
- Return Value : Character
- Purpose : This function converts the string in lower case alphabets.
- Eg : SQL > SELECT LOWER(ENAME) "EMP NAME" FROM EMP;

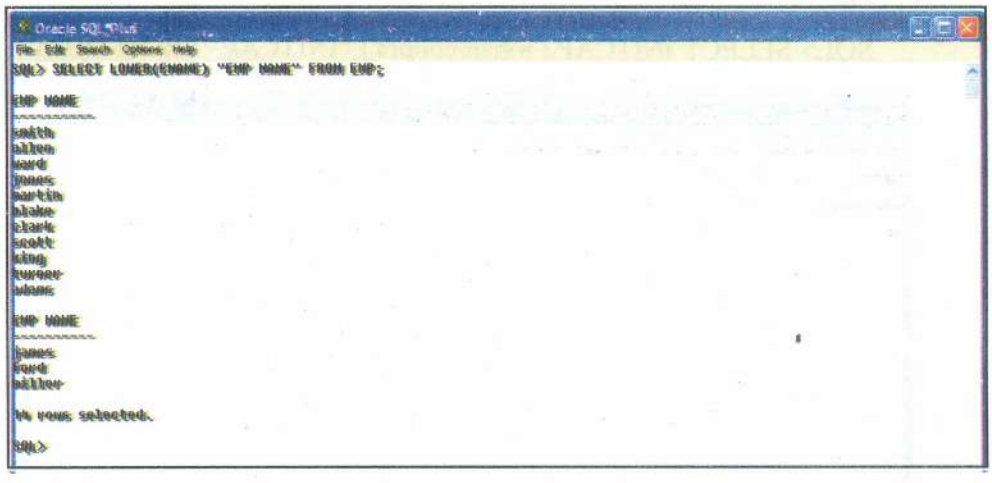


Fig. 20: Displaying names in lower case

b) **UPPER**

- Syntax : Upper (char)
- Parameter : Character
- Return Value : Character
- Purpose : This function converts the string in upper case alphabets.

c) **CONCAT**

- Syntax : Concat (string1, string2)

- Parameter : Character, Character
- Return Value : Character
- Purpose : This function appends string2 with string1 i.e. it concatenates two strings.
- Eg:

  SQL> SELECT EMPNO, CONCAT(ENAME, JOB) FROM EMP WHERE SAL > 3000;

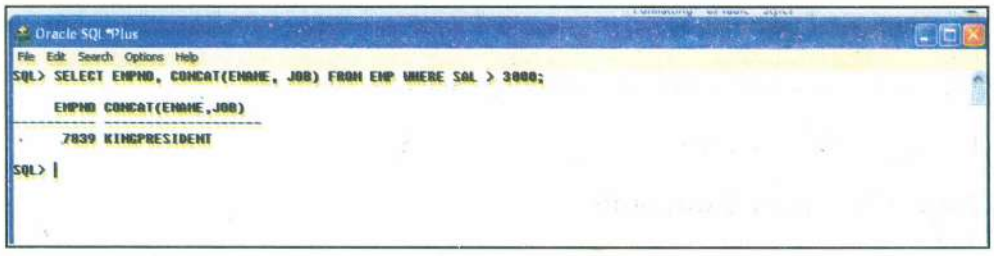


Fig. 21: Displaying Concat Function

d) **INITCAP**

- Syntax : InitCap (char)
- Parameter : Character
- Return Value : Character
- Purpose : This function converts the string to one with initial capitalized letter.
- Eg:

  SQL> SELECT INITCAP ('rohan chopra') "INITCAP" FROM DUAL;

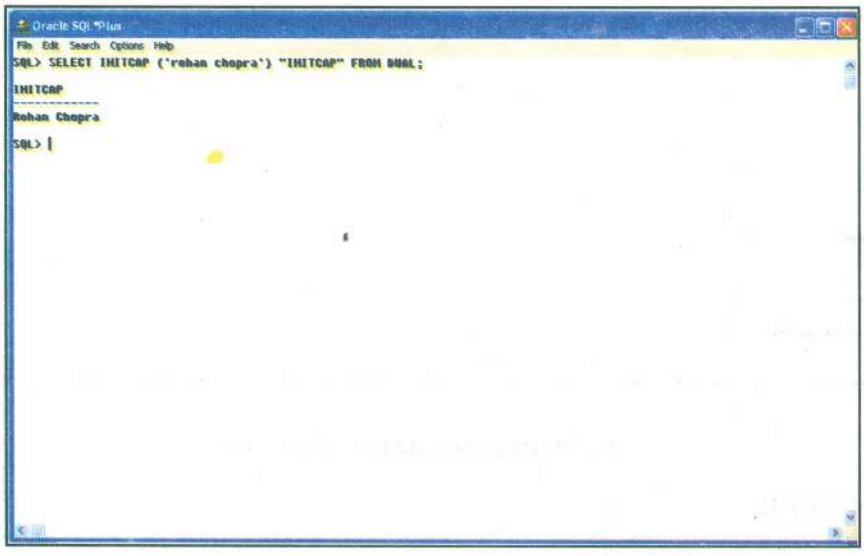


Fig. 22: Displaying Initcap Function

e) **LPAD**

- Syntax : Lpad (char1, n, char2)
- Parameter : Character, numeric, character
- Return Value : Character

- Purpose: This function uses three parameters. It takes the first parameter, fixes the width (i.e. no. of columns for the output) with n and pads the left spaces with char2.

- Eg:

  SQL> SELECT LPAD('876547',10,'*') "CHEQUE AMT" FROM DUAL;

  SQL>SELECT LPAD('87,6547',10,'*#') "CHEQUE AMT" FROM DUAL;

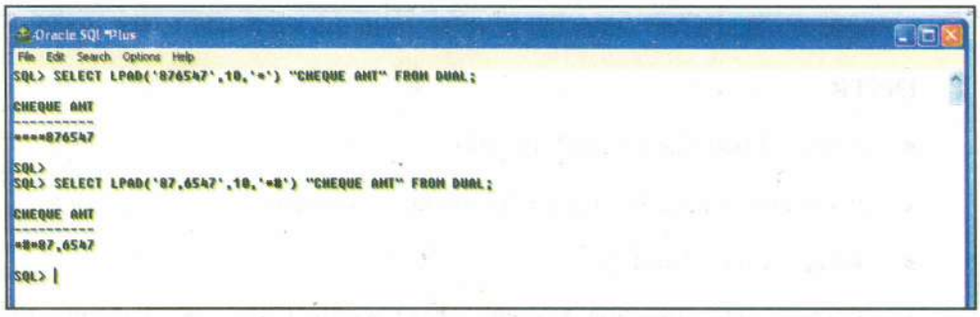


Fig. 23: Displaying LPAD function

f) **RPAD**

- Syntax: Rpad (char1, n, char2)

- Parameter: Character, numeric, character

- Return Value: Character

- Purpose: This function uses three parameters. It takes the first parameter, fixes the width (i.e. no. of columns for the output) with n and pads the right spaces with char2.

g) **SUBSTR**

- Syntax: Substr(char1, n1, n2)

- Parameter: Character, numeric, numeric

- Return Value: Character

- Purpose: This function uses three parameters. It extracts the string from a string. It takes the first parameter as its main string, n1 represents the position number to start extracting the string and n2 represent the no. of characters to be extracted.

- Eg:

  SQL> SELECT SUBSTR ('POSITION',2,3) FROM DUAL;

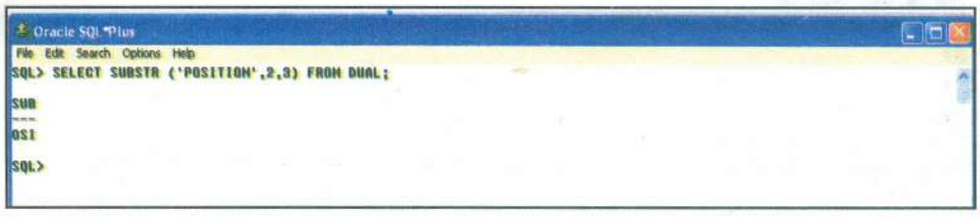


Fig. 24: Displaying SUBSTR function

SQL> SELECT SUBSTR ('POSITION',-2,3) FROM DUAL;

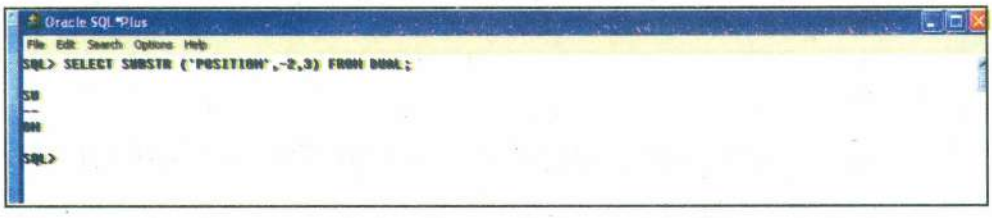


Fig. 25: Displaying SUBSTR function

Note: Negative value of n1 starts the position from the right side of the string.

h) **INSTR**

- Syntax: Instr(char1, char2, n1, n2)
- Parameter: Character, Character, numeric, numeric
- Return Value: Numeric
- Purpose: This function searches char2 within Char1. N1 indicates the position to begin the search in char1 and n2 parameter indicates the nth occurrence of char2.It returns the number of the position of char2 in char1. If n1 is negative, Oracle starts searching from right, n2 is always positive.
- Eg:

  SQL> SELECT INSTR ('PEPSICO COCO 'COLA','CO',7,3) FROM DUAL;

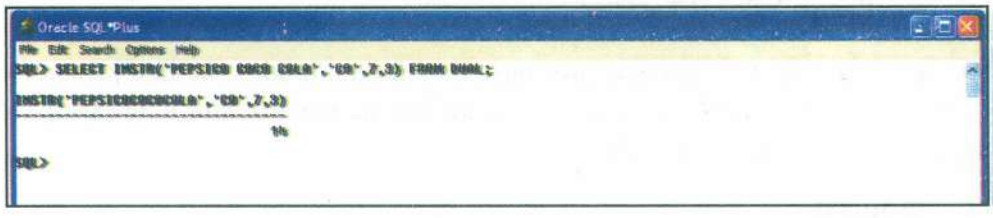


Fig. 26: Displaying INSTR function

i) **LTRIM**

- Syntax : Ltrim (char1, char2)
- Parameter : Character , Character
- Return Value : Character
- Purpose : This function truncates char2 from the left side of char1
- Eg:

