DATABASE MANAGEMENT SYSTEM

A database is an organized collection of data, generally stored and accessed electronically from a computer system.

A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structures and file structure. It also defines rules to validate and manipulate this data.

A DBMS retrieves users of framing programs for data maintanance. Fourth-generation query languages, such as SQL, are used along with the DBMS package to interact with a database. Some other DBMS example include:

* My SQL
* Oracle
* SQL Server
* FoxPro

A database management system recieves instruction from a database administrator (DBA) and accordingly instructs the system to make the necessary changes. These commands can be to load, retrieve or modify existing data from the system

INTRODUCTION TO SQL

SQL is a database computer language designed for the retrieval and management of data in a relational database. SQL stands for Structured Query Language. SQL is a language to operate databases; it includes database creation, deletion, fetching rows, modifying rows, etc. SQL is an ANSI standard language, but there are many different versions of the SQL language. SQL is the standard language for relational Database System. All RDBS like MySQL, MS Access, Oracle, Sybase, Informix and SQL server use SQL as their standard database language. SQL is widely popular because it offers the following advantages.

* Allow users to access data in RDBS.
* Allow users to describe the data.
* Allow users to create and drop databases and tables.

DATA TYPES IN SQL

* Exact Numeric Data Types

- bigint
- int
- smallint
- tinyint
- bit
- decimal
- numeric
- money
- smallmoney

* Approximate Numeric Data Types

- float
- real

* Date and Time Data Types

- datetime
- smalldatetime
- date
- time

* Character Strings Data Types

- char
- varchar
- varchar (max)
- text

* Binary Data Types

- binary
- varbinary
- varbinary (max)
- image

CREATE TABLE

The create table statement is used to create a new table in a database.

Syntax - create table table_name(column1 datatype, column2 datatype,
- - - - - - - - - - - - - -
columnn datatype);

Example - create table Persons (PersonID int,
LastName varchar (255),
FirstName varchar(255),
Address varchar (255),
city varchar (255));

INSERT INTO

The insert into statement is used to insert new records in a table.

Syntax 1 - The first way specifies both the column names and the values to be inserted

- insert into table_name (column1, column2, ...)
values (value1, value2, value3, ...);

Syntax 2 - If we are adding values for all the columns in the table.

- insert into table_name
values (value1, value2, value3, ...);

Create a table and then do alter, add modify, and rename a table

Create a table and then inserting the value as follows

id	name	dname	doj
101	Ayush	CSE	14-Mar-2015
102	___	ME	___
103	Amit	___	15-Mar-2015
104	___	___	17-Mar-2015
105	___	___	___

Syntax:-

```
create table nempr(id number(15), name
varchar(10), dname varchar(10), doj date);
table created
insert into nempr(id, name, dname, doj)
values(101, 'Ayush', 'CSE', '14-Mar-2015');
1 row created
insert into nempr(id, dname);
values(102, ME);
1 row created
insert into nempr(id, name, doj);
values(103, 'Amit', '15-Mar-2015');
1 row created
```

```
insert into nempt( id, doj);
values( 104, '17-Mar-2015');
1 row created
insert into nempt(id)
values (105);
1 row created
select * from nempt
```

id	name	dname	doj
101	Ayush	CSE	14-Mar-2015
102		ME	
103	Amit		15-Mar-2015
104			17-Mar-2015
105			

Alter table :-

```
alter table nempt add city varchar(10);
```

Table altered

```
desc nempt
```

Name	NULL?	Type
ID		NUMBER(5)
NAME		VARCHAR2(10)
DNAME		VARCHAR2(10)
DOJ		DATE
CITY		VARCHAR2(10)

Alter the datatype of variable

```
alter table nempt modify name varchar(12);
table altered
```

Alter the table name of the column

```
alter table nempt rename column dname
to dept;
table altered
```

Alter the name of the table

```
rename nempt to itemp;
table altered.
select * from itemp;
```

ID	NAME	DEPT	DOJ	CITY
101	Ayush	CSE	14-Mar-2015	
102		ME		
103	Amit		15-Mar-2015	
104			17-Mar-2015	
105				

UPDATE

```
update itemp set doj = '29-Mar-2015'
where id = 102
1 row updated
update itemp set city = 'Lucknow';
5 rows updated
update itemp set city = 'kanpur' where
  dept = 'CSE' or dept = 'ME';
2 rows updated
```

update itemp set doj = '' where id = 104;
1 rows updated

select * from itemp

ID	NAME	DEPT	DOJ	CITY
101	Ayush	CSE	14-Mar-2015	Kanpur
102		ME		Kanpur
103	Amit		15-Mar-2015	Lucknow
104				Lucknow
105				Lucknow

Write SQL queries for arithmatic and logical operators.

```
select 4+5 sum_of_two_no from dual;

SUM_OF_TWO_NO
-------------
            9
```

To select all tables from login

```
select * from tab;

ITEMP
TEMPR
HOSPITAL
   :
   :
EMPLOYEE
```

EMP_NO	NAME	DOJ	SALARY
11	Babulal	0	5000
12	Ramlal	0	6300

```
select name, salary, salary+500 new_sal from
employee
```

NAME	SALARY	NEW_SAL
Babulal	5000	5500
Ramlal	6300	6800

These columns are only representing the name at the table only it does not replace the actual data.

Logical Operators

In this three logical operators are used

- → AND
- → OR
- → NOT

If we chuck both the conditions then apply AND.

If we chuck one condition then apply OR.

```
select name from employee where salary > 4000
or salary < 10000;
```

```
NAME
---
Babulal
Ramlal
```

```
select name from employee where salary > 4000
and salary < 6000;
```

```
NAME
------
Babulal
```

```
select name from employee where salary > 4000 and
not (salary = 5000);
```

```
NAME
---
Babulal
Ramlal
```

```
select sum(salary) from employee
```

```
SUM(SALARY)
---
      11300
```

Write SQL queries for Union, Intersect, Minus.

Union

```
select ename from emp where sal>5000
Union
select ename from emp where deptno=10;
```

ename
Raj
Kaushik
Vishwas
Bhanu

Intersect

```
Select ename from emp where sal>5000
Intersect
select ename from emp where deptno=10;
```

ename
Raj
Kaushik
Bhanu

Minus

```
Select ename from emp where sal>5000
Minus
select ename from emp where deptno=10;
```

ename
Raj
Kaushik

Range Searching => This is used for searching within a range.

select ename from emp where sal between 10000 and 20000;

ename ...

Bhanu
Susheel
Tanveer
Sudhir

Pattern matching => In this the words are matched by matching the exact with their database.

select ename, sal from emp where ename in ('KING', 'SMITH');

select ename, sal from emp where ename not in ('KING', 'SMITH');

select ename from emp where ename like 'A%';

select ename from emp where ename like 'K_N%';

select ename from emp where ename like '%H';

select ename from emp where ename like '%N_';