DATABASE MANAGEMENT SYSTEMS

Faculty name: Ms. Karuna Middha

Student Name: DEEPANSHU GAUR
Roll No.: 60414802716
Semester: IV
Group: 4-C-12

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY
MAIT
उद्यमेन हि सिध्यन्ति
कार्याणि न मनोरथैः

Maharaja Agrasen Institute of Technology, PSP area,
Sector – 22, Rohini, New Delhi – 110085
(Affiliated to Guru Gobind Singh Indraprastha University,
New Delhi)

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

MAIT

उद्यमेन हि सिध्यन्ति
कार्याणि न मनोरथैः

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

COMPUTER SCIENCE & ENGINEERING DEPARTMENT

VISION

To Produce "Critical thinkers of Innovative Technology"

MISSION

To provide an excellent learning environment across the computer science discipline to inculcate professional behavior, strong ethical values, innovative research capabilitiesand leadership abilities which enable them to become successful entrepreneurs in this globalized world.

1. To nurture an **excellent learning environment** that helps students to enhance their problem solving skills and to prepare students to be lifelong learners by offering a solid theoretical foundation with applied computing experiences and educating them about their **professional, and ethical responsibilities**.
2. To establish **Industry-Institute Interaction**, making students ready for the industrial environment and be successful in their professional lives.
3. To promote **research activities** in the emerging areas of technology convergence.
4. To build engineers who can look into technical aspects of an engineering solution thereby setting a ground for producing successful **entrepreneur.**

Computer Science and Engineering Department
Outcome Based Education
Course Outcome

Subject:	DBMS LAB	Max Marks External	60 Marks
Subject Code:	ETCS-208	Max Marks Internal	40 Marks
Total Credit:	1	**Evaluation Scheme**	
Contact Hours:		Evaluation 75 Marks	End-term Exam
		Evaluation 25 Marks	Sessional Exam

Course Objective:

The major objective of this lab is to provide a strong formal foundation in database concepts, technology and practice to the participants to groom them into well-informed database application developers. This course prepares the students for developing intelligent and normalized database management systems. Upon completion of this course, the student should be able to understand the research trends in database management system.

Course Outcome:

At the end of the course, a student will be able to:

C256.1: Uunderstand, the importance, features and basic terminologies used in database.

C256.2: Design and implement a database schema for a given problem-domain.

C256.3: Write SQL DDL/DML queries for a given database.

C256.4: Implement and define integrity constraints on the relational database.

C256.5: Able to use database security & authorization checks on the given database.

C256.6: Identify the potential research areas in Database Management System.

Name of Student
Roll No
Group

INDEX

R1 | R2 | R3 | R4 | R5 | Total marks | Sign/Grade

Index

Exp. no	Experiment Name	Date of performance	Date of checking	Marks	Signature
1.	Introduction to DBMS & SQL	9/01/18	16/1/18		Varuna 16/1/18
2.	To create a table and insert data into them	16/1/18	23/1/18		Varuna 23/1/18
3.	To execute following queries on retrieving records from a table client master, salesman	23/1/18	30/1/18		Varuna 30/1/18
4.	SQL statement - ALTER, UPDATE, DELETE	30/1/18	6/2/18		Varuna 6/2/18
5.	Write Queries to implement integrity constraint	6/2/18	13/2/18		Varuna 13/2/18
6.	Queries - MAX() MIN(), AVG() COUNT().	13/2/18	20/2/18		Varuna 20/2/18
7.	Queries - month and date command	20/2/18	6/3/18		Varuna 6/3/18
8.	Queries - Having and Group By	6/3/18	13/3/18		Varuna 13/3/18
9.	Queries to implement joins	13/3/18	20/3/18		Varuna 20/3/18
10.	Queries to create view	20/3/18	27/3/18		Varuna 27/3/18

11.	Intro on PL/SQL, triggers, cursors, stored procedures.	27/3/18	3/4/18		Karuna 3/4/18
12.	comparison on different databases.	3/4/18	17/4/18		Karuna 17/4/18
13.	Brief note on research in field of database.	10/4/18	17/4/18		Karuna 17/4/18

Date

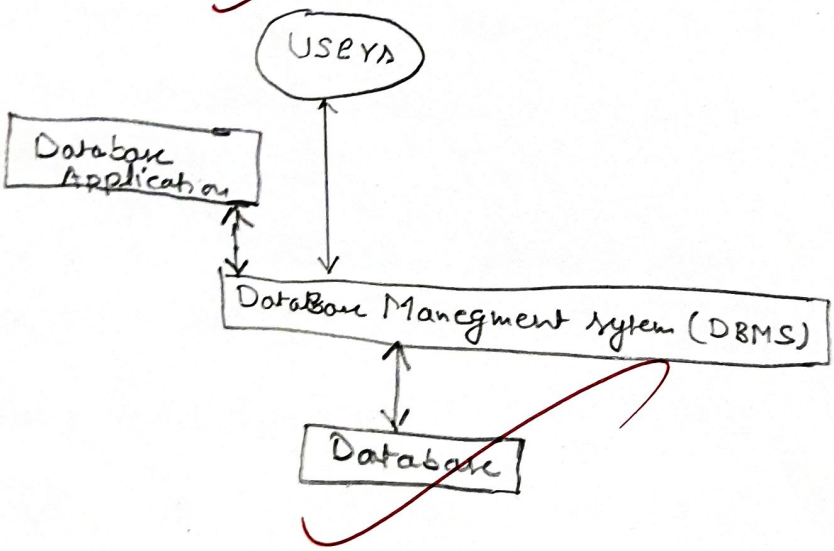
Experiment - 01

Aim: Introduction to DBMS and SQL

DBMS (Database Management System):

A DBMS is a software that allows creation, definition and manipulation of database.
It also provides protection, security and maintains data consistency in a database.

For Eg: MySQL, oracle, Sybase, Microsoft Access, IBM DB2



Users: Users may be of various types such as DBadministration system developer, end users etc.

Database Application: It can be personell, departmental enterprise and internal.

DBMS - software that allows its users to create and manages data in a database.

Database - collection of logical data.

* Advantages of DBMS :

- Controlling redundancy : By being centralised database, data redundancy can be avoided in case of DBMS.
- Data Integrity : DBMS provides definition and enforcement of integrity constraints to ensure that data stored is always accurate.
- Data sharing : Data sharing is the most usable feature of a DBMS.
- Enforcement of standard since DBMS is a central system, standard can be enforced at company level, national level, departmental level.
- Provides Back up & recovery : prevents data loses due to hardware or software failures.
- Maintenance and developing costs are lower.
- Data model can be developed & altered.
- Concurrency control : provides concurrent access of data to multiple users.

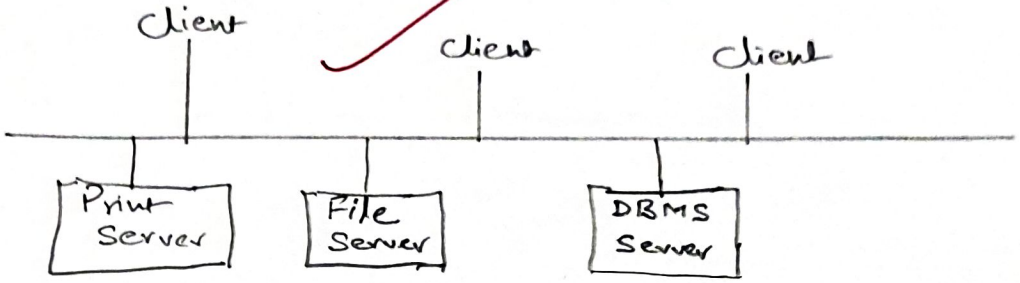
* Disadvantages of DBMS :

- Complexity : more efficient the DBMS is, more its complexity
- Size : complexity and functionality makes it an extremely big software.
- Performance : Due to its multi-tasking behaviour some app. may not run fast.
- Cost of # convertion - convertion to the updation version of DBMS is high.

* Database Architecture :

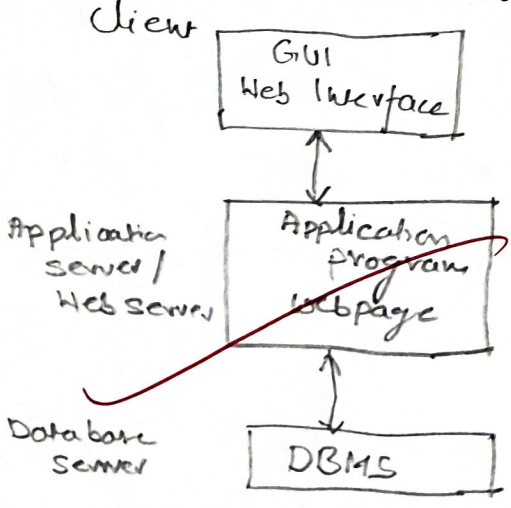
Database architecture is logically divided into 2 types:

① Logical two-tier client/server architecture



Two-tier client/server architecture is used for user interface program and application programs that runs on client side. A client program may connect to several DBMSs. In this architecture some variation of client is also possible such as functionality of data dictionary, optimization is transferred to client, such client are called data servers.

② Three-tier client/server architecture :



This architecture is commonly used for web applications. Intermediate layer called application / web server stores the web-connectivity software and the business logic (constraints) part of application used to access the right amount of data.

SQL (Structured Query Language):-

This is a computer language for storing, manipulating and retrieving data stored in a relational database. It is the standard language for relational DBMS. All the RDMS like MYSQL, MS Access, oracle and SQL server or their standard database language some dialects are:

(i) MS SQL server using T-SQL;
(ii) Oracle using PL/SQL;
(iii) MS-Access version of SQL is called JET-SQL etc.

* Advantages of SQL:-

- Allow users to access data in a RDBMS.
- Allow users to describe the data.
- Allow users to define and manipulate the data.
- Allows to create databases.
- Allows to drop databases and Tables
- Allows users to create, view stored procedures & functions.
- Allow users to set permission on tables.

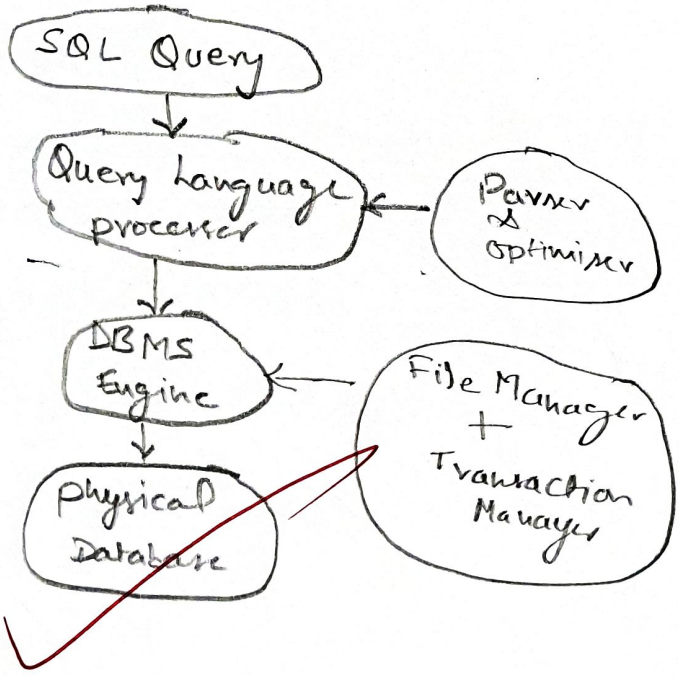
* SQL process :

→ When you are executing an SQL command in RDBMS, the system determines the best way to carry out your request and SQL engine figures exact how to interpret the task.

Components included in this process are :

* Query dispatcher.
* optimization Engines.
* Classic Query Engine.

SQL Architecture :. →



SQL commands:

SQL commands can be classified in following groups. based on their nature:

* DDL (Data Definition Language)

1. Create - create a new table, a view of a table, or other object in a database.
2. Alter - modifies an existing database object i.e table.
3. Drop - deletes an entire table, a view of a table or other objects in Database.

* DML - Data Manipulation Language

1. Select - Retrieves certain records from one / more tables
2. Insert - creates a record.
3. Update - Modifies record.

* DCL - Data control Language.

1. Select - extracts data from the database.

SQL Data types:

1) CHAR(SIZE) - holds a fixed length string.
2) VARCHAR(size) - holds a variable length string.
3) TEXT - holds a string with maximum length of 65,535 char.
4) BLOB - Binary large objects - holds upto 65,535

5) ENUM – LET enter a list of possible values.