  SQL> SELECT LTRIM('PEPSI','PEP') FROM DUAL;

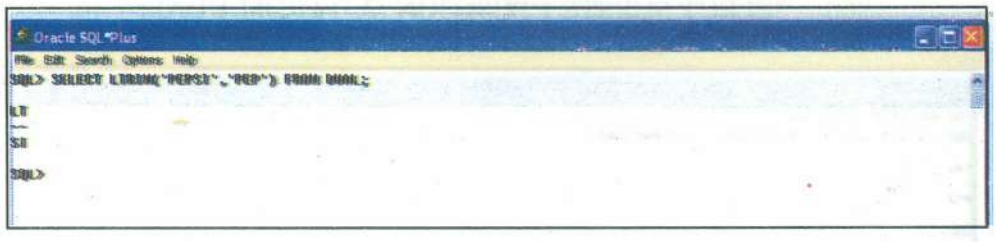


Fig. 27: Displaying Ltrim function

j) **RTRIM**

- Syntax : Rtrim (char1, char2)
- Parameter : Character , Character
- Return Value : Character
- Purpose : This function truncates char2 from the right side of char1

k) **LENGTH**

- Syntax : Length(char)
- Parameter : Character
- Return Value : Numeric
- Purpose : This function returns the number of characters in the string
- Eg:

  SQL> SELECT LENGTH('ROHAN CHOPRA') "LENGTH" FROM DUAL;

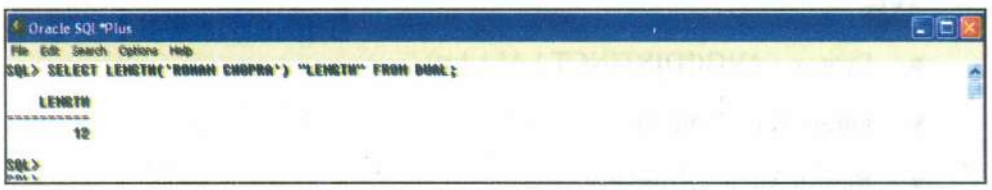


Fig. 28: Displaying Length function

2.4.2 Numeric Functions

These functions accept numeric values and after processing return a numeric value.

a) **MOD**

- Syntax : Mod(number1, number2)
- Parameter : Numeric, Numeric
- Return Value : Numeric
- Purpose : This function returns the remainder by dividing number1 by number2.

b) **SIGN**

- Syntax : Sign(number)
- Parameter : Numeric
- Return Value : Numeric
- Purpose : This function returns the sign of the number. It returns –1 if number <0 , 1 if number is > 0 and 0 if number =0

c) **POWER**

- Syntax : Power(number1, number2)
- Parameter : Numeric, Numeric
- Return Value : Numeric
- Purpose : This function returns number1 raised to the power number2.

d) **SQRT**

- Syntax : SQRT(number)
- Parameter : Numeric
- Return Value : Numeric

  Purpose : This function returns square root of the number

e) **ROUND**

- Syntax : Round(number1, number2)
- Parameter : Numeric, Numeric
- Return Value : Numeric

  Purpose : This function returns the parameter number1 rounded to number2.

Multi Row Functions

2.4.3 Aggregate Functions

a) **AVG**

- Syntax : AVG([DISTINCT | ALL] n)
- Parameter : Numeric
- Return Value : Numeric
- Purpose : This function returns average value of the parameter i.e. n
- Eg.

  SQL> SELECT AVG(SAL) FROM EMP;

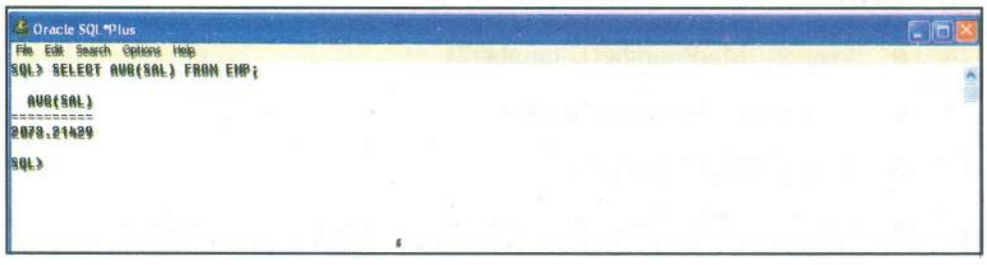


Fig. 29: Average function

b) **COUNT**

- Syntax : COUNT(*|[DISTINCT | ALL] n)
- Parameter : Numeric
- Return Value : Numeric
- Purpose : This function returns the counted number of rows.* parameter indicates all rows whether duplicate or null.

c) **MAX**

- Syntax : MAX([DISTINCT | ALL] n)
- Parameter : Numeric

- Return Value : Numeric
- Purpose : This function returns the maximum values from a group of values

d) **MIN**

- Syntax : MIN([DISTINCT | ALL] n)
- Parameter : Numeric
- Return Value : Numeric
- Purpose : This function returns the minimum values from a group of values

e) **SUM**

- Syntax : SUM([DISTINCT | ALL] n)
- Parameter : Numeric
- Return Value : Numeric
- Purpose : This function returns the sum of a group of values

SQL> SELECT SUM(SAL) FROM EMP;

Fig. 30: Sum function

2.4.4 GROUPING – Group By Clause

This clause is used to combine all the fields that have identical values in a particular field or group of fields. This divides a table into two or more groups.

SQL> SELECT JOB, COUNT(*) FROM EMP GROUP BY JOB;

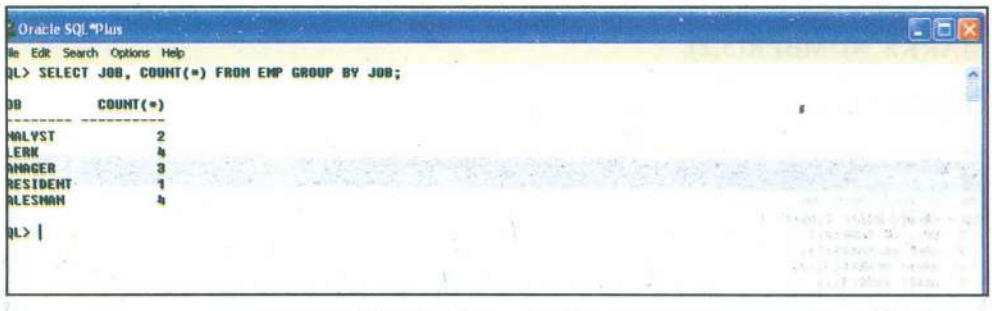


Fig. 31: Group By Clause

SQL> SELECT DEPTNO, COUNT(*) FROM EMP GROUP BY DEPTNO;

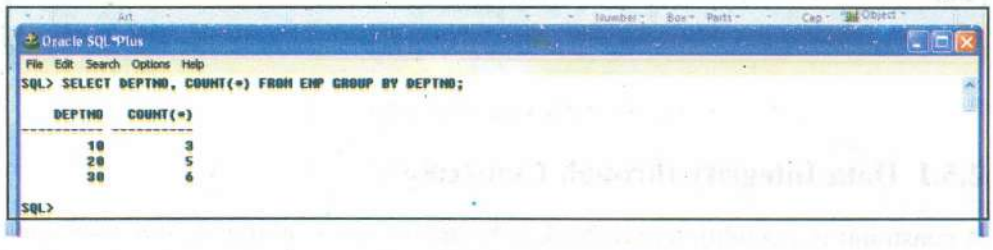


Fig. 32: Group By Clause

Placing condition on Group By – use of Having clause

The having clause is used to place conditions on the group by clause.

Eg:

SQL> SELECT JOB, COUNT(*) FROM EMP GROUP BY JOB HAVING COUNT(*)<=3;

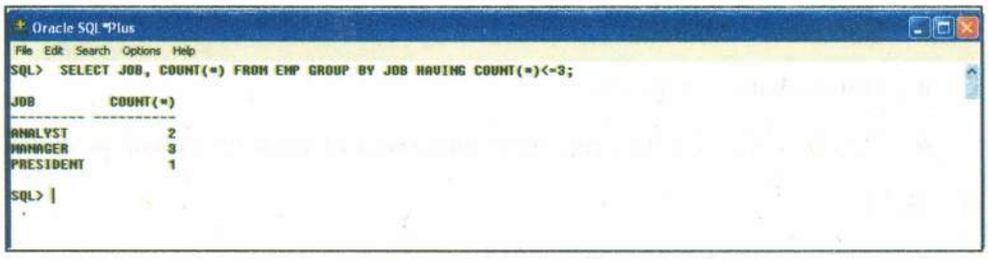


Fig. 33: Having Clause

2.5 DDL

This stands for Data Definition Language. It is this language that decides upon the database schema. The creation/alteration/deletion of all database objects is done by Data Definition Language. This is also used to apply some constraints on the fields of the table so as to maintain the data integrity.

