SQL

`CREATE` is the DDL(data definition language) commands used for the creation of the tables in a database

CREATE TABLE

- ***Create*** command is used to create a table in the database with the structure specified by the user
- This structure includes the ***number of columns to be present in the table and the data type of the column, size of data,***

```
create table emp(
empno number(4,0),
ename varchar2(10),
job varchar2(9),
mgr number(4,0),
hiredate date,
sal number(7,2),
deptno number(2,0)
PRIMARY KEY (ID)
);
```

- **Table name**:***emp***
- **Column names**: In the above table that we have created have ***7 columns namely empno, ename, job, mgr,hiredate, sal,deptno***
- **Data types:** What ***type of data should be entered for each column value***. for example, we have used ***number*** data type for the column empno , which means you must enter numerical values only while inserting data for emp column
- ***Size***: ***Specifies the length of the value that is inserted***, for example, we have used size 10 for ename as varchar2(10), which
  means the maximum number of characters that can be entered for the ename column is 10.

ALTER TABLE (ADD, DROP, MODIFY, RENAME)

In this article, we will learn about ALTER in DBMS.

- Alter command in SQL is used to make modifications to the columns in the existing table

- It is used to add columns, delete columns, drop constraints, renaming the columns, changing the data type and data type size of the column existing in the table.

ALTER Command: ADD

- ADD command is ***used to add one or more new columns to the existing database tables***
- The ***newly added columns will be empty*** and data can be entered by using insert command

```
// Adding a single column
ALTER TABLE student ADD(address VARCHAR(100));


//Adding multiple columns
ALTER TABLE studentADD (
father name VARCHAR(60),
mother name VARCHAR(60),
DOB DATE);

//Adding constraints
ALTER TABLE Persons
ADD PRIMARY KEY (ID);

//Adding a column with a default value
ALTER TABLE student ADD(
gender char(1) default 'M'
);
```

ALTER command: MODIFY

By using the modify command, ***we can change the data type of the existing column or the size of the data type*** of the existing column

```
//Modifying a single column
ALTER TABLE studentMODIFY(
address varchar(75));

//Modifying multiple columns
ALTER TABLE student MODIFY(
address varchar(75),dob date);
```

Alter command: Rename

Rename command in combination alter is ***used to rename the existing column name of a table***

```
ALTER TABLE studentRENAME
address to location;
```

ALTER command: DROP

- Alter command is ***used to delete one or more existing columns*** present in the table
- We ***can also drop more than one column*** by separating the column names with commas

```
//Modifying a single column
ALTER TABLE studentDROP(
address);

//Modifying multiple columns
ALTER TABLE student DROP(
   Caste, religion);

//Dropping constraints
ALTER TABLE emp
DROP CONSTRAINT PK_id;
```

DROP/TRUNCATE/RENAME

1. DROP command

- Drop command delete the table existence completely i.e ***drop statement destroys the existing database object*** of that particular table, index or view
- After dropping a table if you try to use the table then compiler shows an error as " ***table or view does not exist***"

`drop table emp;`

2. TRUNCATE command

- The truncate command will make the table empty i.e all the table data will be deleted but the structure and database object is still alive ***and the table can be reused normally***
- The truncate command logically nothing but using delete command for deleting the records in the table without specifying the where condition i.e all the rows get deleted in that case

`truncate table emp;`

3. RENAME command

The rename command is used to change the existing table name and give a new name to the table.

`rename emp to Employees`

SELECT Query in DBMS

- A `SELECT` query is used to retrieve data(records) from the table
- We can retrieve complete table data, or specific records by specifying conditions using the WHERE clause.

`SELECT * from student;`

```
SELECT * from
student
where branch='computers';


SELECT stuid,sname,score,score+5
from student
where score>80;
```

INSERT Query in DBMS

- `INSERT` is a widely used data manipulation language(DML)command for adding new data to the existing database table
- Insert command is used to add one or more rows of data to the database table with specified column values

```
insert intostudent
 values(66'trishaank','computers','24-07-1998',92);
```

```
insert into student(stuid,sname,branch) values (82,'Srinivas','Computers');

insert into student values(&stuid,'&sname','&branch','&dob',&score);

INSERT INTO sample2 SELECT * FROM sample1;
```

WHERE Query in DBMS

- WHERE clause is used to specify a condition while retrieving and updating data from the database table and display only those records in the table for which the condition specified in the where clause becomes true
- The ***WHERE*** clause is most commonly used with a select, update and delete statements
-

Operator	Description
=	Equal to
!=	Not Equal to
<	Less than
>	Greater than
<=	Less than or Equal to
>=	Greater than or Equal to
BETWEEN	Between a specified range of values
LIKE	This is used to search for a pattern in value.
IN	In a given set of values

```
select * from emp
where job='MANAGER';

select ename,job,sal
from emp
where sal>2900;

select ename,sal,job,deptno from emp
where job != 'MANAGER';

select ename,sal,job,deptno from emp
where SAL between 1000 AND 2500;

delete from emp
where empno=7934;
```