6) Int - 2147483648 to 2147483647

7) Float (size, d) - a small no. with floating dec. pt

8) Double (size, d) - large no. with floating decimal point

9) Date () - YYYY-MM-DD.

10) Time () - A time format HH:MM:ss

11) YEAR() - a year in two-digit of 4 digit format.

```
MariaDB [file]> create table client_master( clientno varchar(6), name varchar(20), add1 varchar(30), add2 varchar(30), city varchar(15)
, pincode varchar(8), state varchar(15), baldue int );
Query OK, 0 rows affected (0.84 sec)

MariaDB [file]> create table product_master( production varchar(6), description varchar(15), profitpercent float, unitmeaure varchar(10
    -> ),  qtyonhand int, reorderlvl int, sellprice float, costprice float);
Query OK, 0 rows affected (0.34 sec)

MariaDB [file]> create table salesman_master( salesmanno varchar(6), name varchar(20), add1 varchar(30), add2 varchar(30), city varchar
(20), pincode int, state varchar(20),salamt float, tgttoget float, ytdsales float, remarks varchar(60) );
Query OK, 0 rows affected (0.81 sec)
```

```
MariaDB [file]> insert into client_master(clientno,name,city,pincode,state,baldue) values('c00004','Aswini Joshi','Banglore','560001','
Karnataka',0);
Query OK, 1 row affected (0.07 sec)

MariaDB [file]> insert into client_master(clientno,name,city,pincode,state,baldue) values('c00005','Hansel Calaco','Mumbai','400060','M
aharashtra',2000);
Query OK, 1 row affected (0.06 sec)

MariaDB [file]> insert into client_master(clientno,name,city,pincode,state,baldue) values('c00006','Deepak Sharma','Manglore','560050',
'Karnataka',0);
Query OK, 1 row affected (0.32 sec)

MariaDB [file]> select * from client_master;
+----------+---------------+------+------+-----------+---------+-------------+--------+
| clientno | name          | add1 | add2 | city      | pincode | state       | baldue |
+----------+---------------+------+------+-----------+---------+-------------+--------+
| c00001   | Ivan          | NULL | NULL | Mumbai    | 400054  | Maharashtra |  15000 |
| c00002   | Mamta Muzumdar | NULL | NULL | Madras    | 780001  | Tamil Nadu  |      0 |
| c00003   | Chhaya Bankar | NULL | NULL | Mumbai    | 400057  | Maharashtra |   5000 |
| c00004   | Aswini Joshi  | NULL | NULL | Banglore  | 560001  | Karnataka   |      0 |
| c00005   | Hansel Calaco | NULL | NULL | Mumbai    | 400060  | Maharashtra |   2000 |
| c00006   | Deepak Sharma | NULL | NULL | Manglore  | 560050  | Karnataka   |      0 |
+----------+---------------+------+------+-----------+---------+-------------+--------+
6 rows in set (0.00 sec)
```

```
MariaDB [file]> insert into product_master values('P00001','T-Shirts',5,'Piece',200,50,350,250);
Query OK, 1 row affected (0.32 sec)

MariaDB [file]> insert into product_master values('P0345','Shirts',6,'Piece',150,20,500,350);
Query OK, 1 row affected (0.08 sec)

MariaDB [file]> insert into product_master values('P06734','Cotton Jeans',5,'Piece',100,20,600,450);
Query OK, 1 row affected (0.34 sec)

MariaDB [file]> insert into product_master values('P07865','Cotton Jeans',5,'Piece',100,20,750,500);
Query OK, 1 row affected (0.06 sec)
```

Experiment No. 2

Aim: To create a table and insert data into them using commands.

1. Table : Client_Master :

Client No	Name	City	Pincode	State	Baldue
c00001	Ivan	Mumbai	400054	Maharashtra	15000
c00002	Mamta Mazumdar	Madras	780001	Tamil Nadu	0
c00003	Chhaya Bankar	Mumbai	4000053	Maharashtra	5000
c00004	Ashwini Joshi	Bangalore	560001	Karnataka	0
c00005	Hansal Calaco	Mumbai	400060	Maharashtra	2000
c00006	Deepak Sharma	Bangalore	560050	Karnataka	0

2. Table Product_master :

Production	Description	Profit Percent	Unit measure	qty onhand	Reorder dlvr	Sell price	Cost price
P00001	Tshirts	5	Piece	200	80	350	250
P0345	Shirts	6	Piece	150	50	500	380
P06734	cotton Jeans	5	~~Piece~~	100	20	600	450
P07865	~~Trousers~~ cotton Jeans	5	Piece	100	20	750	500
P07868	~~Pullovers~~ Trousers	2	Piece	150	50	850	505
P07585	~~Deni~~ Pullovers	2.5	Piece	80	30	700	450
P07965	Denim Shirts	4	Piece	100	40	300	250
P07975	Lycra Tops.	5	Piece	70	30	300	175
P08865	Skirts.	5	Piece	75	30	450	325

```
MariaDB [file]> select * from product_master;
+------------+--------------+---------------+------------+-----------+-------------+-----------+-----------+
| production | description  | profitpercent | unitmeaure | qtyonhand | reorderlvl  | sellprice | costprice |
+------------+--------------+---------------+------------+-----------+-------------+-----------+-----------+
| P00001     | T-Shirts     |             5 | Piece      |       200 |          50 |       350 |       250 |
| P0345      | Shirts       |             6 | Piece      |       150 |          50 |       500 |       350 |
| P06734     | Cotton Jeans |             5 | Piece      |       100 |          20 |       600 |       450 |
| P07865     | Cotton Jeans |             5 | Piece      |       100 |          20 |       750 |       500 |
| P07868     | Trousers     |             2 | Piece      |       150 |          50 |       850 |       550 |
| P07885     | Pull overs   |           2.5 | Piece      |        80 |          30 |       700 |       450 |
| P07965     | Denim Shirts |             4 | Piece      |       100 |          40 |       350 |       250 |
| P07975     | Lycra Tops   |             5 | Piece      |        70 |          30 |       300 |       175 |
| P08865     | Skirts       |             5 | Piece      |        75 |          30 |       450 |       300 |
+------------+--------------+---------------+------------+-----------+-------------+-----------+-----------+
9 rows in set (0.00 sec)
```

```
MariaDB [file]> insert into salesman_master values('S00001','Aman','A/14','Werli','Mumbai',400002,'Maharashtra',3000,100,50,'Good');
Query OK, 1 row affected (0.36 sec)

MariaDB [file]> insert into salesman_master values('S00002','Omkar','65','Nariman','Mumbai',400001,'Maharashtra',3000,200,100,'Good');
Query OK, 1 row affected (0.06 sec)

MariaDB [file]> insert into salesman_master values('S00003','Raj','P-7','Bandra','Mumbai',400032,'Maharashtra',3000,200,100,'Good');
Query OK, 1 row affected (0.06 sec)

MariaDB [file]> insert into salesman_master values('S00004','Ashish','A/5','Juhu','Mumbai',400044,'Maharashtra',3500,200,150,'Good');
Query OK, 1 row affected (0.33 sec)

MariaDB [file]> select * from salesman_master;
+------------+--------+-------+---------+--------+---------+-------------+--------+-----------+----------+---------+
| salesmanno | name   | add1  | add2    | city   | pincode | state       | salamt | tgttoget  | ytdsales | remarks |
+------------+--------+-------+---------+--------+---------+-------------+--------+-----------+----------+---------+
| S00001     | Aman   | A/14  | Werli   | Mumbai |  400002 | Maharashtra |   3000 |       100 |       50 | Good    |
| S00002     | Omkar  | 65    | Nariman | Mumbai |  400001 | Maharashtra |   3000 |       200 |      100 | Good    |
| S00003     | Raj    | P-7   | Bandra  | Mumbai |  400032 | Maharashtra |   3000 |       200 |      100 | Good    |
| S00004     | Ashish | A/5   | Juhu    | Mumbai |  400044 | Maharashtra |   3500 |       200 |      150 | Good    |
+------------+--------+-------+---------+--------+---------+-------------+--------+-----------+----------+---------+
4 rows in set (0.00 sec)
```

③ Table : Salesman_master :

Salesman No.	Name	Add1	Add2	City	Pincode
S00001	Aman	A/14	Worli	Mumbai	400002
S00002	omkar	65	Nariman	Mumbai	400001
S00003	Raj	P-7	Bandra	Mumbai	400032
S00004	Ashish	A/5	Juhu	Mumbai	400044

Salesman No.	State	SalAmt	Tgttoget	Yrdsales	Remarks
S00001	Maharashtra	3000	100	50	Good
S00002	Maharashtra	3000	200	100	Good
S00003	Maharashtra	3000	200	100	Good
S00004	Maharashtra	3500	200	150	Good

(a)

```
MariaDB [file]> select name from client_master;
+---------------+
| name          |
+---------------+
| Ivan          |
| Mamta Muzumdar |
| Chhaya Bankar |
| Aswini Joshi  |
| Hansel Calaco |
| Deepak Sharma |
+---------------+
6 rows in set (0.00 sec)
```

(b)

```
MariaDB [file]> select * from client_master;
+----------+---------------+------+------+----------+---------+-------------+--------+
| clientno | name          | add1 | add2 | city     | pincode | state       | baldue |
+----------+---------------+------+------+----------+---------+-------------+--------+
| c00001   | Ivan          | NULL | NULL | Mumbai   | 400054  | Maharashtra | 15000  |
| c00002   | Mamta Muzumdar | NULL | NULL | Madras   | 780001  | Tamil Nadu  |      0 |
| c00003   | Chhaya Bankar | NULL | NULL | Mumbai   | 400057  | Maharashtra |   5000 |
| c00004   | Aswini Joshi  | NULL | NULL | Banglore | 560001  | Karnataka   |      0 |
| c00005   | Hansel Calaco | NULL | NULL | Mumbai   | 400060  | Maharashtra |   2000 |
| c00006   | Deepak Sharma | NULL | NULL | Manglore | 560050  | Karnataka   |      0 |
+----------+---------------+------+------+----------+---------+-------------+--------+
6 rows in set (0.00 sec)
```

```
MariaDB [file]> select name,city,state from client_master;
+---------------+----------+-------------+
| name          | city     | state       |
+---------------+----------+-------------+
| Ivan          | Mumbai   | Maharashtra |
| Mamta Muzumdar | Madras   | Tamil Nadu  |
| Chhaya Bankar | Mumbai   | Maharashtra |
| Aswini Joshi  | Banglore | Karnataka   |
| Hansel Calaco | Mumbai   | Maharashtra |
| Deepak Sharma | Manglore | Karnataka   |
+---------------+----------+-------------+
6 rows in set (0.00 sec)

MariaDB [file]> select description from product_master;
+--------------+
| description  |
+--------------+
| T-Shirts     |
| Shirts       |
| Cotton Jeans |
| Cotton Jeans |
| Trousers     |
| Pull overs   |
| Denim Shirts |
| Lycra Tops   |
| Skirts       |
+--------------+
9 rows in set (0.01 sec)
```

Experiment 03

Aim: To execute following queries on retrieving records from a table: client_master, salesman_master, product_master.