Creating a table in a database

Syntax:

CREATE TABLE TABLENAME (COLUMNNAME1 DATATYPE (SIZE), COLUMNNAME2 DATATYPE (SIZE), COLUMNNAME3 DATATYPE (SIZE), ...);

CREATE TABLE STUDENT (

ROLL_NO NUMBER(3),

NAME VARCHAR2(20),

MARKS NUMBER(5,2),

GRADE CHAR(1));

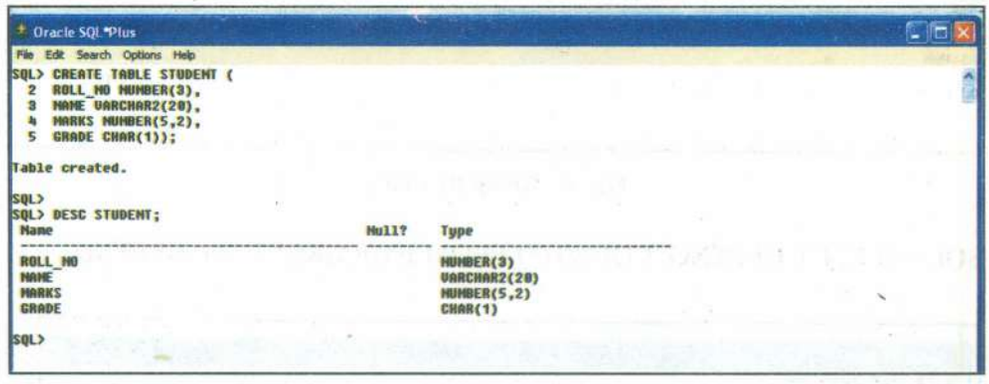


Fig. 34: Table created and described

2.5.1 Data Integrity through Constraints

A constraint is a condition or a check to be applied on a column(s) to maintain the integrity of data.

a) **Primary Key:** It is used to declare the primary key of the relation. This is done by adding PRIMARY KEY keywords while declaring the table.

CREATE TABLE STUDENT (

ROLL_NO NUMBER(3) PRIMARY KEY,

NAME VARCHAR2(20),

MARKS NUMBER(5,2),

GRADE CHAR(1));

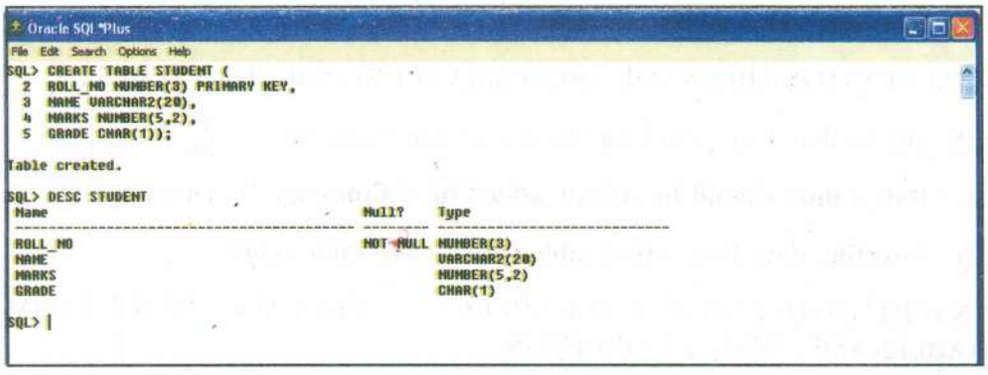


Fig. 35: Primary Key Constraint

b) **Default:** This is used to supply default values to the column in case no value is supplied by the user.

CREATE TABLE STUDENT (

ROLL_NO NUMBER(3),

NAME VARCHAR2(20),

MARKS NUMBER(5,2),

GRADE CHAR(2) DEFAULT ='E');

c) **Check:**

This constraint is used to check the values entered.

d) **Foreign Key:**

This checks whether the relationships between the related tables exists properly.

2.5.2 Create Table, Alter Table, Delete Table Command

Syntax:

ALTER TABLE TABLENAME ADD | MODIFY (COLUMNNAME DATATYPE (SIZE) CONSTAINT (IF ANY));

DELETE TABLE TABLENAME;

2.5.3 Viewing Table Structure

The command is as follows:

DESC[RIBE] TABLENAME;

2.6 DML

This stands for Data Manipulation Language. It is used to INSERT/UPDATE/ DELETE the rows of data in the database.

Insert/Update/Delete commands

Syntax:

a) **To insert a new row in the table**

INSERT INTO TABLENAME VALUES(VALUE1, VALUE2, VALUE3...);

b) **Inserting value at the run time**

INSERT INTO TABLENAME VALUES(&VALUE1, VALUE2, VALUE3...);

The ampersand (&) prompts for the user to input values.

c) **Date values should be strictly added in 'dd-mmm-yy' format.**

d) **Inserting data from other table which have same schema**

INSERT INTO TABLENAME1 SELECT * | COLUMN NAMES FROM TABLENAME2 WHERE CONDITION;

e) **Update command**

Syntax :

UPDATE TABLE TABLENAME SET COLUMNNAME = VALUE WHERE CONDITION;

f) **Delete Command**

Syntax:

DELETE FROM TABLENAME WHERE CONDITION;

2.7 HACKING ATTEMPT

Database Hacking

In the present business industry' scenario, the major concern is about database hacking. The main question that comes in mind of most people is whether or not to give their employees the role of protecting sensitive corporate data. Recent studies have indicated that 80% of the security breach of data mainly involves employees, insiders or those having internal access to the organization, which puts the information at risk. The main challenge that most companies face today is to maintain a proper balance between protecting sensitive information as much as possible and providing appropriate access to their workers, in addition to prevention of hacking. This is mainly because internet and e-mail have made the distribution and sharing of information relatively easier than ever.

Conventionally, database administrators are mainly assigned the role of proper administration of data to handle such situations or are granted multiple system prerogatives. In addition, the DBA also gets to enjoy unbridled access to the company system, in order to manage the IT infrastructure of the company 24x7 and also, to react to emergency situations. Even as firms continue to streamline operations and consolidate databases for maximizing both protection and efficiency of data from external threats such as hacking, the role-based and user-based security model does not comply with the "need-to-know" protection best-practices.

Multi-factored Model for Preventing Hacking

A multi-factored approach is mostly built on the principle of defense-in-depth which mostly inaugurates the multiple mechanisms for augmenting the role security model and the traditional user. This would mean the setting up of restrictions, controls and boundaries such that, those employees having database access privileges cannot freely alter, use or export important sensitive information. Most of these mechanisms are grouped into rules, realms, policies and roles. Most realms are established for encapsulating a set of database objects or an existing application within a protection zone. The one advantage of the consolidated database is the increased economies of scale and the elimination of information silos. However, at the same time, the information comprised in a single database mainly requires different protection levels from hacking. The other mechanism comprises of rules. Based upon the needs and requirements, the rules are further restricted. These are mainly accomplished with the help of domain specific decision factors or the environmental factors such as the authentication models, the time-of-day and the IP address.

Policies of System for Prevention of Hacking

The type of content contained within the structure is defined by the schema of a database. With the advent of new technologies, even the security administrators can set restrictions for preventing hacking of any sorts. With the separation of the data management and the schema within the database system, the system policy further backs the segregation of duties principle. This helps the database administrators to perform their duties while entrusting the security administrator to protect the infrastructure of the database and thereby prevent hacking

Database Vulnerabilities

Vulnerabilities with respect to computer security implies a weakness possessed by the system, which permits the attacker an opportunity to infringe the integrity, confidentiality, availability, access power, audit mechanisms or consistency of data or system and functions it hosts. Vulnerabilities are commonly the outcome from the design faults or the bugs of a system. The significance of vulnerabilities is very crucial at the time, when program bearing the vulnerability functions along with the special rights performing authentication or perhaps effortless entrance to data, user or any facilities as such RDBMS or server. Concept of a computer language is termed with the word vulnerability, while several program flaws root cause is owing to their use. Vulnerabilities usually rise owing to the carelessness attitude adopted by its programmers. Although, there can be other reasons for the same. Vulnerability let the attacker mistreat the application, for instance going around the admission control checks or perhaps even carrying out a command to the hosting system application.

Disclosing Vulnerabilities

Technique used to disclose the vulnerabilities is a debatable topic in the community of computer security. Few of the people urge that complete disclosure of the vital information related to vulnerabilities, subsequent to the discovery is the problem. Whereas, few people argue that restricting disclosure to users pose great risk, the complete details are only issued after a delay sometimes. The time given owing to the delay permits the notified ones to mend the problem via developing as well as applying patches. This will in fact, heighten the risk for people, who are not secluded to complete details. From the point of view of the security, it is very essential to do free as well as public disclosure, in order to make certain that all parties interested are served with the appropriate information. To provide security by the means of obscurity is regarded by the experts as a concept which is most unreliable. The concept needs to be impartial so as to allow reasonable distribution of security important information. Very often, it is regarded that a channel which is extensively

accepted as a source of securing information in industry circles is the most trusted channel. For instance: SecurityFocus and FrSIRT.

Discovery and Removal of Vulnerabilities

Numerous software tools survive, which can help in uncovering the vulnerabilities of the computer systems. Although, such tools can help the auditor in getting a proper summary of the potential vulnerabilities exhibiting, they simply cannot substitute the human discernment. By relying completely on the scanners can output sham positives and also, a restricted overview of problem persisting in the computer system. The vulnerabilities are discovered in majority of the operating system such as Mac OS, Windows, few forms of Linux and UNIX. One way by which, the vulnerability occurrence can be reduced is through constant vigilance. Few instances of vulnerabilities are: symlink races, stack smashing as well as buffer overflows, validation error of input as such SQL injection, directory traversal. Session Hijacking as well as distant Code Execution are also examples of vulnerabilities.

Check Your Progress 2

Notes: a) Space is given below for writing your answer.

b) Compare your answer with the one given at the end of this Unit.

1) Consider the following table Teacher and create SQL queries for the points that follow:

Table: Teacher

Teacher_No	T_Name	Age	Department	HireDate	Salary	Gender
IIT-DEL-0001	SANJIV CHOPRA	32	COMPUTER	01-JAN-2010	45000	MALE
IIT-DEL-0002	MUGDHA	25	ELECTRONICS	09-DEC-2010	20000	FEMALE
IIT-DEL-0003	PRIYANKA	44	MECHANICAL	14-JUL-2003	55450	FEMALE
IIT-DEL-0004	SONAM	43	COMPUTER	25-FEB-2009	52500	FEMALE
IIT-DEL-0005	PRITAM SINGH	51	CIVIL	11-AUG-2007	70000	MALE
IIT-DEL-0006	AMITABH RANA	27	ELECTRONICS	11-AUG-2007	25000	MALE
IIT-DEL-0007	KAUSTUBH	28	INFORMATION TECHNOLOGY	23-MAR-2009	29000	MALE
IIT-DEL-0008	APARNA SHAH	42	COMPUTER	01-JAN-2010	43000	FEMALE
IIT-DEL-0009	MONIKA SHARMA	41	ELECTRONICS	08-JUN-2011	34750	FEMALE
IIT-DEL-0010	AVINASH SINGHAL	47	CIVIL	17-DEC-2010	50000	MALE

a) Display all the records from the table.

