SQL

In Relational database Model all the information is stored on Tables, these tables are divided into rows and columns. A collection on related tables are called DATABASE. A named table in a database is called RELATION in Relational Data Model. Row in a table is called TUPLES, and column of a Table are called ATTRIBUTE. No. of rows in a table is called DEGREE of the table and no. of columns in a tables in called CARDINALITY.

Primary Key : An attribute or a group of attribute which can distinguish a row uniquely in a table is Called Primary key/Key Field/Key attribute.
Candidate key : The attributes which are capable to act as a primary key is known as candidate key.
Alternate key : An attribute which can act as a primary key in place of primary key as called alternate key.
Foreign Key : An attribute in it's present table whose values are derived from some other table, is called foreign key in the present table.

SQL stand for structured query language, a language to manipulate database. Now a days It is a universal database language to create , manipulate database across plate forms.

The data types supported by the standard SQL are as follows

S.No	Data Type	Meaning	Example
1	Char	It can store information containing alphabets and numeric or alphanumeric	**Char (30)** can store "ram" , "ramji007" or "80- b surya Nagar
2	VarChar2	It can store information containing alphabets and numeric or alphanumeric. But these data types when used can increase it 's size when required	**Varchar2 (30)** can store "ram" , "ramji007" or "80- b surya Nagar
3	Number	It can store numeric values which can have fractional part or without fractional part	(1) **Number(2)** mean it can store a numeric values between 00 and 99 (2) **number(8,2)** means it can store values whose after decimal points values can go upto two places
4	Date	It is used to store date type information	**Date** can store any valid date

Operator : SQL supports different types of operators some of them is as follows

1. **Arithmetic operator** Example are : + , - , * and /
2. **Logical operator** examples are : > , < , =, >= , <=, != (not equal to)
3. **Relational Operator** Examples area : AND , OR , NOT

DDL : DDL stand for data definition language . it is basically responsible to define the structure of any database table. The commands falls under this category is as follows

1. Create Table
2. Alter Table
3. Drop Table

DML : DML stand for data manipulation language. This is basically responsible for managing the records(row/tuple) of any table. The main command falls under this category are

1. Insert
2. Delete
3. Update
4. Select
5. Create view

6. Drop View

Create Table : This DDL command is used to create a new table into any existing database . The syntax of this command is as follows

Syntax

```
Create table <tblname> ( column_name   datatype   size  constraints ,
                         column_name   datatype   size  constraints ,
                         column_name   datatype   size  constraints ,
                         ----------------
                         ----------------
                         );
```

Example :

```
Sql>Create table student ( admno number(4) ,      roll number(2),
                           Name   char(30),       fname  char(30),
                           DOB    date
                           Address char(80)
                           );
```

To see the structure of the above defined table is as follows

```
Sql> desc student ;
```

ALTER TABLE : This DDL command is used to add / modify a column in any existing table of a database The syntax of this command is as follows

Syntax

```
Alter table <tblname>  ADD/MODIFY/DROP (  column_name   datatype   size  constraints ,
                         column_name   datatype   size  constraints ,
                         column_name   datatype   size  constraints ,
                         ----------------
                         ----------------
                         );
```

Example

Task : To add a new column " Phone " having datatype char and size 12 in a table student

Sql > alter table student ADD (phone char (12)) ;

Task : To change the datatype of admno into char and increase the size as 5

```
Sql > alter table student MODIFY (  admno char(5));
```

Task : Delete a column DOB from the table student

```
Sql > alter table student drop DOB;
```

Drop Table ; This DDL command is used to remove a table from any existing database. This syntax of this command is as follows

Syntax

```
Drop table < table name>
```

Example

```
Sql> drop table student ;
```

INSERT COMMAND : This DML command is used to add a new row(record/ tuple) in any existing database table . The syntax is as follows

Syntax

```
insert into < table name > values ( value1, value2, value 3,.....value N );
```

Example

Task : Add a new row in a table student (which we have created earlier)

Sql > Insert into student values ('1234A','12,' 'joshin gulati',''abcd', '9-apr-2008','b-152 surya

nagar','0120-2626132'

Note : Always put char type values and date type values in single inverted comma

SELECT COMMAND : This DML command is used to retrieve information from any table (s) to display/ list/ report on the screen, but this command can not any how update/ modify any table. The syntax of this command is as follows

Syntax

Select [* / column list] from < table name>
[where < condition >]
[order by < col name..> [asc/desc]]
[group by < col name>]

Examples

Task : To display all the records of table EMP

*Sql > select * from EMP;*

Task : To display only the employee no and employee name

Sql > select empno , ename from EMP;

Task : to display employee name in first column and employee number on second column

Sql> select ename, empno from EMP;

Where clause : it is used to restrict the no no of rows from being displayed on the screen

Task : display all the employees whose salary is greater than 1500

Sql > select * from emp where sal > 1500

Task : display all the employees information whose salary is between 1500 and 2500

Sql > select * from emp where sal > =500 and sal <=2500

Task : Display all the information of clerks

Sql > select * from emp where job ='CLERK';

Task :display the information of all the employees whose job is 'salesman ' or 'CLEAK'

Sql> select * from emp where job ='SALESMAN ' OR job ='CLERK'

Task : Display the information of all those employee whose job is not president

Sql> select * from emp where job !='PRESIDENT'

OR

Sql> select * from emp where NOT job='PRESIDENT'

Calculated field :

Task : Display employee number, name and salary earned by each employee in a year from the table EMP (Suppose salary is paid in every month)

Sql > select empno,ename , salary *12 from EMP;

NOTE : In the above example salary * 12 is a calculated field.