(a) Find out the names of all the clients:

→ select name from client_master;

(b) Retrieve the entire contents of client_master table:

→ select * from client_master;

(c) Retrieve the list of names, city and state of all clients:

→ SELECT name, city, state from client_master;

(d) List the various products available from product master table:

→ select description from product_master;

```
MariaDB [file]> select name from salesman_master where salamt = 3000;
+-------+
| name  |
+-------+
| Aman  |
| Omkar |
| Raj   |
+-------+
3 rows in set (0.00 sec)

MariaDB [file]> select salesmanno from sales_master where city ='Mumbai';
ERROR 1146 (42S02): Table 'file.sales_master' doesn't exist
MariaDB [file]> select salesmanno from salesman_master where city ='Mumbai';
+------------+
| salesmanno |
+------------+
| S00001     |
| S00002     |
| S00003     |
| S00004     |
+------------+
4 rows in set (0.00 sec)

MariaDB [file]> select production from product_master where costprice=150;
Empty set (0.00 sec)
```

```
MariaDB [file]> select production from product_master where description = 'T-Shirts';
+------------+
| production |
+------------+
| P00001     |
+------------+
1 row in set (0.00 sec)

MariaDB [file]> select name from client_master where baldue=0;
+---------------+
| name          |
+---------------+
| Mamta Muzumdar |
| Aswini Joshi  |
| Deepak Sharma |
+---------------+
3 rows in set (0.00 sec)

MariaDB [file]> select name from client_master where city ='Mumbai';
+--------------+
| name         |
+--------------+
| Ivan         |
| Chhaya Bankar |
| Hansel Calaco |
+--------------+
3 rows in set (0.00 sec)
```

(e) List all the clients who are located in mumbai:

→ select name from client_master where city = 'Mumbai';

(f) Find the names of salesman who have a salary equal to Rs 3000:

→ select name from salesman_master where salamt = 3000;

(g) List all salesman no. who live in city mumbai:

→ select salesman from salesman_master where city = 'mumbai';

(h) Find product no. whose price is equal to Rs 150:-

→ select product no. from product_master where cost price = 150;

(i) Find product no. of Tshirts in product_master tables

→ select production from product_master where description = 'T-shirts';

(j) Find names of client from client_master table where Balance is '0':

→ select name from client_master where balance = 0;

```
MariaDB [file]>
MariaDB [file]> update client_master set city ='Bangalore' where clientno ='c00005';
Query OK, 0 rows affected (0.06 sec)
Rows matched: 1  Changed: 0  Warnings: 0

MariaDB [file]> update client_master set baldue=1000 where clientno ='c00001';
Query OK, 0 rows affected (0.05 sec)
Rows matched: 1  Changed: 0  Warnings: 0

MariaDB [file]> select * from client_master where (clientno ='c00001') OR (clientno ='c00005');
+----------+---------------+-------+-------+-----------+---------+-------------+---------+
| clientno | name          | add1  | add2  | city      | pincode | state       | baldue  |
+----------+---------------+-------+-------+-----------+---------+-------------+---------+
| c00001   | Ivan          | NULL  | NULL  | Mumbai    |  400054 | Maharashtra |    1000 |
| c00005   | Hansel Calaco | NULL  | NULL  | Bangalore |  400060 | Maharashtra |    2000 |
+----------+---------------+-------+-------+-----------+---------+-------------+---------+
2 rows in set (0.01 sec)
```

```
MariaDB [file]> update product_master set costprice=950 where description ='Trousers';
Query OK, 1 row affected (0.31 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [file]> select * from product_master where description ='Trousers';
+------------+-------------+---------------+-------------+-----------+--------------+-----------+-----------+
| production | description | profitpercent | unitmeasure | qtyonhand | reorderlvl | sellprice | costprice |
+------------+-------------+---------------+-------------+-----------+--------------+-----------+-----------+
| P07868     | Trousers    |             2 | Piece       |       150 |           50 |       850 |       950 |
+------------+-------------+---------------+-------------+-----------+--------------+-----------+-----------+
1 row in set (0.00 sec)

MariaDB [file]> update salesman_master set city ='Pune';
Query OK, 4 rows affected (0.08 sec)
Rows matched: 4  Changed: 4  Warnings: 0

MariaDB [file]> delete from salesman_product where salamt=3500;
ERROR 1146 (42S02): Table 'file.salesman_product' doesn't exist
MariaDB [file]> delete from salesman_master where salamt=3500;
Query OK, 1 row affected (0.06 sec)

MariaDB [file]> select * from salesman_master;
+------------+-------+-------+---------+------+---------+-------------+---------+-----------+---------+---------+
| salesmanno | name  | add1  | add2    | city | pincode | state       | salamt  | tgttoget  | ytdsales | remarks |
+------------+-------+-------+---------+------+---------+-------------+---------+-----------+---------+---------+
| S00001     | Aman  | A/14  | Worli   | Pune |  400002 | Maharashtra |    3000 |       100 |       50 | Good    |
| S00002     | Omkar | 65    | Nariman | Pune |  400001 | Maharashtra |    3000 |       200 |      100 | Good    |
| S00003     | Raj   | P-7   | Chandra | Pune |  400032 | Maharashtra |    3000 |       200 |      100 | Good    |
+------------+-------+-------+---------+------+---------+-------------+---------+-----------+---------+---------+
3 rows in set (0.00 sec)
```

```
MariaDB [file]> delete from product_master where qtyonhand=100;
Query OK, 3 rows affected (0.06 sec)

MariaDB [file]> delete from client_master where state ='Tamil Nadu';
Query OK, 1 row affected (0.07 sec)
```

Experiment 4.

Aim: To update, delete, alter, rename records in the tables:

(a) Change the city of client no. 'c00005' to 'Banglore'.

→ update client_master set city = 'Bangalore' where clientNo = 'c00005';

(b) change the baldue of clientNo 'c00001' to Rs 1000.

→ update client_master set Baldue = 1000 where clientNo = 'c00001';

(c) change the cost price of Trousers to Rs 950.00.

→ update product_master set costprice = 950 where description = 'Trousers';

(d) change the city of salesman to pune:

→ update salesman_master set city = 'Pune';

(e) Delete all salesman from salesman_master whose salaries are equal to Rs 3500.

→ delete from salesman_master where salAmt = 3500;

(f) Delete all product from product_master where the quantity on hand is 100

→ delete from product_master where qty_onHand = 100;

(g) Delete from client_master where columns state holds value 'Tamil Nadu':

→ delete from client_master where state = 'Tamil Nadu'

```
MariaDB [file]> alter table client_master add column(telephone int(10) );
Query OK, 0 rows affected (0.67 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [file]> select * from client_master;
+----------+--------------+------+------+-----------+---------+-------------+---------+-----------+
| clientno | name         | add1 | add2 | city      | pincode | state       | baldue  | telephone |
+----------+--------------+------+------+-----------+---------+-------------+---------+-----------+
| c00001   | Ivan         | NULL | NULL | Mumbai    | 400054  | Maharashtra |    1000 |      NULL |
| c00003   | Chhaya Bankar| NULL | NULL | Mumbai    | 400057  | Maharashtra |    5000 |      NULL |
| c00004   | Aswini Joshi | NULL | NULL | Banglore  | 560001  | Karnataka   |       0 |      NULL |
| c00005   | Hansel Calaco| NULL | NULL | Bangalore | 400060  | Maharashtra |    2000 |      NULL |
| c00006   | Deepak Sharma| NULL | NULL | Manglore  | 560050  | Karnataka   |       0 |      NULL |
+----------+--------------+------+------+-----------+---------+-------------+---------+-----------+
5 rows in set (0.00 sec)

MariaDB [file]> select * from product_master;
+------------+-------------+---------------+-------------+-------------+-------------+-----------+-----------+
| production | description | profitpercent | unitmeaure  | qtyonhand   | reorderlvl  | sellprice | costprice |
+------------+-------------+---------------+-------------+-------------+-------------+-----------+-----------+
| P00001     | T-Shirts    |             5 | Piece       |         200 |          50 |       350 |       250 |
| P0345      | Shirts      |             6 | Piece       |         150 |          50 |       500 |       350 |
| P07868     | Trousers    |             2 | Piece       |         150 |          50 |       850 |       950 |
| P07885     | Pull overs  |           2.5 | Piece       |          80 |          30 |       700 |       450 |
| P07975     | Lycra Tops  |             5 | Piece       |          70 |          30 |       300 |       175 |
| P08865     | Skirts      |             5 | Piece       |          75 |          30 |       450 |       300 |
+------------+-------------+---------------+-------------+-------------+-------------+-----------+-----------+
6 rows in set (0.00 sec)
```

```
MariaDB [file]> drop table client_master;
Query OK, 0 rows affected (0.22 sec)

MariaDB [file]> select * from client_master;
ERROR 1146 (42S02): Table 'file.client_master' doesn't exist
MariaDB [file]> exec sp_rename 'salesman_master','sman_mast';
```

```
MariaDB [file]> rename table salesman_master to sman_mast;
Query OK, 0 rows affected (0.23 sec)

MariaDB [file]> select* from salesman_master;
ERROR 1146 (42S02): Table 'file.salesman_master' doesn't exist
MariaDB [file]> select* from sman_mast;
+------------+--------+------+--------+------+---------+-------------+---------+-----------+-----------+---------+
| salesmanno | name   | add1 | add2   | city | pincode | state       | salamt  | tgttoget  | ytdsales  | remarks |
+------------+--------+------+--------+------+---------+-------------+---------+-----------+-----------+---------+
| S00001     | Aman   | A/14 | Worli  | Pune |  400002 | Maharashtra |    3000 |       100 |        50 | Good    |
| S00002     | Omkar  | 65   | Nariman| Pune |  400001 | Maharashtra |    3000 |       200 |       100 | Good    |
| S00003     | Raj    | P-7  | Bandra | Pune |  400032 | Maharashtra |    3000 |       200 |       100 | Good    |
+------------+--------+------+--------+------+---------+-------------+---------+-----------+-----------+---------+
3 rows in set (0.00 sec)
```

(h) Add a column called 'Telephone' of datatype 'number' and size = 10 to the client_master table:

→ Alter table client_master add column (telephone int(10));

(i) Change the size of Sell price column in product_master to 10,2:

→ Alter table product_master update column (sellprice float(10,2));

(j) Destroy the table client_master along with its data:

→ Drop table client_master;

(k) Change the name of salesman_master table to sman_mast:

→ rename table salesman_master to sman_mast;

Aasawa

```
MariaDB [file]> create table client_master ( clientno varchar(6), name varchar(20) NOT NULL, add1 varchar(30), add2 varchar(30), city v
archar(15), pincode int(8), state varchar(15), baldue dec(10,2), primary key(clientno) ,check(clientno like 'C%') );
Query OK, 0 rows affected (0.35 sec)

MariaDB [file]> create table product_master (productno varchar(6) , description varchar(15) NOT NULL, profitpercent dec(4,2) NOT NULL,
unitmeasure varchar(10) NOT NULL, qtyonhead int(8), reorderlvl int(8) NOT NULL, sellprice dec(8,2) NOT NULL , costprice dec(8,2) NOT NU
LL , primary key(productno), check(productno like 'P%'), check(sellprice <> 0) , check(costprice<>0) );
Query OK, 0 rows affected (0.35 sec)

MariaDB [file]> create table salesman_master (salesmanno varchar(6) ,name varchar(20) NOT NULL, add1 varchar(30) NOT NULL, add2 varchar
(30), city varchar(20) , pincode int(8), state varchar(20) , salamt dec(8,2) NOT NULL, tgttoget dec(6,2) NOT NULL, ytdsales dec(6,2) NO
T NULL , remarks varchar(60) , primary key(salesmanno), check(salesmanno like 'S%'), check(salamt <>0),check(tgttoget <> 0) );
Query OK, 0 rows affected (0.40 sec)

MariaDB [file]> create table sales_order ( orderno varchar(6), clientno varchar(6), orderdate date NOT NULL, delyadd varchar(25), sales
manno varchar(6), delytype char(1) default 'F', billyn char(1), delydate date , orderstatus varchar(10), primary key(orderno), foreign
key(clientno) references client_master(clientno), foreign key(salesmanno) references salesman_master(salesmanno), check(orderno like 'O
%' ), constraint chk_sales_order check((delytype ='F' OR delytype='f' OR delytype ='p' OR delytype ='P') AND (delydate >= orderdate) AN
D ( orderstatus='in progress' OR orderstatus='Fulfilled' OR orderstatus ='Backorder' OR orderstatus ='Cancelled')) );
Query OK, 0 rows affected (0.52 sec)

MariaDB [file]> create table sales_order_details (orderno varchar(6), productno varchar(6), qtyordered int(8) , qtydisp int(8), product
rate(10,2), foreign key(orderno) references sales_order(orderno), foreign key(productno) references product_master(productno) );
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the righ
t syntax to use near '(10,2), foreign key(orderno) references sales_order(orderno), foreign key(produc' at line 1
MariaDB [file]> create table sales_order_details (orderno varchar(6), productno varchar(6), qtyordered int(8) , qtydisp int(8), product
rate dec(10,2), foreign key(orderno) references sales_order(orderno), foreign key(productno) references product_master(productno) );
Query OK, 0 rows affected (0.46 sec)
```

```
MariaDB [file]> insert into client_master(clientno,name,city,pincode,state,baldue) values('C00001','Ivan','Mumbai',400054,'Maharashtra
',15000);
Query OK, 1 row affected (0.08 sec)

MariaDB [file]> insert into client_master(clientno,name,city,pincode,state,baldue) values('C00002','Mamta Muzumdar','Madras',780001,'T
amil Nadu',0);
Query OK, 1 row affected (0.05 sec)

MariaDB [file]> insert into client_master(clientno,name,city,pincode,state,baldue) values('C00003','chhaya bankar','Mumbai',400057,'Ma
harashtra',5000);
Query OK, 1 row affected (0.36 sec)
```

```
MariaDB [file]> insert into product_master values('P00001','T-Shirts',5,'Piece',200,50,350,250);
Query OK, 1 row affected (0.06 sec)

MariaDB [file]> insert into product_master values('P0345','Shirts',6,'Piece',150,50,500,350);
Query OK, 1 row affected (0.07 sec)

MariaDB [file]> insert into product_master values('P06734','Cotton Jeans',5,'Piece',100,20,600,450);
Query OK, 1 row affected (0.06 sec)
```

Experiment - 05.