.....

.....

.....

.....

b) Display TeacherNo, Teacher Name, Salary of those teachers who are in computer department.

.....

.....

.....

.....

c) Display all the information of Female teachers.

.....

.....

.....

.....

d) Display records of all the teachers with their Hiredate in ascending order.

.....

.....

.....

.....

e) Display list of all male teachers who belong to Civil and Mechanical departments.

.....

.....

.....

.....

f) Display names of only those teachers who have salary more than 30000.

.....

.....

.....

.....

g) Display only distinct salary values.

.....

.....

.....

.....

h) Display Name, Salary and Bonus for all the teachers, if bonus is 20% of the salary of the teacher.

.....

.....

.....

.....

i) Display Names of teachers who joined in the year 2011.

.....

.....

.....

j) Display all the names starting with letter S.

.....

.....

.....

.....

k) Display all the records of teachers having their names ending with A.

.....

.....

.....

.....

2.8 LET US SUM UP

This unit deals with hand on experience on the database. It starts with the concepts of oracle architecture. The methods to open oracle and SQL*PLUS which is the command line interpreter. It handles different types of queries that can be handled on Oracle database. It gives information about all the commands that are most commonly used in Oracle Sql database. The different types of hacking attempts are also a raised issue. It also talks about data dictionary and database objects that exist in the database. All in all it is a very useful unit in terms of hands on database.

2.9 CHECK YOUR PROGRESS: THE KEY

Check Your Progress 1

1) SQL stands for Structured Query Language. It has a proper structured syntax which is used to query/insert/update the database.

2) **Data Definition Language**

It is a type of language which is responsible for creating and setting the database schema consisting of relations, views, stored procedures, functions etc. Some of the DDL Commands include:

- Create/Alter/Drop Schema objects
- Grant and Revoke Commands for granting and revoking privileges

Data Manipulation Language

Data Manipulation Language (DML) includes retrieval of information/inserting records/deletion of records/modification of data etc. Some of the commands from the command set include:

- Insert/Update/Delete commands
- Select and its variations etc.

3) **Data Dictionary** – It is a repository of data about data i.e. metadata.

Upon creation of the database objects, the details are stored in Data Dictionary. For any further modification in the definition of database objects, Data Dictionary is consulted.

Data Dictionary gives the inside view and the structure of the Oracle database. It contains information and details about database objects, their logical structure, their relationships amongst themselves etc. The views of data dictionary are divided into three general categories:

- User
- All
- DBA

4)

DATATYPE	DESCRIPTION
CHAR(SIZE)	FIXED LENGTH CHARACTER STRING WITH SIZE BYTES
VARCHAR2 (SIZE)	TO STORE VARIABLE LENGTH CHARACTER STRING WITH MAXIMUM SIZE SPECIFIED WITH SIZE ATTRIBUTE
NUMBER (P,S)	TO STORE NUMERIC VALUES WITH PRECISION P RANGING FROM 1-38 AND S SCALE IN BETWEEN RANGE -84 TILL 127
DATE	VALID DATE IN DD-MON-YYYY FORMAT
LONG	VARIABLE LENGTH CHARACTER DATA UPTO FEW GIGA BYTES
RAW(SIZE)	IT STORES BINARY DATA OF LENGTH SIZE BYTES
LONG RAW	VARIABLE LENGTH BINARY DATA UPTO FEW GIGA BYTES

5) **Processing capabilities of SQL**

The following are some of the capabilities of SQL:

- It provides commands to define or alter the database objects through Data Definition Language (DDL)
- It allows users of the database to insert/update/delete/query data from the database through simple Data Manipulation Language (DML).
- It provides security to the system through specifying access rights to database objects.
- It includes data integrity checking.
- It provides control over transaction processing through transaction Control Language (TCL).

6) Goto Start Button >> Programs >> Oracle – Orahome 92>> SQL*PLUS

Check Your Progress 2

1) a) Display all the records from the table.

SQL> SELECT * FROM TEACHER;

b) Display TeacherNo, Teacher Name, Salary of those teachers who are in computer department.

SQL> SELECT TEACHERNO, T_NAME, SALARY FROM TEACHER WHERE DEPARTMENT = 'COMPUTER';

c) Display all the information of Female teachers.

d) Display records of all the teachers with their Hiredate in ascending order.

SQL> SELECT TEACHERNO, T_NAME, AGE, DEPARTMENT, HIREDATE, SALARY FROM TEACHER WHERE GENDER='FEMALE';

e) Display list of all male teachers who belong to Civil and Mechanical departments.

SQL> SELECT T_NAME FROM TEACHER WHERE GENDER = 'MALE' AND DEPARTMENT IN ('MECHANICAL','CIVIL');

f) Display names of only those teachers who have salary more than 30000.

SQL> SELECT T_NAME FROM TEACHER WHERE SALARY > 30000;

g) Display only distinct salary values.

SQL> SELECT DISTINCT (SALARY) FROM TEACHER;

h) Display Name, Salary and Bonus for all the teachers, if bonus is 20% of the salary of the teacher.

SQL> SELECT T_NAME, SALARY, SALARY + SALARY * 0.2 "BONUS" FROM TEACHER;

i) Display Names of teachers who joined in the year 2011.

SQL> SELECT T_NAME FROM TEACHER WHERE HIREDATE > '01-JAN-2011';

j) Display all the names starting with letter S.

SQL> SELECT T_NAME FROM TEACHER WHERE T_NAME LIKE 'S%';

k) Display all the records of teachers having their names ending with A.

SQL> SELECT T_NAME FROM TEACHER WHERE T_NAME LIKE '%A';

UNIT 3 DATABASE SECURITY- I

Structure

3.0 INTRODUCTION

Distributed Databases

Distributed databases is defined as databases located at different machines at the same or different locations that looks like one centralized database to the end user. Thus, instead of having one centralized database bear the entire load, it is shared by a collection of machines/computers. It is actually a set of server machines working in synchronization to cater the needs to multiple users. These machines in a distributed system are connected to each other either through wireless connection or through various communication media that serve data transfer at high rate. The machines don't have a shared memory nor do they share a clock. The processors in the distributed system may vary from microcomputers to work station to mini computers to computers used in day to day life. The distributed database can be shown as:

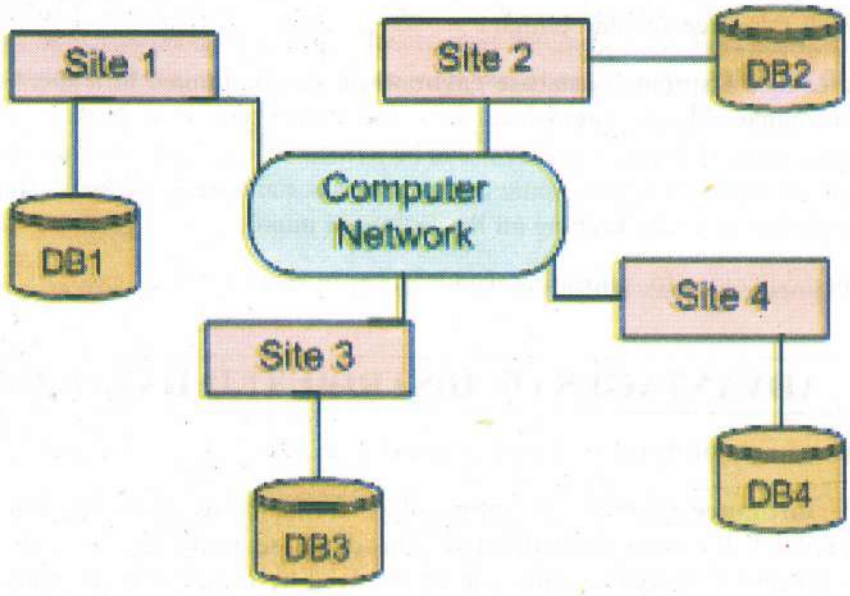


Fig. 1: Distributed Database

Why Distributed Databases?

Distributed databases are useful nowadays as many branches of the organisations are geographically separated. Thus, accessing a centralised database at one location may cause many issues such as slow accessibility, session's time out, inefficiency, no load sharing etc. Thus, to have more efficient system, concept of distributed databases provides proves to be more lucrative.

3.1 OBJECTIVES

After studying this unit, you should be able to:

- explain Distributed Database;
- explain Distributed Database Design;
- list the advantages of Distributed Database Design;
- distinguish distributed over Centralised databases; and
- list the disadvantages of distributed databases.

3.2 DISTRIBUTED DATABASE DESIGN

The design of the distributed database should be such that it meets its requirements and its purpose for which it is meant to be. As already mentioned, in a distributed database each site can perform the local query transaction or can participate in global query transaction as well. The global query is the one that requires machines at multiple sites to participate as data needs to be sent from all these machines so as to complete the transaction. The sites can be connected to each other through different topologies such a Bus, Star, Tree, Ring, Mesh etc. But the choice of connection depends on the following factors:

a) Installation cost should be low

b) Communication cost should be low

c) Reliability should be high

d) Availability should be high

e) Fault tolerance should be high

The sites in a distributed database environment can be limited to a small are or may encompass a huge geographical area. The former type of network is called as local area network whereas the latter can be termed as long haul network. At times lon haul networks may pose some communication and speed problems. Thus, the design should be made keeping all the factors in mind.