Comparing with NULL values

Task : Display employees information who earns NULL commission

Sql > select * from EMP
Where sal is NULL

NOTE : Do not use equal sign to compare NULL values.

Like Clause -----Pattern Matching : This clause is used to display only those records which contains a single or multiple char from a given set of values .

Task : Display all the names starting with alphabet 'A' from table EMP.

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

Task : display all the name starting with alphabet 'A' and must contains only 5 letters

Sql> select ename from emp where ename like 'A_ _ _ _';

Note : % --- > replace all the remaining char(s)
_ (under Score) ----- > Replace only a single char at a time

Removing Duplicate Rows (Distinct Keyword)

Task : Display Different jobs available in Table EMP
Sql > select distinct (job) from EMP;

Order By clause : This select clause is used to display the retrieved data in an arrange format

Task : display employee table information according to the name in asceding order from the table EMP
Sql > select * from emp order by name asc
Task : Display only employee names in descending order from the table EMP
Sql > display ename from emp order by ename desc
Task : Dispay the information of only clerk in ascending order according to their name from table EMP
Sql > select ename from EMP
Where job ='clerk'
Order by ename asc;

NOTE : Asc is not compulsory but desc is compulsory

Grouping Function: These functions are operative on the whole table, so the result of these functions are based on the whole table. These functions can not be applied on a single record/row

1. **AVG( ) : try to find out average of given data--------Applicable on numeric only**
2. **Sum( ) : Try to find out sum of given data ----------Applicable on numeric only**
3. **Count :** count total no of rows of a table ……**-Applicable on all type of data**
4. **Min : find out the minimum numeric values …….-Applicable on numeric only**
5. **Max : : find out the maxi numeric values…… …….-Applicable on numeric only**

Example :

Task : find out Maximum salary paid
Sql> select max(sal) from EMp;
Task : find out minimum salary paid
Sql > select min(sal) from emp;
Task : find out average salary paid to all employees
Sql > select avg(sal) from EMP;
Task : find out total salary paid to all employees
Sql > select sum (sal) from EMp;
Task : find out total no of employee
Sql > select count(*) from emp;
Task : find out total no of different types of job from the table EMp
Sql > Select count(distinct job) from emp;

Group By Clause Example

Task : Find out the total no of employee in each jobs from EMP table
Sql > select job , count(job) from EMP

Group by job;

Task : Find out average salary of clerks from table EMP

Sql > select job , avg(sal) from EMP
Where job ='CLERK;
Group By job

Joins

Joins : A join is a query that combines rows from two or more than two tables. The function of combining Data from multiple tables is called joining. In a join query more than one table is listed in FROM Clause.

Example

Sql > Select * from EMP, DEPT;

NOTE : when two tables are joined without any condition then the Numbers of columns in the resultant query is the addition of first table and second tables, and the number of row are the multiplication of both tables row --- This is also called **Cartesian product** of two tables)

EQUI JOIN

Sql > select * from emp , dept
Where emp.deptno = dept. deptno
(The above two tables are joined by some condition ---This is called equi Join – but one column appear twice)

Natural JOIN

Sql > select empNp, ename , emp.deptno from emp , dept
Where emp.deptno = dept. deptno
(In this case duplicate column appears only once)

Update : This DML command is used to change/modify the record(s) of any table. The syntax is as follows

Syntax

Update < tblname>
Set column Name = value
[where < condition>]

Example

1. Update emp
Set sal = sal + sal * 0.05; (Give an increment of 5 % to each employee)
2. update emp
set sal = sal+ sal * 0.05
where job ='PRESIDENT'; (Give an increment of 5% to president only)

Delete : This DML command is used to delete a row/ tuple(s) from a table. The Syntax is as follows

Syntax

DELETE from <tblname >
[where < condition >]

Example

1. Delete from emp ; (delete all the records leaving it's structure intact)
2. delete from emp where sal >=3000 ;

Create View : This Command is used to create a new View in any existing database. The syntax is as Follows

Syntax

Create view < viewname>
As

Select command

Example

1. create view abc
   as
   select empno,job,sal from emp;

Drop View : This command is used to drop any existing view from the database. The syntax is as follows

Syntax

Drop View < viewname>

Example

Drop view abc;