Aim: Create the table described as below and insert data into it.

(a) Client_Master table:

~~Client_master table:~~

Client No.	Name	city	Pincode	State	Baldue
C00001	Ivan	Mumbai	400054	Maharashtra	15000
C00002	Mamta Majumdar	Madras	780001	TamilNadu	0
C00003	Chhaya Bankar	Mumbai	400057	Maharashtra	5000
C00004	Ashwini Joshi	Bangalur	560001	Karnataka	0
C00005	Hansel colaco	Mumbai	400060	Maharashtra	20000.
C00006	Deepak Sharma	Mangalore	560050	Karnataka	0

(b) Product - Master Table:

Product No.	Description	Profit %	Unit measure	Qty in hand	Reorder	Sell price	Cost price
P00001	Tshirts	5	Piece	200	50	350	250
P0345	Shirts	6	Piece	150	50	600	350
P06734	Cotton Jeans	5	Piece	100	20	600	400
P07865	Jeans	5	Piece	100	20	750	500
P07868	Trousers	2	Piece	150	50	850	550
P07885	Pullovers	2.5	Piece	80	30	700	450
P07965	Denim Shirts	4	Piece	100	40	350	250
P07975	Lycra Tops	5	Piece	70	30	300	175
P08865	Skirts	5	Piece	75	30	450	300

```
MariaDB [file]> insert into sales_order (orderno,clientno,orderdate,salesmanno,delytype,billyn,delydate,orderstatus ) values('019001','
C00001','040612','S00001','F','N','020720','in process');
Query OK, 1 row affected (0.09 sec)

MariaDB [file]> insert into sales_order (orderno,clientno,orderdate,salesmanno,delytype,billyn,delydate,orderstatus ) values('019002','
C00002','040625','S00002','P','N','020627','Cancelled');
Query OK, 1 row affected (1.15 sec)

MariaDB [file]> insert into sales_order (orderno,clientno,orderdate,salesmanno,delytype,billyn,delydate,orderstatus ) values('046865','
C00003','040218','S00003','F','Y','020220','Fulfilled');
Query OK, 1 row affected (0.34 sec)
```

```
MariaDB [file]> insert into sales_order_details values('046866','P07965',1,0,8400);
Query OK, 1 row affected (0.06 sec)

MariaDB [file]> insert into sales_order_details values('046866','P07975',1,0,1850);
Query OK, 1 row affected (0.34 sec)

MariaDB [file]> insert into sales_order_details values('019008','P00001',10,5,525);
Query OK, 1 row affected (0.32 sec)

MariaDB [file]> insert into sales_order_details values('019008','P07975',5,3,1850);
Query OK, 1 row affected (0.32 sec)
```

```
MariaDB [file]> insert into salesman_master values('S00001','Aman','A/14','Werli','Mumbai',400002,'Maharashtra',3000,100,50,'Good');
Query OK, 1 row affected (0.05 sec)

MariaDB [file]> insert into salesman_master values('S00002','Omkar','65','Nariman','Mumbai',400001,'Maharashtra',3000,200,100,'Good');
Query OK, 1 row affected (0.06 sec)

MariaDB [file]> insert into salesman_master values('S00003','Raj','P-7','Bandra','Mumbai',400032,'Maharashtra',3000,200,100,'Good');
Query OK, 1 row affected (0.06 sec)

MariaDB [file]> insert into salesman_master values('S00004','Ashish','A/5','Juhu','Mumbai',400044,'Maharashtra',3500,200,150,'Good');
Query OK, 1 row affected (0.06 sec)
```

```
MariaDB [file]> select * from client_master;
+----------+---------------+-------+-------+-----------+---------+-------------+---------+
| clientno | name          | add1  | add2  | city      | pincode | state       | baldue  |
+----------+---------------+-------+-------+-----------+---------+-------------+---------+
| C00001   | Ivan          | NULL  | NULL  | Mumbai    |  400054 | Maharashtra | 15000.00 |
| C00002   | Mamta Muzumdar | NULL | NULL  | Madras    |  780001 | Tamil Nadu  |     0.00 |
| C00003   | chhaya bankar | NULL  | NULL  | Mumbai    |  400057 | Maharashtra |  5000.00 |
| C00004   | Aswini Joshi  | NULL  | NULL  | Banglore  |  560001 | Karnataka   |     0.00 |
| C00005   | Hansel Colaco | NULL  | NULL  | Mumbai    |  400060 | Maharashtra |  2000.00 |
| C00006   | Deepak Sharma | NULL  | NULL  | Manglore  |  560050 | Karnataka   |     0.00 |
+----------+---------------+-------+-------+-----------+---------+-------------+---------+
6 rows in set (0.00 sec)

MariaDB [file]> select* from product_master;
+-----------+-------------+---------------+-------------+-------------+-----------+-----------+-----------+
| productno | description | profitpercent | unitmeasure | qtyonhand | reorderlvl | sellprice | costprice |
+-----------+-------------+---------------+-------------+-------------+-----------+-----------+-----------+
| P00001    | T-Shirts    |          5.00 | Piece       |         200 |         50 |    350.00 |    250.00 |
| P0345     | Shirts      |          6.00 | Piece       |         150 |         50 |    500.00 |    350.00 |
| P06734    | Cotton Jeans |         5.00 | Piece       |         100 |         20 |    600.00 |    450.00 |
| P07865    | Jeans       |          5.00 | Piece       |         100 |         20 |    750.00 |    500.00 |
| P07868    | Trousers    |          2.00 | Piece       |         150 |         50 |    850.00 |    550.00 |
| P07885    | Pull overs  |          2.50 | Piece       |          80 |         30 |    700.00 |    450.00 |
| P07965    | Denim Shirts |         4.00 | Piece       |         100 |         40 |    350.00 |    250.00 |
| P07975    | Lycra Tops  |          5.00 | Piece       |          70 |         30 |    300.00 |    175.00 |
| P08865    | Skirts      |          5.00 | Piece       |          75 |         30 |    450.00 |    300.00 |
+-----------+-------------+---------------+-------------+-------------+-----------+-----------+-----------+
9 rows in set (0.01 sec)
```

(c) Salesman - master Table:

Salesman No.	name	Address 1	Address 2	city	Pincode
S00001	Aman	A/14	Worli	Mumbai	400002
S00002	Omkar	65	Nariman	Mumbai	400021
S00003	Raj	P-7	Bandra	Mumbai	400032
S00004	Ashish	A/5	Juhu	Mumbai	400044

Salesman No.	State	Salamt	Ytdtarget	Ytd Sales	Remarks
S00001	Maharashtra 3000	3000	100	50	Good
S00002	Maharashtra 3000	3000	200	100	Good
S00003	Maharashtra 3000	3000	200	100	Good
S00004	Maharashtra 3500	3500	200	150	Good

(d) Sales - Order Table:

Order No.	Client No.	Order date	Salesman No.	Dely type	Bill Y/N	Date	Order status
019001	C00001	12-Jun-04	S00001	F	N	20-July-04	In Process
019002	C00002	25-Jun-04	S00002	P	N	27-Jun-02	Cancelled
046865	C00003	12-Feb-04	S00003	F	Y	20-Feb-02	Fulfilled
019003	C00001	03-April-04	S00001	F	Y	07-April-02	Fulfilled
046866	C00004	20-May-04	S00002	P	N	22-May-02	Cancelled
019008	C00005	24-May-04	S00004	F	N	26-July-02	En Process

```
MariaDB [file]> select * from salesman_master;
+-----------+--------+-------+---------+--------+---------+-------------+---------+---------+---------+---------+
| salesmanno | name   | add1  | add2    | city   | pincode | state       | salamt  | tgttoget | ytdsales | remarks |
+-----------+--------+-------+---------+--------+---------+-------------+---------+---------+---------+---------+
| S00001    | Aman   | A/14  | Worli   | Mumbai | 400002  | Maharashtra | 3000.00 |  100.00 |   50.00 | Good    |
| S00002    | Omkar  | 65    | Nariman | Mumbai | 400001  | Maharashtra | 3000.00 |  200.00 |  100.00 | Good    |
| S00003    | Raj    | P-7   | Bandra  | Mumbai | 400032  | Maharashtra | 3000.00 |  200.00 |  100.00 | Good    |
| S00004    | Ashish | A/5   | Juhu    | Mumbai | 400044  | Maharashtra | 3500.00 |  200.00 |  150.00 | Good    |
+-----------+--------+-------+---------+--------+---------+-------------+---------+---------+---------+---------+
4 rows in set (0.00 sec)

MariaDB [file]> select * from sales_order
    -> ;
+---------+----------+------------+----------+------------+----------+--------+------------+-------------+
| orderno | clientno | orderdate  | delyadd  | salesmanno | delytype | billyn | delydate   | orderstatus |
+---------+----------+------------+----------+------------+----------+--------+------------+-------------+
| 019001  | C00001   | 2004-06-12 | NULL     | S00001     | F        | N      | 2002-07-20 | in process  |
| 019002  | C00002   | 2004-06-25 | NULL     | S00002     | P        | N      | 2002-06-27 | Cancelled   |
| 019003  | C00001   | 2004-04-03 | NULL     | S00001     | F        | Y      | 2002-04-07 | Fulfilled   |
| 019008  | C00005   | 2004-05-24 | NULL     | S00004     | F        | N      | 2002-07-26 | in progres  |
| 046865  | C00003   | 2004-02-18 | NULL     | S00003     | F        | Y      | 2002-02-20 | Fulfilled   |
| 046866  | C00004   | 2004-05-20 | NULL     | S00002     | P        | N      | 2002-05-22 | Cancelled   |
+---------+----------+------------+----------+------------+----------+--------+------------+-------------+
6 rows in set (0.00 sec)
```

```
MariaDB [file]> select * from sales_order_details;
+---------+-----------+-------------+---------+-------------+
| orderno | productno | qtyordered  | qtydisp | productrate |
+---------+-----------+-------------+---------+-------------+
| 019001  | P00001    |           4 |       4 |      525.00 |
| 019001  | P07965    |           2 |       1 |     8400.00 |
| 019001  | P07885    |           2 |       1 |     5250.00 |
| 019002  | P00001    |          10 |       0 |      525.00 |
| 046865  | P07868    |           3 |       3 |     3150.00 |
| 046865  | P07885    |           3 |       1 |     5250.00 |
| 046865  | P00001    |          10 |      10 |      525.00 |
| 046865  | P0345     |           4 |       4 |     1050.00 |
| 019003  | P0345     |           2 |       2 |     1050.00 |
| 019003  | P06734    |           1 |       1 |    12000.00 |
| 046866  | P07965    |           1 |       0 |     8400.00 |
| 046866  | P07975    |           1 |       0 |     1050.00 |
| 019008  | P00001    |          10 |       5 |      525.00 |
| 019008  | P07975    |           5 |       3 |     1050.00 |
+---------+-----------+-------------+---------+-------------+
14 rows in set (0.00 sec)
```

(2)