The methodologies for allocating data is given in section 3.4

3.3 ADVANTAGES OF DISTRIBUTED DATABASES

The advantages of distributed databases are as follows:

a) **Performance:** It leads to improved performance as many machines are involved, the load is distributed. The database is divided into database fragments, thus local queries can be resolved by local databases rather than all queries being targeted to one centralised database. Thus, the query processing time is reduced and performance is increased.

b) **Sharing:** Data at multiple sites is shared by users at different sites.

c) **Robustness:** The entire system becomes more robust as multiple servers are involved in handling data. Thus, failure of one system doesn't lead to failure of entire system.

d) **Availability:** The data is replicated at multiple sites. In case the local server is unavailable due to some reason, the data can be retrieved from the other available server.

e) **Multiple query evaluation:** This type of system leads to multiple query evaluation together. Thus, resulting in high performance.

f) **Ease of growth:** To add more clients to such a system is quite easy as overloading is never an issue.

g) Management of distributed data with different levels of transparency.

h) Hardware, Operating System, Network and Location Independence.

i) It provides Continuous operation.

j) No more reliance on the central site.

3.4 METHODOLOGIES FOR ALLOCATING DATA

In distributed databases, the database is divided into different logical units of data called as data fragments. This process is termed as data fragmentation. These fragments can be stored at different locations or some fragments may be stored at more than one location.

3.4.1 Data Fragmentation

In this type of fragmentation, a relation may be divided into different pieces or fragments based on their:

i) **Horizontal Fragmentation:** In this type of fragmentation certain tuples/ records satisfying one type of condition can be used to generate a horizontal subset of the relation. This subset can be stored at one location and similarly other subsets created can be stored at other locations.

ii) **Vertical Fragmentation:** In this type of fragmentation certain attributes (most commonly used) of the relation can be stored at one location whereas other attributes (less commonly used) can be stored at the other location.

iii) **Mixed fragmentation**: a procedure that follows a mix of the above two techniques can be used to do so.

3.4.2 Data Replication

This includes storing same piece of data at more than one location. This can be done using any of the following methods:

a) **Partial Replication:** In this method, some fragments are stored at multiple locations. This is usually done to make some critical data available to all the users located at multiple sites.

b) **Full Replication:** The entire database is replicated at multiple sites. This is done to maintain the full backup of the system. But this may at times lead to slow down of the processing.

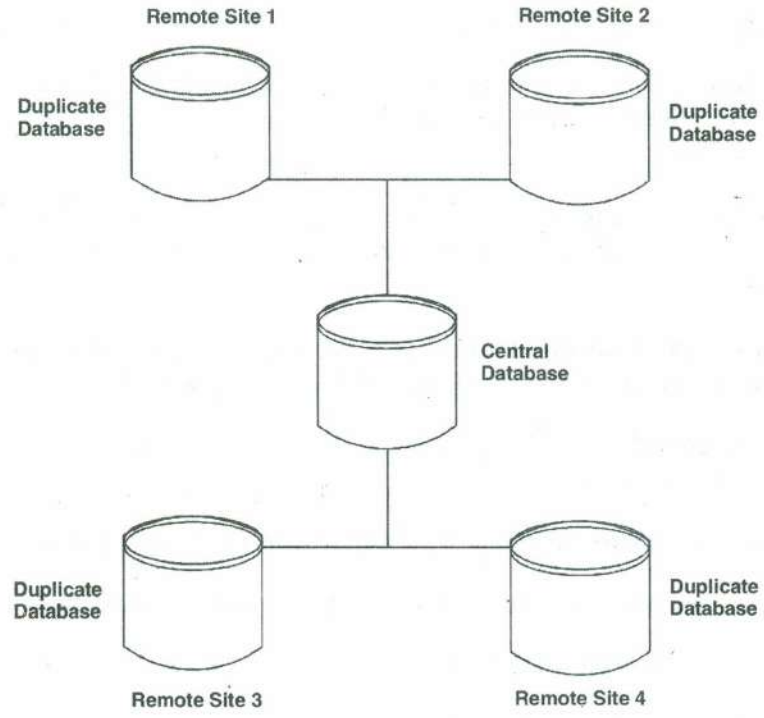


Fig. 2: Full Replicated Database

c) **No replication:** in this type of system, no replication of data is done at multiple sites. Thus, a fragment stored at one site is not stored at any other site.

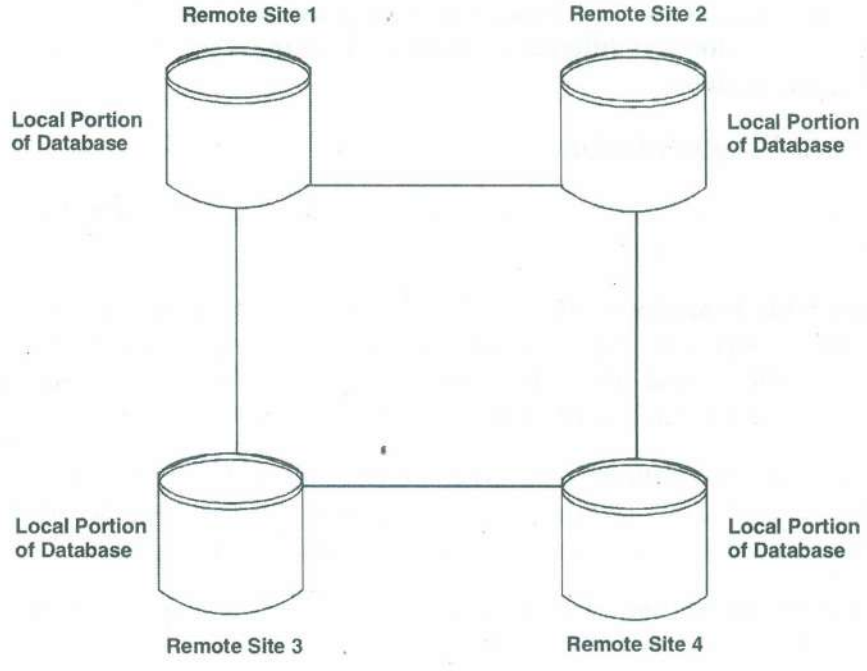


Fig. 3: No replication

3.5 DISADVANTAGES OF DISTRIBUTED DATABASES

a) It may lead to a complex system

b) The cost to manufacture such a system may turned out to be really high

c) More machines in the system also means more security standards to be incorporated

d) Some sites may not strictly adhere to data storage standards

e) In a distributed database, enforcing integrity over a network may require too much of the network's resources to be feasible.

3.6 CENTRALIZED DATABASE

Centralized Database Systems

Concept

It consists of one large system located at one site having several CPU's and the devices controller are connected through a common bus to provide shared memory. CPU and the device controllers can execute concurrently and share the same memory unit. There are divided into two ways in which computers are used:

- Single user system
- Multi user system

A typical single user system is a desktop unit used by single person usually with one or two hard – disks and used by single user at a time.

On the other hand, multi user systems have more than one disk, more memory and multiple CPU's and multi user operating system that can be used by large number of people at the same time.

Benefits

- **Data Integrity –** the single greatest benefit of centralizing data management is data integrity. One of the cardinal rules of database design is that no redundancy is allowed. That is, no piece of data should ever be repeated within the database. When an organization is operating multiple databases for the same group of people (for example, a membership database with a separate meeting registration database), they are by definition breaking this rule. And this leads to major data integrity issues. A centralized database means that each member has one primary record, with primary contact information. Thus when there is a change required (like a new phone number or e-mail address), there is only one place to look to make these changes.
- **Valuable broad marketing info/history –** with all the information centralized, it is much easier to develop reports that show the broad range of activities that your members are engaged in. With multiple databases, records need to be matched, de-duping needs to occur and the opportunity for duplicate records is greatly increased.
- **Ease of training (it's the same system for everything) –** another benefit of a centralized system is that the learning curve for users is greatly reduced. If all processes (membership, meetings, products, etc.) are in the same database, then users need only learn one system, not multiple systems.
- **Support –** With a centralized system, support is focused on one product. With many databases, even if they are built on the same platform, separate support is required for each.

Disadvantages

- Lack of cooperation from managers, who do not like to be under control of centralised Data Processing department.
- Resistance from managers for mechanising the data processing activities relating to their various functions.
- It is difficult to provide equitable services to various departments.
- The data security is also questioned.

3.7 DATABASE SECURITY – DISTRIBUTED VS. CENTRALIZED DATABASE

Security Issues of Centralized Database Systems

Three interrelated technologies are used to achieve information confidentiality and integrity in traditional DBMSs which are authentication, access control and audit.

Authentication identifies one party to another and it can be performed in both directions. It ensures the true identity of a user, computer or process. Once the identity is established, the party can access data under the guidance of the access control service. Access control regulations are set by the system security administrator and define who can access what data with which privileges. However, authentication and access control do not comprise a complete security solution - they must be complemented by an auditing service. An auditing service records important user activities in system logs for real-time or a posteriori analysis. Real-time auditing is often referred to as intrusion detection. An audit service protects a system in three ways: detecting actual security violations; assisting the security administrator in discovering attempted attacks by recognizing abnormal activity patterns; and detecting possible system security flaws.

Security Issues of Distributed Database Systems

In developing a distributed database, one of the first questions to answer is where to grant system access i.e. Users are granted system access at their home site or at the remote site. Probably the most glaring is the additional processing overhead required when granting the access at remote site, particularly if the given operation requires the participation of several sites. Furthermore, the maintenance of replicated clearance tables is computationally expensive and more prone to error. Finally, the replication of passwords, even though they're encrypted, increases the risk of theft.

As in the centralized relational database, access control in the distributed environment is accomplished with the view. Instead of developing the view from local relations, it is developed from the global relations of the distributed database. Accordingly, it is referred to as a global view. The view mechanism is even more important in the distributed environment because the problem is typically more complex (more users and a more complex database) and while centralized databases may not be maintained as multilevel access systems, a distributed database is more likely to require the suppression of information.

Although global views are effective at data suppression and to a lesser extent at inference protection, their use can be computationally expensive. One of the key problems with a relational distributed database is the computation required to execute a complex query (particularly one with several JOINs, which join tables and table

fragments that are stored at geographically separate locations). Since each view is unique, a different query is necessary for each view. This additional overhead is partially offset by query optimizers. Nonetheless, the addition of global views adds computing time to a process that already takes too long.