AND & OR in DBMS

- Sometimes user requires ***more than one condition for filtering the data***, this purpose is served by AND / OR clauses
- The `AND` and `OR` operators are used with the where clause for precise filtration of data from the database tables by combining more than one condition along with select, update and delete queries

```
select * from emp
where job='MANAGER' ANDsal>2500;


select * from emp
where JOB='ANALYST' OR JOB='MANAGER';

select * from emp
where (sal>1500 OR job='MANAGER') AND (deptno=10 OR deptno=30);
```

UPDATE Query In DBMS

- The UPDATE statement is used to modify or change the data of the existing table in the database
- We can update a single column as well as multiple columns as per our requirement

```
UPDATEempSET comm=200
WHERE sal<3000;

update emp set sal=sal+500 ,comm=1000
where job='ANALYST';

update emp
set sal=500;
```

DELETE Query In DBMS

DELETE statement is used to delete single or multiple records present in the existing database table based on a specific condition

```
deletefrom emp
where empno=7698;

delete from emp
```

```
where sal>2500  and job!=  'PRESIDENT';

delete from emp;
```

LIKE IN DBMS

- If you want to ***search all employees starting with letter P or names of all products which which consists of exactly 4 letter***s etc
- LIKE in DBMS operator used to search specified pattern in the data and retrieve the record when there is a pattern match as required

Wildcard operators

% : ***Percent(%)***represents 0,1 or multiple characters
_   : ***Underscore(_)*** is used to match exactly a single character

This wild card operator is used in conjunction with where clause and filter the records based on specified patterns as required.

Different ways that we can use this LIKE clause

The following are the rules for pattern matching with the LIKE Clause:

PATTERN	MEANING
‘r%’	Matches strings which start with ‘r’
‘%r’	Matches strings with end with ‘r’
‘r%t’	Matches strings which contain the start with ‘r’ and end with ‘t’.
‘%tri%’	Matches strings which contain the substring ‘tri’ in them at any position.
‘_tri%’	Matches strings which contain the substring ‘tri’ in them at the second position.
‘_r%’	Matches strings which contain ‘r’ at the second position.
‘r_%_%’	Matches strings which start with ‘r’ and contain at least 2 more characters.

```
select ename  from emp
where enamelike 'M%';

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

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

select ename from
```

```
emp where ename like '___M%';

select ename,hiredate from emp
where hiredate LIKE '%DEC%';

select ename from emp
where ename like '____';

select ename from emp
where ename not like '%A%';
```

ORDER BY in DBMS

The `order by` clause is used to arrange the fetched data from the database table in ascending or descending order of data values based on one or more columns

- Sometimes the user may be interested in ***arranging the data in the table in some increasing or decreasing order of values***
- **Example**: If you want to display the details of all students based on descending order of their attendance or marks etc

```
select ename,sal from emp
ORDER BY salDESC;

select deptno,ename from emp
ORDER BY deptno;

select ename,job,sal,hiredate
from emp
order by hiredate desc;

select ename,sal,deptno from emp
ORDER BY deptno,sal DESC;

select ename,job,sal from emp
where job='MANAGER'
order by sal desc ;

select ename, sal, job
from emp
order by 2 DESC;

select ename ,sal*12 annsal ,job,hiredate
from emp
order by annsal desc;
```

GROUP BY IN DBMS

Group by clause in SQL used to arrange logically related data into groups with help of some functions i.e if a particular column has the same type of data in different rows then they can be organized this into a logical groups.

```
select deptno,count(*)
from emp
group bydeptno;

select deptno,max(sal)
from emp
group by deptno

select deptno,max(sal)
from emp
where deptno!=30
group by deptno;

select deptno,max(sal)
from emp
group by deptno
order by deptno;

SELECT ename , SUM(sal) FROM emp
GROUP BY ename
HAVING SUM(sal)>2000;
```

DISTINCT IN DBMS

- Inside table columns may contain many duplicate values and sometimes we require to list only unique values, this is done by using `DISTINCT` clause along with a select statement
- DISTINCT statement is used to return only unique values present in a column or combination of columns
- The DISTINCT clause is only ***for display purpose and will not affect the original database table***

`select` ***`DISTINCT`*** `job from emp;`

```
selectDISTINCT job,sal
from emp;

select DISTINCT ename, job
from emp;
```

ROW

Each row represents a complete record of the specific data item, each row stores different data with the same structure

```
SELECT empno,ename,job,sal,deptno FROM EMP

select rownum,empno,ename,job,sal,deptno
FROM EMP;

select rownum,empno,ename,job,sal,deptno
FROM EMP
where rownum<=3;
```