Order No.	Product No.	Qty. ordered	Qty Disp	Product Rate.
019001	P00001	4	4	525
019001	P07965	2	1	8400
019001	P07885	2	1	5250
019002	P00001	10	0	525
046865	P07868	3	3	3150
046865	P07885	3	1	5250
046865	P00001	10	10	525
046865	P0345	4	4	1050
046865	P03453	2	2	1050
019003	P06734	1	1	12000
019003	P07965	1	0	8400
046866	P07975	1	0	1050
046866	P00001	10	5	525
019008	P07975	5	3	1050
019008				

```
MariaDB [file]> select name from client_master where name like '_a%';
+--------------+
| name         |
+--------------+
| Mamta Muzumdar |
| Hansel Calaco  |
+--------------+
2 rows in set (0.00 sec)

MariaDB [file]> select name from client_master where ((name like 'M%') and (city != 'NULL'));
+--------------+
| name         |
+--------------+
| Mamta Muzumdar |
+--------------+
1 row in set (0.00 sec)

MariaDB [file]> select name from client_master where ((city = 'Banglore') or (city = 'Manglore'));
+--------------+
| name         |
+--------------+
| Aswini Joshi  |
| Deepak Sharma |
+--------------+
2 rows in set (0.00 sec)

MariaDB [file]> select name from client_master where (baldue > 10000);
+------+
| name |
+------+
| Ivan |
```

```
MariaDB [file]> select * from sales_order where orderdate like '%-06-%' ;
+---------+----------+-----------+----------+------------+----------+---------+-----------+-------------+
| orderno | clientno | orderdate | delyadd  | salesmanno | delytype | billyn  | delydate  | orderstatus |
+---------+----------+-----------+----------+------------+----------+---------+-----------+-------------+
| 019001  | C00001   | 2004-06-12 | NULL    | S00001     | F        | N       | 2002-07-20 | in process  |
| 019002  | C00002   | 2004-06-25 | NULL    | S00002     | P        | N       | 2002-06-27 | Cancelled   |
+---------+----------+-----------+----------+------------+----------+---------+-----------+-------------+
2 rows in set (0.00 sec)

MariaDB [file]>  select * from sales_order so, sales_order_details sod where (so.clientno ='C00001' OR so.clientno='C00002') AND (sod.o
rderno = so.orderno);
+---------+----------+-----------+----------+------------+----------+---------+-----------+-------------+---------+-----------+-----------+
| orderno | clientno | orderdate | delyadd  | salesmanno | delytype | billyn  | delydate  | orderstatus | orderno | productno | qtyorde
red | qtydisp | productrate |
+---------+----------+-----------+----------+------------+----------+---------+-----------+-------------+---------+-----------+-----------+
| 019001  | C00001   | 2004-06-12 | NULL    | S00001     | F        | N       | 2002-07-20 | in process  | 019001  | P00001    |
  4 |       4 |      525.00 |
| 019001  | C00001   | 2004-06-12 | NULL    | S00001     | F        | N       | 2002-07-20 | in process  | 019001  | P07965    |
  2 |       1 |     8400.00 |
| 019001  | C00001   | 2004-06-12 | NULL    | S00001     | F        | N       | 2002-07-20 | in process  | 019001  | P07885    |
  2 |       1 |     5250.00 |
| 019002  | C00002   | 2004-06-25 | NULL    | S00002     | P        | N       | 2002-06-27 | Cancelled   | 019002  | P00001    |
 10 |       0 |      525.00 |
| 019003  | C00001   | 2004-04-03 | NULL    | S00001     | F        | Y       | 2002-04-07 | Fulfilled   | 019003  | P0345     |
  2 |       2 |     1050.00 |
| 019003  | C00001   | 2004-04-03 | NULL    | S00001     | F        | Y       | 2002-04-07 | Fulfilled   | 019003  | P06734    |
  1 |       1 |    12000.00 |
+---------+----------+-----------+----------+------------+----------+---------+-----------+-------------+---------+-----------+-----------+
```

Experiment - 06

Aim : Perform following operations on Tables.

(a) List the ~~value~~ names of all clients having 'a' as the second letter in their name.

→ select name from client_master where name like '_a%';

(b) List the clients who stay in a city whose first letter is 'M'

→ select name from client_master where ((name like 'M%') AND (city != NULL));

(c) List all clients who stay in 'Bangalore' or 'Mangalore':

→ select name from client_master where ((city = 'Bangalore') OR (city = 'Mangalore'));

(d) List all the clients when Baldue is greater than value 10000

→ select name from client_master where (baldue > 10000);

(e) List all ~~info~~ information from the sales order table for orders placed in month of June.

→ select * from select-orders where order date like '%-06-%'

(f) List the order information ~~for~~ client No '100001' and '100002';

→ select * sales-order so, sales-Orders details sod where (so.clientno = '100001' OR so.client No. = '100002') AND (SOD ORDERNO = SO.ORDERNO);

```
MariaDB [file]> select description from product_master where ((sellprice > 500) AND (sellprice <= 750));
+-------------+
| description |
+-------------+
| Cotton Jeans |
| Jeans        |
| Pull overs   |
+-------------+
3 rows in set (0.00 sec)

MariaDB [file]> create view products(products , new_price) as select description,sellprice*(0.15) from product_master where sellprice>5
00;
Query OK, 0 rows affected (1.69 sec)

MariaDB [file]> select* from products;
+-------------+-----------+
| products    | new price |
+-------------+-----------+
| Cotton Jeans |   90.0000 |
| Jeans        |  112.5000 |
| Trousers     |  127.5000 |
| Pull overs   |  105.0000 |
+-------------+-----------+
4 rows in set (0.00 sec)
```

```
MariaDB [file]> select name,city,state from client_master where state <> 'Maharashtra';
+---------------+---------+-----------+
| name          | city    | state     |
+---------------+---------+-----------+
| Mamta Muzumdar | Madras  | Tamil Nadu |
| Aswini Joshi   | Banglore | Karnataka |
| Deepak Sharma  | Manglore | Karnataka |
+---------------+---------+-----------+
3 rows in set (0.01 sec)

MariaDB [file]> select count(distinct orderno) from sales_order_details;
+-------------------------+
| count(distinct orderno) |
+-------------------------+
|                       6 |
+-------------------------+
1 row in set (0.09 sec)

MariaDB [file]> select avg(sellprice),avg(costprice) from product_master;
+----------------+----------------+
| avg(sellprice) | avg(costprice) |
+----------------+----------------+
|     538.888889 |     363.888889 |
+----------------+----------------+
1 row in set (0.01 sec)
```

```
MariaDB [file]> create view minmax(max_price, min_price) as (select max(sellprice),min(sellprice) from product_master);
Query OK, 0 rows affected (0.33 sec)

MariaDB [file]> select * from minmax;
+-----------+-----------+
| max_price | min_price |
+-----------+-----------+
|    850.00 |    300.00 |
+-----------+-----------+
1 row in set (0.00 sec)

MariaDB [file]> select count(*) from product_master where sellprice <=500;
+----------+
| count(*) |
+----------+
|        5 |
+----------+
1 row in set (0.00 sec)

MariaDB [file]> select description from product_master where(product_master.qtyonhead < product_master.reorderlvl);
Empty set (0.00 sec)
```

(g) list the products whose selling price is greater than 500 and less than or equal to 750.

→ Select description from product_master where ((sellprice>500) AND (sellprice<=750));

(h) list products whose selling price is more than 500. Calculate a new selling price as original selling price * 0.15. Rename the new column in the output of the above query as new price.

→ create view products (product, new price) as select description, sellprice * 0.15 from product_master where sellprice>500;
select * from products;

(i) list the names, city and state of clients who are not in state of maharashtra:

→ select name, city, state from client-master where state <> maharashtra;

(j) count the total no. of orders:

→ select count (distinct order no) from sales-order-details;

(k) calculate the average price of all products.

→ select avg (sell price), avg (cost price) from product_master

(l) count the no. of products having price less than or equal to 500.

→ select count (*) from product_master where sell price <=500;

(m) list all the products whose qty on hand is less than reorder level.

→ select description from product_master where (product_master. qty on hand < product-master. reorderlvl).

```
File  Edit  View  Search  Terminal  Help

MariaDB [file]> select clientno, orderno, date_format(orderdate,'%a') as day from sales_order order by clientno;
+----------+---------+------+
| clientno | orderno | day  |
+----------+---------+------+
| C00001   | 019001  | Sat  |
| C00001   | 019003  | Sat  |
| C00002   | 019002  | Fri  |
| C00003   | 046865  | Wed  |
| C00004   | 046866  | Thu  |
| C00005   | 019008  | Mon  |
+----------+---------+------+
6 rows in set (0.00 sec)

MariaDB [file]> select orderno, date_format(delydate ,'%M') as Month, date_format(delydate,'%d/%m/%Y') as date from sales_order;
+---------+----------+------------+
| orderno | Month    | date       |
+---------+----------+------------+
| 019001  | July     | 20/07/2002 |
| 019002  | June     | 27/06/2002 |
| 019003  | April    | 07/04/2002 |
| 019008  | July     | 26/07/2002 |
| 046865  | February | 20/02/2002 |
| 046866  | May      | 22/05/2002 |
+---------+----------+------------+
6 rows in set (0.00 sec)

MariaDB [file]>
```

```
File  Edit  View  Search  Terminal  Help

MariaDB [file]> select date_format(orderdate,'%d-%M-%y') as order_date from sales_order;
+----------------+
| order_date     |
+----------------+
| 12-June-04     |
| 25-June-04     |
| 03-April-04    |
| 24-May-04      |
| 18-February-04 |
| 20-May-04      |
+----------------+
6 rows in set (0.00 sec)

MariaDB [file]> select date_format(current_date +15 ,'%d-%M-%Y') as date_after_15_days;
+--------------------+
| date_after_15_days |
+--------------------+
| 27-March-2018      |
+--------------------+
1 row in set (0.00 sec)

MariaDB [file]>
```

Experiment - 07.

Aim : Perform following computations on Data manipulation of Tables.

(a) List the order number and day on which clients placed the order.

→ select clientno, orderno, date format (order date, '%.a') as day from sales-order order by clientNo;

(b) List the month (in alphabets) and date when the orders must be delivered.

→ select orderNo, date_format (dely date, '%.M') as month, date-format (dely date, '%.d %.M %.Y') as date from sales-order;

(c) List the order date in format 'DD-MM-YY'.

→ select date-format (order date, '%.d-%.M-%.Y') as order date from sales-order;

(d) List the date, 15 days after today date.

→ select date-format (current-date +15, '%.d-%.M-%.Y') as date_after_15_days;

```
File  Edit  View  Search  Terminal  Help

MariaDB [file]> select p.description, sum(sod.qtyordered) as qtysold from product_master p, sales_order_details sod where p.productno=s
od.productno group by p.productno;
+-------------+----------+
| description | qtysold  |
+-------------+----------+
| T-Shirts    |       34 |
| Shirts      |        6 |
| Cotton Jeans|        1 |
| Trousers    |        3 |
| Pull overs  |        5 |
| Denim Shirts|        3 |
| Lycra Tops  |        6 |
+-------------+----------+
7 rows in set (0.00 sec)

MariaDB [file]> select p.description, sum(sod.productrate) as value from product_master p,sales_order_details sod where p.productno=sod
.productno group by p.productno;
+-------------+----------+
| description | value    |
+-------------+----------+
| T-Shirts    |  2100.00 |
| Shirts      |  2100.00 |
| Cotton Jeans| 12000.00 |
| Trousers    |  3150.00 |
| Pull overs  | 10500.00 |
| Denim Shirts| 16800.00 |
| Lycra Tops  |  2100.00 |
+-------------+----------+
7 rows in set (0.00 sec)
```

```
File  Edit  View  Search  Terminal  Help

MariaDB [file]> select s.clientno,avg(sod.qtyordered) as avg_qty_sold from sales_order s,sales_order_details sod where s.orderno=sod
.orderno group by s.clientno having( sum(sod.productrate) <=15000);
+----------+--------------+
| clientno | avg_qty_sold |
+----------+--------------+
| C00002   |      10.0000 |
| C00003   |       5.0000 |
| C00004   |       1.0000 |
| C00005   |       7.5000 |
+----------+--------------+
4 rows in set (0.00 sec)

MariaDB [file]> select sum(sod.productrate) as total_of_billed_order from sales_order_details sod,sales_order so where (sod.orderno=
so.orderno)and(month(so.orderdate)=6) group by month(so.orderdate);
+----------------------+
| total_of_billed_order |
+----------------------+
|              14700.00 |
+----------------------+
1 row in set (0.00 sec)

MariaDB [file]>
```

Experiment-08.

Aim: To perform following operations by using having and groupby clause using tables:

(a) Print the description and total qty sold for each product

→ select p.description, sum(sod.qtyordered) as qtyhold from product_master, sales-order_details sod where p.productno = sod.productno group by p.productno;

(b) Find the value of each product sold.

→ select p.description, sum(sod.productrate) as value product_master p, sales-order_details sod where p.productno = sod.productno group by p.productno;

(c) Calculate the average quantity sold for each client that has a maximum order value of 15000.00.