Multilevel Constraint Processing in a Distributed Environment

As with the centralized model, inference engines are added to the standard distributed database architecture at each site. Their model assumes that the distributed database is homogeneous. In this case, the inference engines at the user's site processes the query and update constraints. Only a small amount of overhead is added. If the distributed database is heterogeneous, however, then the processing overhead would be prohibitively expensive since the inference engines at each site involved in the action would need to process the security constraints

for all the local data. Considering the processing demands already in place in a relational database management system (RDBMS), this appears to be impractical.

Check Your Progress 1

Note: a) Space is given below for writing your answers.

b) Compare your answers with the one given at the end of this Unit.

1) Define Distributed Database.

.....

.....

.....

.....

.....

2) Why do we need Distributed Database?

.....

.....

.....

.....

.....

3) What is Centralised Database system?

.....

.....

.....

.....

.....

4) Mention advantages of distributed database system over centralised database system and vice versa.

.....

.....

.....

.....

.....

5) How is security achieved in Centralised system?

.....

.....

.....

.....

.....

6) Are security measures very tight and difficult to implement in distributed databases?

.....

.....

.....

.....

.....

7) Define Data Fragmentation and Data Replication.

.....

.....

.....

.....

.....

3.8 LET US SUM UP

This chapter is very important as it talks about the Distributed databases, their advantages and disadvantages, distributed database design. The concept of centralised databases is also highlighted. How distributed databases are advantageous over centralised database is mentioned too. In the end it talks about disadvantages existing with distributed database design. Overall it is a chapter that covers security to the database by considering the two mechanisms of storing data.

3.9 CHECK YOUR PROGRESS: THE KEY

Check Your Progress 1

1) Distributed databases is defined as databases located at different machines at the same or different locations that looks like one centralized database to the end user.

2) Distributed databases are useful nowadays as many branches of the organisations are geographically separated. Thus, accessing a centralised database at one location may cause many issues such as slow accessibility, session's time out, inefficiency, no load sharing etc. Thus, to have more efficient system, concept of distributed databases provides proves to be more lucrative.

3) It consists of one large system located at one site having several CPU's and the devices controller are connected through a common bus to provide shared memory. CPU and the device controllers can execute concurrently and share the same memory unit. There are divided into two ways in which computers are used:

   - Single user system
   - Multi user system

4) The advantages of distributed databases are as follows:

   - **Performance:** It leads to improved performance as many machines are involved, the load is distributed. The database is divided into database

fragments, thus local queries can be resolved by local databases rather than all queries being targeted to one centralised database. Thus, the query processing time is reduced and performance is increased.

- **Sharing:** Data at multiple sites is shared by users at different sites.
- **Robustness:** The entire system becomes more robust as multiple servers are involved in handling data. Thus, failure of one system doesn't lead to failure of entire system.
- **Availability:** The data is replicated at multiple sites. In case the local server is unavailable due to some reason, the data can be retrieved from the other available server.
- **Multiple query evaluation:** This type of system leads to multiple query evaluation together. Thus, resulting in high performance.
- **Ease of growth:** To add more clients to such a system is quite easy as overloading is never an issue.
- Management of distributed data with different levels of transparency.
- Hardware, Operating System, Network and Location Independence.
- It provides Continuous operation.
- No more reliance on the central site.

5) Three interrelated technologies are used to achieve information confidentiality and integrity in traditional DBMSs which are authentication, access control and audit.

Authentication identifies one party to another and it can be performed in both directions. It ensures the true identity of a user, computer or process. Once the identity is established, the party can access data under the guidance of the access control service. Access control regulations are set by the system security administrator and define who can access what data with which privileges. However, authentication and access control do not comprise a complete security solution – they must be complemented by an auditing service. An auditing service records important user activities in system logs for real-time or a posteriori analysis. Real-time auditing is often referred to as intrusion detection. An audit service protects a system in three ways: detecting actual security violations; assisting the security administrator in discovering attempted attacks by recognizing abnormal activity patterns; and detecting possible system security flaws.

5) No with all the measures in design, the security aspects become easy to handle.

7) Data Fragmentation

In this type of fragmentation, a relation may be divided into different pieces or fragments based on their:

i) **Horizontal Fragmentation:** In this type of fragmentation certain tuples / records satisfying one type of condition can be used to generate a horizontal subset of the relation. This subset can be stored at one location and similarly other subsets created can be stored at other locations.

ii) **Vertical Fragmentation:** In this type of fragmentation certain attributes (most commonly used) of the relation can be stored at one location whereas other attributes (less commonly used) can be stored at the other location.

iii) **Mixed fragmentation:** a procedure that follows a mix of the above two techniques can be used to do so.

Data Replication

This includes storing same piece of data at more than one location. This can be done using any of the following methods:

i) **Partial Replication**: In this method, some fragments are stored at multiple locations. This is usually done to make some critical data available to all the users located at multiple sites.

ii) **Full Replication:** The entire database is replicated at multiple sites. This is done to maintain the full backup of the system. But this may at times lead to slow down of the processing.

UNIT 4 DATABASE SECURITY - II

Structure

4.0 INTRODUCTION

Many a times databases features failure and are not easily recoverable whereas some of the databases are capable of fault tolerance. Transaction processing is one of the really critically handled concepts

Database Transaction and Database Concurrency

Database transaction refer to a unit a unit of work that must occur or fail in its entirety i.e. it should make some change in the database or it must rollback all together. Thus, a transaction comprises a unit of work performed within a database management system (or similar system) against a database and treated in a coherent and reliable way independent of other transactions.

Database concurrency is a technique that provides control to each transaction and ensures that transactions occur following an order. The main jôb of these controls is to protect transactions issued by different users/applications from the effects of each other. All the transactions follow four simple characteristics **(ACID)** of database transactions: atomicity (A), consistency(C), isolation (I) and durability (D).

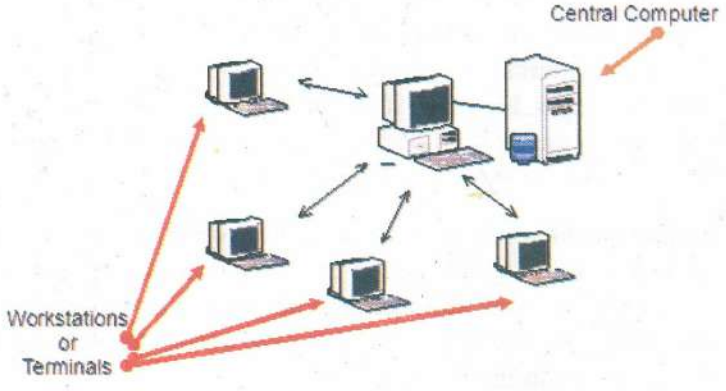


Fig. 1: Database Concurrency

Database transaction and the ACID rules

A database transaction is a unit of work, typically encapsulating a number of operations over a database i.e. reading a database object, writing, acquiring lock etc. an abstraction supported in database and also other systems. Every database transaction obeys the following rules (by support in the database system; i.e. a database system is designed to guarantee them for the transactions it runs):

- **Atomicity** – It is based on all or none concept i.e. either the effects of all or none of its operations when a transaction is completed (committed or aborted respectively). Thus, the transaction results in either done or never started.
- **Consistency** – Every transaction must leave the database in a consistent (correct) state, i.e. maintain the predetermined integrity rules of the database. A transaction must transform a database from one consistent state to another consistent state Thus, since a database can be normally changed only by transactions; all the database's states are consistent. An aborted transaction does not change the state.
- **Isolation** – All the transactions are independent. Transactions cannot interfere with each other. Thus, each transaction is unaware of the concurrently running transactions.
- **Durability** – Effects of successful transactions must persist through crashes.i.e. after the successful completion of the transaction, the changes to the database must persist even in the case with database failure.

4.1 OBJECTIVES

After studying this unit, you should be able to:

- understand Database transaction and concurrency;
- explain how to control database concurrency;
- explain failure recovery mechanisms in a Database;
- explain fault tolerance mechanisms; and
- explain theory of transactions.

4.2 DATABASE CONCURRENCE

Why is concurrency control needed?

Concurrency control in a database management systems (DBMS) concept that is used to address conflicts with the simultaneous accessing or altering of data that can occur with a multi-user system. Concurrency control, when applied to a DBMS, is meant to coordinate simultaneous transactions while preserving data integrity. If transactions are executed serially, i.e. sequentially with no overlap in time, no transaction concurrency exists. However, if concurrent transactions with interleaving operations are allowed in an uncontrolled manner, some unexpected, undesirable result may occur. Here are some typical examples:

1) **The lost update problem**: A second transaction writes a second value of a data-item on top of a first value written by a first concurrent transaction and the first value is lost to other transactions running concurrently which need, by their precedence, to read the first value. The transactions that have read the wrong value end with incorrect results.

2) **The dirty read problem**: Transactions read a value written by a transaction that has been later aborted. This value disappears from the database upon abort and should not have been read by any transaction ("dirty read"). The reading transactions end with incorrect results.

3) **The incorrect summary problem:** While one transaction takes a summary over the values of all the instances of a repeated data-item, a second transaction updates some instances of that data-item. The resulting summary does not reflect a correct result for any (usually needed for correctness) precedence order between the two transactions (if one is executed before the other), but rather some random result, depending on the timing of the updates and whether certain update results have been included in the summary or not. Concurrency control mechanisms.

4.2.1 Concurrency Control Mechanisms

The main categories of concurrency control mechanisms are:

- **Optimistic**

  Delay the checking of whether a transaction meets the isolation and other integrity rules until its end, without blocking any of its (read, write) operations and then abort a transaction to prevent the violation, if the desired rules are to be violated upon its commit. An aborted transaction is immediately restarted and re-executed, which incurs an obvious overhead. If not too many transactions are aborted, then being optimistic is usually a good strategy.

- **Pessimistic**

  Block an operation of a transaction, if it may cause violation of the rules, until the possibility of violation disappears. Blocking operations is typically involved with performance reduction.