→ select s.clientno, avg(sod.qtyordered) as avg-qty-sold sales_order s, sales-order_details sod where s.orderno = sod.orderno group by s.clientno having (sum(sod.productrate) <= 15000);

(d) Find out the total of all the billed orders for month of June.

→ select sum(sod.productrate) as total_of_billed_orders from sales-order_details sod, sales_order so where (sod.orderno = so.orderno) AND (month(so.order date) = 6) group by month(so.order date).

```
File  Edit  View  Search  Terminal  Help
Empty set (0.00 sec)

MariaDB [file]> select p.description from client_master c join sales_order so on c.clientno=so.clientno join sales_order_details sod on
 so.orderno=sod.orderno join product_master p on sod.productno=p.productno where c.name ='Ivan';
+--------------+
| description  |
+--------------+
| T-Shirts     |
| Denim Shirts |
| Pull overs   |
| Shirts       |
| Cotton Jeans |
+--------------+
5 rows in set (0.00 sec)

MariaDB [file]> select p.description, sum(sod.qtyordered) from sales_order so join sales_order_details sod on sod.orderno=so.orderno jo
in product_master p on sod.productno=p.productno where month(so.delydate)=month(current_date) group by p.productno;
Empty set (0.00 sec)

MariaDB [file]> select p.productno,p.description from sales_order_details sod join product_master p on p.productno=sod.productno where
productrate =(select max(productrate) from sales_order_details);
+-----------+-------------+
| productno | description |
+-----------+-------------+
| P06734    | Cotton Jeans |
+-----------+-------------+
1 row in set (0.00 sec)

MariaDB [file]>
```

```
File  Edit  View  Search  Terminal  Help

MariaDB [file]> select name from product_master p,client_master c,sales_order so,sales_order_details sod where((p.description='Trousers
') AND (p.productno=sod.productno) AND (sod.orderno=so.orderno) AND (so.clientno=c.clientno));
+--------------+
| name         |
+--------------+
| Chhaya bankar |
+--------------+
1 row in set (0.00 sec)

MariaDB [file]> select p.productno,sod.orderno from product_master p, sales_order_details sod where p.description='Pull overs' AND p.pr
oductno=sod.productno AND qtyordered<5;
+-----------+---------+
| productno | orderno |
+-----------+---------+
| P07885    | 019001  |
| P07885    | 046865  |
+-----------+---------+
2 rows in set (0.00 sec)

MariaDB [file]> select name,description,qtyordered from client_master c, product_master p,sales_order so, sales_order_details sod where
 name IN('Ivan', 'Mamta Muzumdar') AND c.clientno=so.clientno AND so.orderno=sod.orderno AND sod.productno=p.productno;
+---------------+-------------+------------+
| name          | description | qtyordered |
+---------------+-------------+------------+
| Ivan          | T-Shirts    |          4 |
| Ivan          | Denim Shirts |          2 |
| Ivan          | Pull overs  |          2 |
| Ivan          | Shirts      |          2 |
| Ivan          | Cotton Jeans |          1 |
| Mamta Muzumdar | T-Shirts    |         10 |
+---------------+-------------+------------+
6 rows in set (0.00 sec)
```

Experiment - 9.

Aim: To perform exercises on joins and correlation tables.

(a) Find out the products which have been sold to 'Ivan Bayross'

→ select p.description from client_master c join sales_order so on c.clientno = so.clientno join sales_order_details sod on so.orderno = sod.orderno join product_master p on sod.productno = p.productno where c.name = 'Ivan';

(b) Find out the products & their quantities that will have to be delivered in the current month.

→ select p.description, sum(sod.qtyordered) from sales_order so join sales_order_details sod on sod.orderno = so.orderno join product_master p on sod.productno = p.productno where month(so.delydate) = month(current_date) group by p.productno;

(c) List the productno. and description of constantly sold products.

→ select p.productno, p.description from sales_order_details sod join product_master p, sales_order so, client_master c on p.productno = sod.productno where (select max(productrate) from sales_order_details);

(d) List the products & orders from customers who have ordered less than 5 units of pullovers.

→ select p.productno, sod.orderno from product_master p, sales_order_details sod where p.description = 'Pull overs' AND p.productno = sod.productno AND qtyordered < 5;

File Edit View Search Terminal Help

```
MariaDB [file]> select c.clientno, description,qtyordered from sales_order_details sod join
sales_order so on so.orderno=sod.orderno join product_master p on sod.productno=p.productno
join client_master c on c.clientno=so.clientno where so.clientno in('C00001','C00002');
+----------+--------------+------------+
| clientno | description  | qtyordered |
+----------+--------------+------------+
| C00001   | T-Shirts     |          4 |
| C00001   | Denim Shirts |          2 |
| C00001   | Pull overs   |          2 |
| C00001   | Shirts       |          2 |
| C00001   | Cotton Jeans |          1 |
| C00002   | T-Shirts     |         10 |
+----------+--------------+------------+
6 rows in set (0.00 sec)
```

(2) Find the products and their quantities for the orders placed by 'Ivan Bayross' and 'Mamta Muzumdar'.

→ select name, description, qtyordered from client_master c, product_master p, sales.order so, sales_order_details sod where name IN ('Ivan', 'Mamta Muzumdar') AND c.clientno = so.clientno AND so.orderno = sod.orderno AND sod.productno = p.productno;

(3) find the products and their quantities for the ordered placed by client No 'C00001' & 'C00002'.

→ select c.clientno, description, qtyordered from sales_order_details sod join sales_order so on so.orderno = sod.orderno join product_master p on sod.productno join client_master c on c.clientno = so.clientno where clientno IN ('C00001', 'C00002');

Experiment -10.

Aim : To perform following computations by creating views of tables created earlier.

(a) Create a view having orderNo and clientNo:

→ create view clntorderno AS (select orderno, clientno from sales order);
select * from clntorderno;

(b) In above view, change clientNo to 'C00006'

→ update clntordrno set clientNo = 'C00006' where orderno = 'O19008';
select * from clntordrno;

```
File  Edit  View  Search  Terminal  Help
MariaDB [file]> create view clntordrno AS(select orderno,clientno from sales_order);
Query OK, 0 rows affected (0.07 sec)

MariaDB [file]> select *from clntordrno;
+---------+----------+
| orderno | clientno |
+---------+----------+
| O19001  | C00001   |
| O19003  | C00001   |
| O19002  | C00002   |
| O46865  | C00003   |
| O46866  | C00004   |
| O19008  | C00005   |
+---------+----------+
6 rows in set (0.00 sec)

MariaDB [file]>
```

```
File  Edit  View  Search  Terminal  Help

MariaDB [file]> update clntordrno set clientno='C00006' where orderno='O19008';
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [file]> select *from clntordrno;
+---------+----------+
| orderno | clientno |
+---------+----------+
| O19001  | C00001   |
| O19003  | C00001   |
| O19002  | C00002   |
| O46865  | C00003   |
| O46866  | C00004   |
| O19008  | C00006   |
+---------+----------+
6 rows in set (0.00 sec)

MariaDB [file]>
```

```
File  Edit  View  Search  Terminal  Help

MariaDB [file]> create view sale AS (select s.orderno,orderdate,orderstatus,productno,productrate,qtyordered from sales_order s,
sales_order_details sod where sod.orderno=s.orderno);
Query OK, 0 rows affected (0.05 sec)

MariaDB [file]> select *from sale
    -> ;
+---------+-----------+-------------+-----------+-------------+-----------+
| orderno | orderdate | orderstatus | productno | productrate | qtyordered |
+---------+-----------+-------------+-----------+-------------+-----------+
| 019001  | 2004-06-12 | In process | P00001    |      525.00 |         4 |
| 019001  | 2004-06-12 | In process | P07965    |     8400.00 |         2 |
| 019001  | 2004-06-12 | In process | P07885    |     5250.00 |         2 |
| 019002  | 2004-06-25 | Cancelled  | P00001    |      525.00 |        10 |
| 019003  | 2004-04-03 | Fulfilled  | P0345     |     1050.00 |         2 |
| 019003  | 2004-04-03 | Fulfilled  | P06734    |    12000.00 |         1 |
| 019008  | 2004-05-24 | In progres | P00001    |      525.00 |        10 |
| 019008  | 2004-05-24 | In progres | P07975    |     1050.00 |         5 |
| 046865  | 2004-02-18 | Fulfilled  | P07868    |     3150.00 |         3 |
| 046865  | 2004-02-18 | Fulfilled  | P07885    |     5250.00 |         3 |
| 046865  | 2004-02-18 | Fulfilled  | P00001    |      525.00 |        10 |
| 046865  | 2004-02-18 | Fulfilled  | P0345     |     1050.00 |         4 |
| 046866  | 2004-05-20 | Cancelled  | P07965    |     8400.00 |         1 |
| 046866  | 2004-05-20 | Cancelled  | P07975    |     1050.00 |         1 |
+---------+-----------+-------------+-----------+-------------+-----------+
14 rows in set (0.00 sec)

MariaDB [file]>
```

Experiment - 11

Aim : Write an introduction on PL/SQL, triggers, cursors and Stored Procedures.

PL/SQL :

It is a combination of SQL along with the procedural features of programming languages. It was developed by Oracle corporation in early 90's to enhance the capabilities of SQL. It is one of the 3 key programming languages embedded in the oracle database along with SQL & java.

Basic syntax of PL/SQL :

1. Declaration - This section starts with the keyword DECLARE It is an optional section and defines all variables, cursors etc.
2. Executable command - This section is enclosed b/w the keywords BEGIN and END and it is a mandatory section. It consists of executable PL/SQL statements of the program.
3. Exception Handling - This section starts with the keyword 'Exception'. This optional section contains exception(s) that handle errors in the program.

Example :

```
DECLARE
    message varchar2(20) := 'Hello, World!';
BEGIN
    dbms_output.put_line(message);
END;
```

Triggers :

They are stored programs which are automatically executed or fired when some events occur. They are written to be executed in response to any of following events :-

- A DML statement (Delete, Insert or update)
- A DDL statement (create, alter or drop)
- A database operation (server error, logon, logoff)

Triggers can be defined on the table view, schema or database with which the event is associated.

Syntax - create [or replace] Trigger trigger_name

{ Before | After | Instead of }

{ Insert [OR] | Update [OR] | DELETE }

[OF col_name]

on table_name

[Referencing old as new as]

[for each Row]

when (condition)

DECLARE { }

BEGIN { }

Exception ()

END;

Cursors : A cursor is a pointer to this context area which is a memory space created by Oracle. PL/SQL controls the context area through a cursor. A cursor holds the rows.

Syntax for cursors :

- Declared as a variable in the same way as standard variable
- Identified as cursor type SQL included.

Eg : Cursor cur_emp is
select emp_id, surname, name.
from employee
where grade is true ;

Stored procedure :

A stored procedure (also termed as proc, sproc, sproc, pro, storedpro, storepc, sporsp) is a subroutine, a predefined batch of code, available to the applications that access a RDBMS.

Eg : DELIMITER $$
create Procedure hello_world2()
BEGIN
SELECT * from student orderBy am;
SELECT * from course orderBy course_id;
END $$
call Helloworld2();

Experiment 12

Aim : Perform the comparative analysis of different types of databases on various parameters.

SNO	HIERARCHICAL DATA MODEL	NETWORK DATA MODEL	RELATIONAL DATA MODEL
1	Relationship between records is of parent child type.	Relationship between records is expressed in form of pointers or links.	Relationship between records is represented by a relation that consists a key for each record involved in a relationship.
2	Many- to- many relationship cannot be expressed in this model.	Many- to- many relationship can also be implemented.	Many- to- many relationship can be easily implemented.
3	It is a simple straight forward and natural method of implementing record relationship.	Records relationship implementation is quite complex due to use of points	Relationship implementation is very easy though the use of key or composite key field .
4	This type of model is useful only when there is some hierarchical characteristics in database.	Network model is useful for representing such records which have many-to-many relationship.	Relationship model is useful for representing most of the real world objects and relationship among them.
5	In order to represent links among records pointer are used thus relationship among records are physical.	In network model also the relationship among records are physical.	Relational model does not maintain physical connection among records. Data is organized logically in the form of rows and columns and stored in table.
6	Searching for record is very difficult since one can retrieve a child only after going through parent records.	Searching a record is easy since there are multiple access path to a data element.	A unique, indexed key field is used to search for a data element.
7	During updating or deletion process, change of data is inconsistent .	No problem of inconsistency exists in network model because a data element is physically located at just one place.	Data integrity maintaining methods like Normalization process etc. Are adopted for consistency.

Karan

International Journal of Computer Applications (0975 – 8887)
Volume 85 – No 13, January 2014

Security Issues and Their Techniques in DBMS - A Novel Survey

Mohd Muntjir
College of Computers and Information Technology
Taif University, KSA

Sultan Aljahdali
College of Computers and Information Technology
Taif University, KSA

Mohd Asadullah
College of Computers and Information Technology
Taif University, KSA

Junedul Haq
College of Computers and Information Technology
Taif University, KSA

ABSTRACT
Nowadays a Database security has become an important issue in technical world. The main objective of database security is to forbid unnecessary information exposure and modification data while ensuring the availability of the needed services. A numbers of security methods have been created for protecting the databases. Many security models have been developed based on different security aspects of database. All of these security methods are useful only when the database management system is designed and developing for protecting the database. Recently the growth of web application with database at its backend Secure Database Management System is more essential than only a Secure Database. Therefore this paper highlight on the Threats, Security Methods and Vulnerabilities in Database Management System with the help of survey performed on the field of secure databases.

Keywords
Vulnerability, threats, security methods, DBMS

1. INTRODUCTION
These days including the invention of internet technology securing database is a needed aspect in today's world. Individually we use database every day unknowingly when we browse on internet. The information we get on the web page is the consequences of query accomplished by the webpage to the database it is connected. Hence indirectly via the webpage we are connected to different databases. The web pages are open for any anonymous person in the world or we can say the databases are indirectly opened for everyone. As we know data in the database is the most valuable asset which can be the source of information. All the information cannot be revealed for everyone. Hence many security tools have been devised to protect the database. As the database is accessible via web pages security should be implemented in database management system (DBMS).Looking towards the implementation this paper focus on Vulnerabilities in Database Management System (VDBMS), Threats in Database Management System (TDBMS) and Security Methods in Database Management System (SMDBMS).

Rest of the paper is organized as follows: section II provides overview of recent trends in database protection, section III, IV and V are devoted to VDBMS, TDBMS and SMDBMS.

Section VI deals with the summary of above section in a tabular format and section VII deals with the conclusion.

2. PROTECTED DATABASE
There are many ways of securing the database. These ways are based on different aspects of securing the database. Different aspects with traditional approaches from different researchers view are summarized below:

2.1 Confidentiality, Integrity and Availability (CIA) in Database Management System
As mentioned in [1] a complete solution to data security must fulfilled the following three requirements Confidentiality, Integrity, Availability (CIA): these entire factors can gained in database using following ways:

2.1.1 Confidentiality
Means to the protection of data against unauthorized disclosure can be achieved using access control mechanism. It is already further enhanced by the use of encryption techniques is applied to data when being stored on secondary storage or transmitted on a Network.

2.1.2 Integrity
Means to the prevention of unauthorized and improper data modification and can be achieved in combination of access control mechanism by semantic integrity constraints.

2.1.3 Availability
Means to the prevention and recovery from hardware and software errors and from malicious data access denials making the database system inaccessible. The data that are available on the Web can be powered by the use of techniques protecting against denial-of-service attacks and such as the ones based on machine learning techniques.

Fig. 1 CIA Triad

2.2 Different Aspects
Latest approaches of protecting database are illustrated in [2]. All of these approaches are related to CIA. In these approaches the author proposes that it can be implemented with the help of below listed appropriate techniques:

2.2.1 Authentication of Users

Within this point the author has mentioned about public key encryption (PKI) for the databases that require higher levels of safety one-time passwords X.509 digital certificate smart cards can be used. PKI is very useful when contacting over irrelevant networks like the Internet and both on the internal servers.

2.2.2 Access control to objects and authentication of authorized applications

In this point means the access control should be defined at the design state. Here main emphasis is given on the roles and based on this access is given to the user.

2.2.3 Administration policies and procedure

Its mean Security and safety policies with plans are required for varying requirements of data security.

2.2.4 Secure initial configuration

This point indicates the Policies and procedures also define auditing requirements for securing initial configuration and managing change regulation.

2.2.5 Auditing

This point refers for auditing the author emphasizes on maintaining logs of the changes to the database management system.

2.2.6 Backup and recovery strategies

This point refers to the backup and strategies, As the backup and recovery there should be three kinds of backup's cold, hot and logical. All these aspects are traditional and there are vulnerabilities in these security methods which may cause threats to the database system. Henceforth this paper gives the detailed information about the vulnerabilities, threats and different security methods to avoid them.

3. ABOUT THE VULNERABILITIES IN DATABASE MANAGEMENT SYSTEM (VDBMS)

Based on our survey conducted the vulnerabilities in database are defined as: poor architecture, misconfigurations, and vendor bugs incorrect usage [2].

3.1 Vendor bugs refer to buffer overflows and other programming errors that result in users executing the commands they are allowed to be execute. Furthermore Downloading and applying patches usually fix vendor bugs and viruses.

3.2 Poor architecture refers the result of inadequate factoring security into the design of how an application works there. These vulnerabilities are typically the hardest to fix because they require a major rework by the vendor. We can give an example of poor architecture; it would be when a vendor utilizes a weak form of inscription.

3.3 Misconfigurations are caused by not accurately locking down databases. Mostly the configuration options of databases can be set in a way that compromises security and safety for that database. Some of these parameters are concluded insecurely by default. But mostly it is not a problem unless you unsuspectingly change the configuration and setting. An example of this in Oracle is the REMOTE_OS_AUTHENT parameter. When you set REMOTE_OS_AUTHENT to true you are allowing unauthenticated users to connect to your database, so that he can do his task correctly.

3.4 Incorrect usage means to building applications utilizing developer tools in ways that can be used to break into a database. SQL injection is an example of incorrect usage for developer.

The authors Marco Vieira and Henrique Madeira [3] have defined that the vulnerabilities in DBMS are an internal factor related to the set of security mechanisms available or not available in the database, the correct configuration of those mechanisms (it is a responsibility of the DBA), and the hidden flaws on the system configuration.

He has described that security in database can be violated due to points as given below:

3.5 Irresponsible DBA: Refers to deactivation of the necessary security mechanisms such that user privileges, authentication, auditing, data encryption which allows intruders to find a way to getting access the data into database.

3.6 Incorrect configuration: Permits unauthorized users or hackers to access the data in our system.

3.7 Hidden flaws in the database: May allow hackers to connect to the database server by exploring those faults.

3.8 Unauthorized users: Means these users "still" the credentials of authorized users in order to access the database server for searching the data.

3.9 Misused Privileges: Refers to authorized users take advantage of their privileges to maliciously access or destroy our data in a database.

Vulnerabilities are also defined by Hassan A. Afyouni [4] in the following manners:

3.10 Configuration and Installation: Using a default installation and configuration that is known by publicly. For example failure to change default password or default privileges or permissions.

3.11 User Mistakes: Sometimes Carelessness in implementing procedures failures to follow directly, or accidental errors with some faults. For example users lack bad authentication process, technical information or implementation, untested disaster recovery plan in a database.

3.12 Software: Refers to vulnerabilities found in commercial software for all types of programs such that all applications, operating systems, database management systems and network systems with other different programs.

3.13 Design and Implementation: Inaccurate software analysis and design as well as coding problems and faults may lead to vulnerabilities in a database.

4. ABOUT THE THREATS IN DATABASE MANAGEMENT SYSTEM (TDBMS)

Threat in database is defined by Aziah Asmawi in [5] as a set of policies, measures and mechanisms to provide safety, availability and integrity of data and to combat possible attacks on the system from outsiders as well as insiders, both accidental and malicious. Aziah Asmawi has mentioned about SQL injection which can be executed by two ways by unauthorized user accessing the database via webpage connected network:

4.1 Access through login page: This is the easiest technique in which it bypasses the login forms where users are authenticated by using password. This type of technique can be done by the attackers through *'or' condition, 'having' clause, multiple queries and executed stored procedure with package.*

4.2 Access through URL: The attackers use this technique through manipulating the query string in URL and using the SELECT and UNION statements.

Further Ravi Sandhu [1] has described in his paper that threat to the database can be internal or external. By this technique he has characterized the security breach as incorrect data modification, unauthorized data observation and data unavailability.

As mentioned in [4] types of threats are following:

4.3 People: In this point the different people involved in database management system can be a government authority , an employee or a person in charge, consultants, contractors, visitors, hackers, organized criminals, spies, terrorists and social engineers may deliberately or unintentionally exact damage to any of the database environment factor.

4.4 Malicious Code – Refers to Software code , in which most cases is intentionally written to damage or violate one or more of the database environment components are boot sector worms, viruses, spoofing code Trojan horses, denial-of-service flood, bots, root kits, bots.

E-mail spamming, macro code.

4.5 Natural disaster: Calamities caused by nature can destroy any of the entire database environment components.

Technological disasters: Refers to Some sort of malfunction in hardware or equipment, technological disorders like media failure, hardware failure, power failure, or network failure can inflict damage to database management systems, data files or data or whole database.

5. SECURITY METHODS IN DATABASE MANAGEMENT SYSTEM (SMDBMS)

Here we will discuss about some security methods in DBMS. In early days security methods in database management system focus only on role base access control or maintaining the confidentiality or authenticity of the database. But in the current scenario the unauthorized user working on a web page which is connected via internet connection has access to the database, since all the queries sent by the user is converted to SQL query in the database. The user may send malicious query and confirm or modify the transactions of the database without affecting the performance of the database. This type of attack is called SQL injection. But in the current scenario the security method of database should focus on role base access control and maintain CIA and avoid attacks due to network. This section emphasizes the same, based on various papers and books available on similar topic or same issue.

5.1 A SECURING DATABASE BASED ON ACCESS CONTROL:

- In this section we will discussed about the database security based on access control. The role based access control method has been proposed by Guoliang Zou, Jing Wang, Dongmei Huang [6] where he has implemented security using the following points:
- Preventing illegal users from logging the system
- Identity validation
- Access Control Interface
- Verification codes
- Database security: storage procedure
- Database security: oracle parameter

The author Ravi Sandhu has created various security approaches [10] where he has considered that access control policies in early days were based on the development of two different classes of models, the discretionary access control policy and on the required access control policy and procedure. Based on these models of early days [7] have proposed two assumptions:

5.1.1 The first assumption was that the access control models for databases should be defined in terms of the logical data model; hence authorizations for a relational database should be defined in terms of relational model such as relations, relation attributes and tuples etc.

5.1.2 The second assumption is that for databases, in accession to name-based access control, where the secure and protected objects are categorized by giving their names, content-based access control has to be promoted.

Discretionary access control policy has subsidized in the creation and development of System R access control for relational database management system which altered strongly on some key features such as distributed authorization administration, effective grant and revoke of authorizations and the use of views for supporting and developing content-based authorizations. Furthermore the access control policies of an object oriented database (OODBMS) are defined in [8]. Here in this point the author has discussed about two proposed security models for OODBMS. They are given below as:

5.1.2.1 Sorion Security Model: This is a security model proposed by Thuraisingham to associate a secure access control into the ORION model system.

5.1.2.2 Jajodia-Dogan Security Model: Jajodia-Dogan

(6, 12) has proposed a security model for OOBBMS that control access by using the encapsulation characteristic of object oriented database.

Henceforth using the access control policies and procedure the confidentiality of the database can be supported.

The second security issue of database management systems has various fields of database integrity as described in [5]:

Physical database integrity protection: It manages data integrity through physical obstacles such as fires and power failures.

Logical data integrity protection: It refers to the assertion that information is can be changed only by users.

Data element integrity protection: It involves data efficiency and data regularity.

And the third security issue availability as described above belongs to the data availability from the database management system. Henceforth, Due to the availability of company's whole information on the web page which is connected via Internet to its database, the whole data of the company is available using the SQL injection. Thus below section describes the security methods to prevent SQL injection in the scenario.

5.2 SOME SECURITY METHODS TO PREVENT SQL INJECTION

Hence as a protection from SQL injection many Intrusion Detection Systems (IDS) have been suggested. A brief description of these IDS is discussed below:

5.2.1 Misuse Detection System for DBMS

(DEMIDS): This method has been proposed by Chung *et al.* (1999). It is called a misuse-detection system, created for relational databases. It uses audit data log to retrieve profiles describing typical behaviour of users in Database Management System.