- **Semi-optimistic**

  Block operations in some situations, if they may cause violation of some rules and do not block in other situations while delaying rules checking (if needed) to transaction's end, as done with optimistic.

4.3 METHODS OF DATABASE CONCURRENCY CONTROL

4.3.1 Methodologies

There are different methods of concurrency control mechanisms exist, some of the following are most commonly used ones:

1) **Locking** – Restricting the access to data by locks assigned to the data. This can be done by the other transaction to a data item to block the availability of the data item i.e. database object etc.

2) **Serializability** – This involves checking for cycles in the schedule's graph and breaking them by aborts.

3) **Timestamp ordering** – Assigning timestamps or time slices to transactions and controlling or checking access to data by timestamp order.

4) **Commitment ordering** – Controlling or checking transactions' chronological order of commit events to be compatible with their respective precedence order.

The most common mechanism type in database systems since their early days in the 1970s has been **Strong strict Two-phase locking** (SS2PL; also called Rigorous scheduling or Rigorous 2PL) which is a special case (variant) of both Two-phase locking (2PL) and **Commitment ordering** (CO). It is pessimistic. In spite of its long name (for historical reasons) the idea of the SS2PL mechanism is simple: "Release all locks applied by a transaction only after the transaction has ended". SS2PL (or Rigorousness) is also the name of the set of all schedules that can be generated by this mechanism, i.e. these are SS2PL (or Rigorous) schedules, have the SS2PL (or Rigorousness) property.

4.3.2 Major Goals of Database Concurrency Control Mechanisms

Concurrency control mechanisms firstly need to operate correctly, i.e. to maintain each transaction's integrity rules (as related to concurrency; application-specific integrity rule are out of the scope here) while transactions are running concurrently and thus the integrity of the entire transactional system. Correctness needs to be achieved with as good performance as possible. In addition, increasingly a need exists to operate effectively while transactions are distributed over processes, computers and computer networks. Other subjects that may affect concurrency control are recovery and replication.

Correctness

i) **Serializability**

For correctness, a common major goal of most concurrency control mechanisms is generating schedules with the Serializability property. **Serializability** of a schedule means equivalence to some serial schedule with the same transactions (i.e. in which transactions are sequential with no overlap in time and thus completely isolated from each other: No concurrent access by any two transactions to the same data is possible). Serializability is considered the highest level of isolation among database transactions and the major correctness criterion for concurrent transactions. In some cases compromised, relaxed forms of serializability are allowed for better performance or to meet availability requirements in highly distributed systems.

Almost all implemented concurrency control mechanisms achieve serializability by providing Conflict serializablity, a broad special case of serializability (i.e. it covers, enables most serializable schedules and does not impose significant additional delay-causing constraints) which can be implemented efficiently.

ii) **Recoverability**

The term "recoverability" may refer to the ability of a system to recover from failure; within concurrency control of database systems this term has received a specific meaning.

Concurrency control typically also ensures the Recoverability property of schedules for maintaining correctness in cases of aborted transactions. Recoverability means that no committed transaction in a schedule has read data written by an aborted transaction. Such data disappear from the database (upon the abort) and are parts of an incorrect database state. Reading such data violates the consistency rule of ACID. Recoverability is one rule that cannot be compromised, since any relaxation results in quick database integrity violation upon aborts. A commonly utilized special case of recoverability is Strictness, which allows efficient database recovery from failure.

4.4 FAILURE RECOVERY OF DATABASES

4.4.1 What is Database Failure?

Database failure is actually deviation from the normal execution of the database. The failure in the database can happen for any number of reasons. First and foremost is user or human error for data damage, loss or corruption. Included in this type of failure is an application modifying or destroying the data on its own or through a user choice. Recovery and restore to the point in time before the corruption occurred.

This returns the data to a clean position at the cost of any other changes that were being made to the data since the point the corruption took place. Any lost work will need to be re-entered or processes repeated if necessary.

Second reason may be due to media failure leading to data loss or damage. Media failure can happen when the media the data files or transaction logs are stored on fail. Most databases will be stored on computer hard drives or across groups of hard drives on designated servers. Hard drives are mechanical devices, just like automobiles and are made up of parts and pieces that work together. Mechanical devices are known for failure and will need to be replaced once or if, the data has been retrieved from them.

The third reason for database failure is a disastrous or catastrophic event. This can be in the form of fire, flood or any naturally occurring storm. It can also happen through electrical outage, a virus or the deliberate hacking of your data. Any of these can corrupt or cause the loss of your data. The true disaster will be the lack of data backup and or the lack of a recovery plan. Without data backup recovery is impossible. And without a recovery plan there is no guarantee that your data backup will make it through the recovery process.

4.4.2 Recovery Measures and Database Security

Backup plays a vital role in the maintenance of databases. Although most database systems do have backup and recovery procedures and schedulers into their interfaces and infrastructure. The back is not just the data files, it must also backup the transaction logs of the database as well. Without the transaction logs the data files are useless in a recovery event.

Database security

It concerns the use of a broad range of information security controls to protect databases against compromises of their confidentiality, integrity and availability. It involves various types or categories of controls, such as technical, procedural/ administrative and physical. Database security is a specialist topic within the broader realms of computer security, information security and risk management.

Security risks to database systems include the following:

- Unauthorized or unintended access or misuse by authorized database users, database administrators or network/systems managers or by unauthorized users or hackers
- Malware infections causing incidents such as unauthorized access, leakage or disclosure of personal or proprietary data, deletion of or damage to the data or programs, interruption or denial of authorized access to the database, attacks on other systems and the unanticipated failure of database services;
- Overloads, performance constraints and capacity issues resulting in the inability of authorized users to use databases as intended;

- Physical damage to database servers caused by computer room fires or floods, overheating, lightning, accidental liquid spills, static discharge, electronic breakdowns/equipment failures and obsolescence;
- Design flaws and programming bugs in databases and the associated programs and systems, creating various security vulnerabilities (e.g. unauthorized privilege escalation), data loss/corruption, performance degradation etc.;
- Data corruption and/or loss caused by the entry of invalid data or commands, mistakes in database or system administration processes, sabotage/criminal damage etc.

Security Measures

The following are most commonly used measures to provide security to the database:

- **Access control:** This includes restricting the access from the unauthorized users using username and password protection.
- **Auditing:** Setting up security standards for the organisation's database and regular checks may help prevent database failure.
- **Authentication:** Some authentication measures to be setup.
- **Encryption:** Data security implementation using different levels of encryption techniques.
- **Integrity controls:** Some measures for maintaining and regular checking for the integrity of the data in the database.
- **Backups:** Regular backup of data and other log files.
- **Application security:** usage of antivirus and other softwares to help prevent damage to the security.

4.5 FAULT TOLERANCE

Definition

The ability of the whole system to respond gracefully, in terms of execution, to an unexpected hardware or software failure Many fault-tolerant database systems mirror all operations – that is, every operation is performed on two or more duplicate systems, so if one fails the other can take over.

Implementation

Fault-tolerant system is the one that in the event that a component fails, a backup component or procedure can immediately replace its working and take its place with no loss of functionality. Fault tolerance can be provided at the software as well as hardware level since both are equally vulnerable to failures.

In the software implementation, operating system plays a major role and it provides different mechanisms to avoid failures. Hardware implementation includes replicating/mirroring the hardware components so that the mirrored component can take over after the hardware failure is encountered.

Database replication

Database replication is nowadays very commonly used technique by many database management systems, usually with a master/slave configuration between the original and the copies of the data in the database. The master replica logs the updates, which then ripple through to the slaves. The slave outputs a message stating that it

has received the update successfully, thus allowing the sending (and potentially re-sending until successfully applied) of subsequent updates.

4.6 TRANSACTION THEORY

A transaction is a logical unit of work that must complete or fail in its entirety. The transaction properties i.e. ACID are already mentioned in section no. 4.0. The transaction can be in any one of the states as follow:

- **Active State:** It is further divided into two phases.
- **Initial Phase:** this is the phase when the execution is just started.
- **Partially Committed Phase:** a database transaction enters this phase when its final statement has been executed but the updation/changes are not committed. At this phase, the database transaction has finished its execution, but it is still possible for the transaction to be aborted because the output from the execution may remain residing temporarily in main memory – an event like hardware failure may erase the output.
- **Failed State:** this is the state where the transaction cannot execute due to some error or failure.
- **Aborted State:** this state arises when the transaction has failed. An aborted transaction must have no effect on the database and thus any changes it made to the database have to be undone or in technical terms, rolled back. The database will return to its consistent state i.e. the state from where the transaction has begun. The DBMS's recovery scheme is responsible to manage transaction aborts.
- **Committed State:** A database transaction enters the committed state when enough information has been written to disk after completing its execution with success.

The following diagram depicts the different states of a transaction:

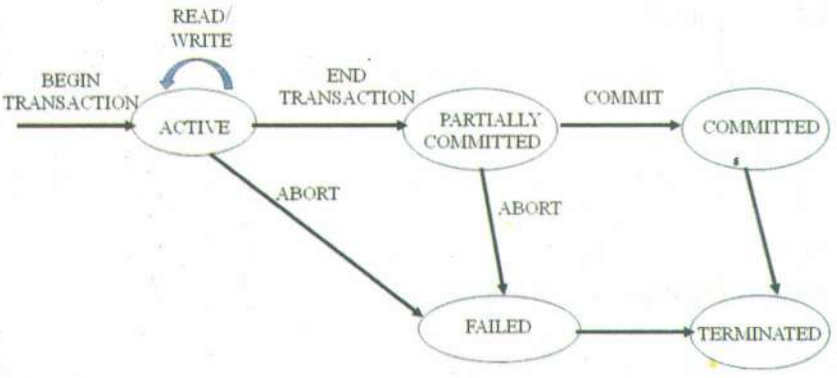


Fig. 2: Transaction states

All database access operations done in between **the beginning and the end of the transaction** constitute a logical unit of work and thus is termed as transaction. There are different operations that take place during the course of execution of a transaction as mentioned below:

READ: This is the operation in which the database items in a transaction are not updated but only retrieved for the pupose of reading data. Thus the operations is written as READ (X) i.e. reading item X.

WRITE: This operation writes the value of program variable **X** into the database item **X** and is denoted as WRITE(X).

Consider the following transactions:

Table1: Concurrent Transactions

T1 TRANSACTION	T2 TRANSACTION
READ (X);	
X:= X-N;	
	READ(X);
	X:=X+M;
WRITE(X);	
READ(Y);	
Y:=Y+N;	
	WRITE(X);
WRITE(Y);	

The table mentioned above illustrates two transactions T1 and T2. The T1 transaction first reads the data item X. It then updates X but doesn't make X permanent to the database. It then goes to wait state. The other transaction named T2 begins execution and reads the original value of X and does some updation and enters wait state. T1 resumes its execution and writes data item X on the database, reads another data item Y, does some updation enters wait state. T2 resumes execution and writes X which the causes ambiguity. Thus, concurrency control measures are used to prevent this situation.

Check Your Progress 1

Note: a) Space is given below for writing your answers.

b) Compare your answers with the one given at the end of this Unit.

1) Define Transaction.

.....

.....

.....

.....

2) What is Database concurrency?

.....

.....

.....

.....

3) Name different concurrency control mechanisms.

.....

.....

.....

.....

4) What is ACID test?

.....

.....

.....

.....

5) What are the different states of a transaction?

.....

.....

.....

.....

6) What are different causes of database failure?

.....

.....

.....

.....

7) Write different ways through which security to the database can be hampered.

.....

.....

.....

.....

4.7 LET US SUM UP

This unit started with database transaction, its definition, database concurrency – a problem and its solution. There are some properties called ACID properties that need to be adhered to by the transactions. Various database concurrency control measures were mentioned for the database to function properly. Different operations and states of the transaction was also mentioned and the various security measures to be taken to prevent database from failure.

4.8 CHECK YOUR PROGRESS: THE KEY

Check Your Progress 1

1) Database transaction refer to a unit a unit of work that must occur or fail in its entirety i.e. it should make some change in the database or it must rollback all together.

2) Database concurrency is a technique that provides control to each transaction and ensures that transactions occur following an order. The main job of these controls is to protect transactions issued by different users/applications from the effects of each other. All the transactions follow four simple characteristics (ACID) of database transactions: atomicity (A), consistency(C), isolation (I) and durability (D).

3) There are different methods of concurrency control mechanisms exist, some of the following are most commonly used ones:

   i) **Locking –** Restricting the access to data by locks assigned to the data. This can be done by the other transaction to a data item to block the availability of the data item i.e. database object etc.

   ii) **Serializability –** This involves checking for cycles in the schedule's graph and breaking them by aborts.

   iii) **Timestamp ordering** – Assigning timestamps or time slices to transactions and controlling or checking access to data by timestamp order.

   iv) **Commitment ordering –** Controlling or checking transactions' chronological order of commit events to be compatible with their respective precedence order.

4) ACID Test is:

   i) **Atomicity –** It is based on all or none concept i.e. either the effects of all or none of its operations when a transaction is completed (committed or aborted respectively). Thus, the transaction results in either done or never started.

   ii) **Consistency –** Every transaction must leave the database in a consistent (correct) state, i.e. maintain the predetermined integrity rules of the database. A transaction must transform a database from one consistent state to another consistent state Thus, since a database can be normally changed only by transactions; all the database's states are consistent. An aborted transaction does not change the state.

   iii) **Isolation –** All the transactions are independent. Transactions cannot interfere with each other. Thus, each transaction is unaware of the concurrently running transactions.

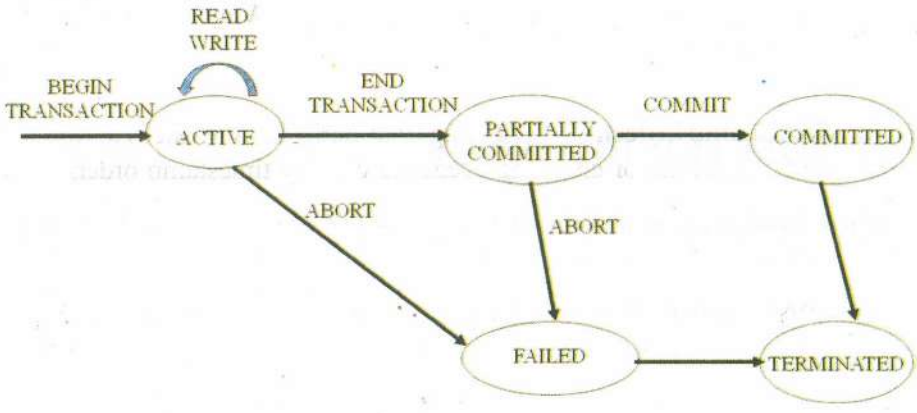
   iv) **Durability –** Effects of successful transactions must persist through crashes.i.e. after the successful completion of the transaction, the changes to the database must persist even in the case with database failure.

5) The transaction can be in any one of the states as follow:

   - Active State: It is further divided into two phases.
   - Initial Phase: this is the phase when the execution is just started.
   - Partially Committed Phase: a database transaction enters this phase when its final statement has been executed but the updation /changes are not committed. At this phase, the database transaction has finished its execution, but it is still possible for the transaction to be aborted because the output from the execution may remain residing temporarily in main memory – an event like hardware failure may erase the output.
   - Failed State: this is the state where the transaction cannot execute due to some error or failure.
   - Aborted State: this state arises when the transaction has failed. An aborted transaction must have no effect on the database and thus any changes it made to the database have to be undone or in technical terms, rolled back. The database will return to its consistent state i.e. the state from where the transaction has begun. The DBMS's recovery scheme is responsible to manage transaction aborts.

- Committed State: A database transaction enters the committed state when enough information has been written to disk after completing its execution with success.

The following diagram depicts the different states of a transaction:



Transaction states

6) First and foremost is user or human error for data damage, loss or corruption. Included in this type of failure is an application modifying or destroying the data on its own or through a user choice. Recovery and restore to the point in time before the corruption occurred.

Second reason may be due to media failure leading to data loss or damage. Media failure can happen when the media the data files or transaction logs are stored on fail. Most databases will be stored on computer hard drives or across groups of hard drives on designated servers. Hard drives are mechanical devices, just like automobiles and are made up of parts and pieces that work together. Mechanical devices are known for failure and will need to be replaced once or if, the data has been retrieved from them.

The third reason for database failure is a disastrous or catastrophic event. This can be in the form of fire, flood or any naturally occurring storm. It can also happen through electrical outage, a virus or the deliberate hacking of your data. Any of these can corrupt or cause the loss of your data. The true disaster will be the lack of data backup and or the lack of a recovery plan. Without data backup recovery is impossible. And without a recovery plan there is no guarantee that your data backup will make it through the recovery process.

7) Security risks to database systems include the following:

- Unauthorized or unintended access or misuse by authorized database users, database administrators or network/systems managers or by unauthorized users or hackers
- Malware infections causing incidents such as unauthorized access, leakage or disclosure of personal or proprietary data, deletion of or damage to the data or programs, interruption or denial of authorized access to the database, attacks on other systems and the unanticipated failure of database services;
- Overloads, performance constraints and capacity issues resulting in the inability of authorized users to use databases as intended;
- Physical damage to database servers caused by computer room fires or floods, overheating, lightning, accidental liquid spills, static discharge, electronic breakdowns/equipment failures and obsolescence;

- Design flaws and programming bugs in databases and the associated programs and systems, creating various security vulnerabilities (e.g. unauthorized privilege escalation), data loss/corruption, performance degradation etc.;
- Data corruption and/or loss caused by the entry of invalid data or commands, mistakes in database or system administration processes, sabotage/criminal damage etc.

4.9 SUGGESTED READINGS

- Database Systems: Design, Implementation and Management By Peter Rob, Carlos Coronel, Steven Morris.
- Database Management Systems by Raghu Ramakrishnan, Johannes Gehrke.

Student Satisfaction Survey

Student Satisfaction Survey of IGNOU Students

ignou THE PEOPLE'S UNIVERSITY

Enrollment No.	
Mobile No.	
Name	
Programme of Study	
Year of Enrolment	
Age Group	☐ Below 30 ☐ 31-40 ☐ 41-50 ☐ 51 and above
Gender	☐ Male ☐ Female
Regional Centre	
States	
Study Center Code	

Please indicate how much you are satisfied or dissatisfied with the following statements

Sl. No.	Questions	Very Satisfied	Satisfied	Average	Dissati-sfied	Very Dissati-sfied
1.	Concepts are clearly explained in the printed learning material	☐	☐	☐	☐	☐
2.	The learning materials were received in time	☐	☐	☐	☐	☐
3.	Supplementary study materials (like video/audio) available	☐	☐	☐	☐	☐
4.	Academic counselors explain the concepts clearly	☐	☐	☐	☐	☐
5.	The counseling sessions were interactive	☐	☐	☐	☐	☐
6.	Changes in the counseling schedule were communicated to you on time	☐	☐	☐	☐	☐
7.	Examination procedures were clearly given to you	☐	☐	☐	☐	☐
8.	Personnel in the study centers are helpful	☐	☐	☐	☐	☐
9.	Academic counseling sessions are well organized	☐	☐	☐	☐	☐
10.	Studying the programme/course provide the knowledge of the subject	☐	☐	☐	☐	☐
11.	Assignments are returned in time	☐	☐	☐	☐	☐
12.	Feedbacks on the assignments helped in clarifying the concepts	☐	☐	☐	☐	☐
13.	Project proposals are clearly marked and discussed	☐	☐	☐	☐	☐
14.	Results and grade card of the examination were provided on time	☐	☐	☐	☐	☐
15.	Overall, I am satisfied with the programme	☐	☐	☐	☐	☐
16.	Guidance from the programme coordinator and teachers from the school	☐	☐	☐	☐	☐

After filling this questionnaire send it to:
Programme Coordinator, School of Vocational Education and Training,
Room no. 19, Block no. 1, IGNOU, Maidangarhi, New Delhi- 110068

MPDD-IGNOU/P.O. 1T/September 2011

ISBN-978-81-266-5616-5