The method is present by Lee *et al.* (2000). This method is based on intrusions. Hence this method has used time signatures to discover database intrusions.

On the other way similar work was proposed by Low *et al.*(2002).This method is used for Detecting Intrusion in Databases through Fingerprinting Transactions (DIDAFIT).It is a system created using misuse detection approach to show database intrusion detection at the application level in a database.

But another approach towards a database specific intrusion detection mechanism is by Hu and Panda (2003). They proposed and developed a mechanism that is more capable of finding data dependency relationships among transactions and use this information to find hidden anomalies in a database log. Ke Chen *et al.* (2005) developed an intrusion detection model for a database system based on digital amnesty. It gives an additional layer of security against DBMS misuse.

On other hand a real-time intrusion detection mechanism based on the profile of user roles has been prescribed by Bertino *et al.* (2005). This total approach is based on mining SQL queries stored in audit log files in a database.

Rietta (2006) described an application layer intrusion detection system, which should take the form of a proxy server and apply an anomaly detection model based on distinct characteristics of SQL and the transaction history of a appropriate user application and user.

Aziah Asmawi has proposed SQL Injection and Insider Misuse Detection System (SIIMDS) in 2008 to define both types of intrusions from external and internal threats.

Malicious users may access a series of safe information and then apply different techniques to retrieve sensitive data by using that information. To address this inference problems,Yu Chen in [9] has created a semantic inference model (SIM) that symbolize all the possible inference channels from any attribute in the system to the set of elevated sensitive attributes. Hence based on the SIM, the violation detection system keeps track of a user's query history in a database. When a new query is stiffed, all the channels where sensitive information can be stored will be recognizing. If the probability of inferring sensitive information increased a more specified threshold, then the current query request will be revoked. Using the security methods mentioned in section A and B secure and safe database can be created. It may be accessed from anywhere and the security would be managed.

Even though there is no such thing as a 100 percent guarantee in network security, awful obstacles can be placed in the path of SQL injection attack. Anybody of these defenses extremely reduces the chances of a successful SQL injection attack to prevent our data. Implementing all four is a best practice that will supply high degree of protection and safety. Despite its extensive application, your web site does not have to be SQL injection's next suspect. The next section briefs up all the vulnerabilities, threats and security methods of database management system in tabular format which will be beneficial for the development of secure and safe database. There actually is a lot method that web site owners can do to secure against SQL injection attack.

Table 1. Details of VDBMS, TDBMS and SMDBMS

Vulnerabilities (VDBMS)		THREATS (TDBMS)	SECURITY METHODS SDBMS)
Vendor Bug	Buffer Overflow, Programming errors	May damage or violate the database	Unauthorized access control policy
Poor Architecture	Weak form of encryption	May damage database environment components (networks, applications, operating systems, DBMS and data)	1.Sorion Security Model 2.Jajodia-Dogan Security Model
Misconfiguration	Not properly locking database	Loss of integrity of the database	1.Physical database integrity protection 2.Logical data integrity protection 3.Data element integrity protection
Incorrect usage	SQL injection	Misuse of availability of database	Intrusion Detection System like 1. A Misuse Detection System for Database System (DEMIDS) 2.SQL Injection and Insider Misuse Detection System (SIIMDS)

			3. Detecting Intrusion in Databases through Fingerprinting Transactions (DIDAFIT) 4. Semantic inference model (SIM)
Irresponsible DBA	Deactivation of necessary security mechanism	Easy access of data	Two principles should be followed: 1. The access control models for databases should be expressed in terms of the logical data model; thus authorizations for a relational database should be expressed in terms of relations, relation attributes, and tuples. 2. For databases, in addition to name-based access control, where the protected objects are specified by giving their names, content-based access control has to be supported.
Hidden Flaws in DB	Undetected defects	Allow hackers to connect to the database server by exploring those defects.	Intrusion Detection System
Unauthorized Users	Unauthorized users "still" the credentials of authorized users	Easy access of database servers.	Intrusion Detection System
Misused Privileges	Authorized users take advantage of their privileges.	Maliciously access or destroy data	Database Administrator should provide security on the basis of above mentioned principles.

6. CONCLUSION

In this paper we have identified the vulnerabilities, threats and security methods of database management system with the help survey conducted on researches of database security. The result of the survey we have described in the paper and summarized in tabular form. As a result we can conclude that though remarkable work has been done in this field, with the invention of internet technology, the risk to database has increased. Many intrusion detection systems for the database have been devised still more research has to be done since there are vulnerabilities in internet connection and website.

7. REFERENCES

[1] Elisa Bertino, Fellow, IEEE, and Ravi Sandhu, Fellow, IEEE, "Database Security—Concepts, Approaches and Challenges" in IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 2, NO. 1, JANUARY-MARCH 2005

[2] Andriy Furmanyuk , Mykola Karpinskyy, Bohdan Borowik, "Modern Approaches to the Database Protection" in IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 6-8 September 2007, Dortmund, Germany

[3] Marco Vieira, Henrique Madeira , "Detection of Malicious Transactions in DBMS", 11th Pacific Rim International Symposium on Dependable Computing

[4] Hassn A. Afyuoni, A Book, "Database security and auditing "

[5] Aziah Asmawi , "System Architecture for SQL Injection and Insider Misuse Detection System for DBMS", my -1-4244-2328 6/08/$25.00 © 2008 IEEE

[6] Guoliang Zou, Jing Wang, Dongmei Huang, LiangJun Jiang, "Model Design of Role-Based Access Control and Methods of Data Security", 2010 International Conference on Web Information Systems and Mining.

[7] E.B. Fernandez,R.C. Summers and C.Wood, Database Security and Integrity. Addison-Wesley, Feb. 1981.

[8] Premchand B. Ambhore,B.B.Meshram,V.B.Waghmare, "A IMPLEMENTATION OF OBJECT ORIENTED DATABASE SECURITY" , Fifth International Conference on Software Engineering Research, Management and Applications.

[9] Yu Chen and Wesley W. Chu, "Protection of Database Security via Collaborative Inference Detection ", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 20, NO. 8, AUGUST 2008

[10] http://www.esecurityplanet.com/hackers/how-to-prevent-sql-injection-attacks.html

Experiment - 13.

SUMMARY OF RESEARCH PAPER

Now-a-days a database security has become an important issue in technical world. The main objective of database security is to forbid unnecessary information exposure and modification data while ensuring the availability of needed services.
A number of security methods have been created for protecting database Many security models have been developed on different security aspects of database. All of these security methods are useful only when the database management system is designed and developing for protecting the database. Recently the growth of web application with database at its backend secure Database management system is more essential than only a secure Database.
Therefore this paper highlight on the Threats, security methods and vulnerabilities in Database management system with the help of survey performed.

CONCLUSION :

In this paper we have identified the vulnerabilities, threats and security methods of dbms. We can conclude that though remarkable work has been done in this field with the invention of internet technology, the risk to database has increased.

Viva - Questions:

Ques.1. What is a database?

A database management system (DBMS) is a software package designed to define, manipulate, retrieve and management in database.

Ques.2. What is SQL?

SQL (Structured Query Language) is a language for accessing and manipulating databases.

Ques.3. What are the different types of SQL commands?

SQL commands are :

Data Definition Language (DDL)
Data Manipulation Language (DML)
Data Control Language (DCL)

Ques.4. Explain DDL commands. What are the different DDL commands in SQL?

Commands	Description
CREATE	create new database/table
ALTER	Modifies the structure of database/table
DROP	DELETES a database/table.
RENAME	Rename the database/table.

Viva - Questions:

Que1. What is a NULL value and how does it differ from a zero value?

A field with a NULL value is a field with no value.
Zero is a number in the set of real numbers with empty magnitude while NULL is a term used to denote empty.

Que2. What are SQL Constraints?

→ Used to specify rules for data.
Example:

NOT NULL

UNIQUE.

PRIMARY KEY.

Que3. What are different Data Types in SQL?

TEXT :

1. CHAR
2. VARCHAR

NUMBER:

1. INT
2. FLOAT
3. DOUBLE

Que4. What is the difference between CHAR and VARCHAR?

CHAR is used to store character string value of fixed length. whereas VARCHAR is used to store variable length alphanumeric data.

Viva- Questions:

Que1. What is the use of SELECT command?

It is used to select data from database.

```
SELECT column1, column2, ....
FROM table_name;
```

Que2. What is the purpose of WHERE clause?

It is used to extract only those records that fulfill a specified condition.

WHERE syntax

```
SELECT column1, column2.
FROM table_name.
WHERE condition
```

Ques3. Which language supports SELECT command?

SQL (Structured Query language).

Que4. How to select entire content of a table.

```
SELECT * FROM table_name.
```

Viva- Questions:

Que1. What are different DML commands?

Commands	Description
SELECT	Retrieve data
INSERT	Insert data
UPDATE	update in table.
DELETE	Delete records

Que2. What is the purpose of ALTER command .What is the Syntax?

It is used to add, delete or modify columns in an existing table. Also used to drop

```
ALTER TABLE table_name.
  ADD column_name datatype;
```

Que3. How we add a Column in a table?

```
ALTER TABLE table_name.
ADD column_name datatype.
```

Que4. What is the purpose of DELETE command?

```
DELETE FROM table_name.
WHERE condition;
```

Que5. What is difference between ALTER and UPDATE commands?

ALTER is a DDL statement. Whereas UPDATE is a DML statement

Viva – Questions

Que1. What are different Constraints in SQL?

NOT NULL
UNIQUE
PRIMARY KEY
FOREIGN KEY
CHECK.
DEFAULT
INDEX.

Que2. What is the purpose of Null Constraint?

NOT NULL ensures that column cannot have a NULL value.

Que 3. What is the function of Reference Constraint?

Reference is used with foreign key constraint when child table have a reference column in base table.

Que 4. What is Index Constraint?

INDEX constraint is used to create & retrieve data from the database very quickly.

Que 5. What is the purpose of Default Constraint.

It sets a default value for a column when no value is specified.

Viva- Questions:

Que1. What id Date command?

It extracts the date part of a date or datetime expression

Que2. What is Month function?

It returns the month from the date passed in argument

Que3. What is the syntax of Date Command?

```
SELECT DATE ('2003-12-31 01:02:03');
```

Que4. Write the function of Month Function.

```
SELECT MONTH('1998-02-03');
```

Que5. What is difference between DATETIME and TIMESTAMP command?

DATETIME represents a date and a time while TIMESTAMP represents a well defined point in time.

Viva – Questions:

Que1. Explain HAVING and GROUPBY clause.

The GROUP BY clause is a SQL command that is used to group rows that have the same values.
For performing grouping on certain data we use HAVING fn.

Que2. What are the syntax of HAVING clause?

```
SELECT column_name(s).
FROM table_name.
WHERE condition.
GROUP BY column_name(s).
HAVING condition;
```

Que3. What is the Syntax of GROUPBY clause?

```
SELECT column_name(s).
FROM table_name.
WHERE condition.
GROUP BY column_name.(s)
```

Que4. Difference between HAVING AND WHERE clauses.

WHERE clause cannot be used with aggregates but having clause can.

Viva- Questions:

Que1. What is a JOIN Operation?

A Join operation is used to combine rows from two or more tables based on a related column b/w them.

Que2. Difference between JOIN and PRODUCT operation?

In Join operation we have keys which will be able to join rows.

But in PRODUCT we have rows with each record in first table matched each and every row of second table, no. of rows is fixed.

Que3. What are different types of JOIN Operations?

(INNER) JOIN
LEFT (OUTER) JOIN
RIGHT (OUTER) JOIN.
FULL (OUTER) JOIN.

Que4. Difference between RIGHT OUTER and LEFT OUTER join?

LEFT OUTER JOIN returns all records from left table & matched records from right table. Whereas RIGHT OUTER JOIN returns all records from right table matched with records from from left table.

Que5. What is an INNER join?

INNER JOIN returns records that have matching values in both tables.

Viva-Questions

Que1. What is a view?

A View is a virtual table based on the result-set of an SQL statement. It contains rows & columns

Que2. How we create a view?

```
CREATE VIEW view_name AS
SELECT column1, column2.
FROM table_name
WHERE condition.
```

Que3. What is the purpose of creating a VIEW?

By creating a view we have many different perspectives of same table.

Que4. What is the syntax of creating a VIEW?

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Que5.How many VIEWS can be created for a table?

You can create as many views for a table as many as you want.

Kasuma