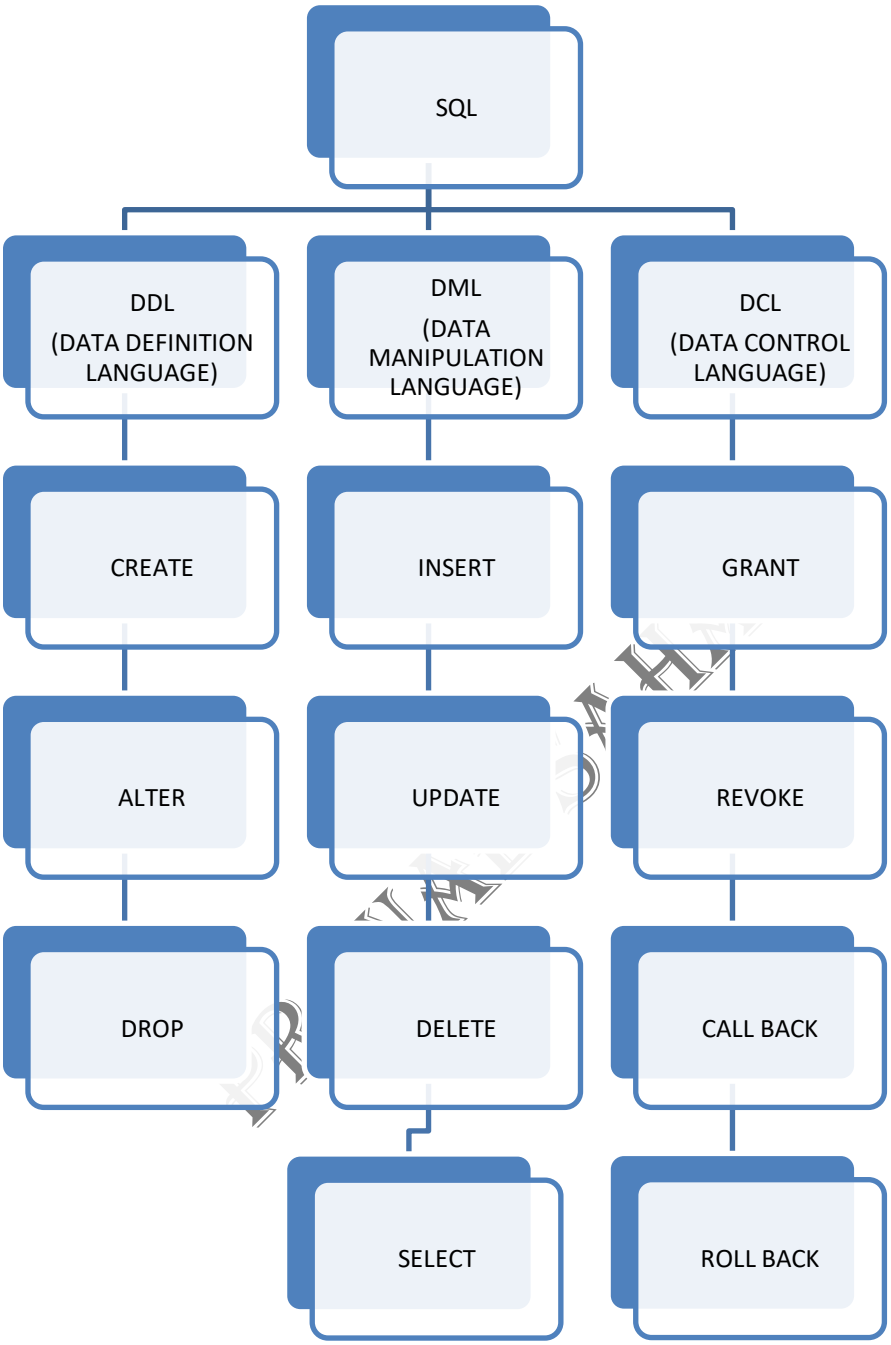
# SQL (Chapter -10)

## SQL(STRUCTURE QUERY LANGUAGE)




DBMS SOFTWARE → MS. ACCESS, ORACLE, SQL SERVER, DB2, FOXPRO, DBASE,MY SQL.

DATABASE → OBJECT

6 OBJECT → TABLE, QUERY, FORM, REPORT, MODULE, MACRO,

2003 OR LOWER VERSION MS. ACCESS EXTENSION = .mdb

(meta database)

2007 OR LOWER VERSION MS. ACCESS EXTENSION = .accdb

(access database)

| Student_Name | Roll_No | Math | Phy | Cs | Chem |
|---|---|---|---|---|---|
| Ankita | A1 | 100 | 90 | -- | 80 |

Primary Key = Unique + Not Null

DML →

INSERT → TO ENTER OR INSERT ANY DATA WITHIN THE TABLE OF THE DATABASE.

UPDATE → TO CHANGE OR EDIT OR UPDATE THE CONTENT OR COLUMN OF ANY DATABASE TABLE.

DELETE → TO DELETE A SPECIFIC ROW WITHIN A TABLE.

SELECT → TO FETCH OR RETRIEVE THE DATA FROM THE TABLE WITHOUT OPENING THE TABLE.

insert query syntax :-

insert into tablename values (fieldname1, filedname2,----)

keyword keyword

text column data entry = ' '

number column data entry = no inverted comma

eg. →

insert into student values('Ankita',1,15,25)

DDL→

ALTER → TO ADD NEW COLUMN WITHIN THE TABLE

SYNTAX :-

ALTER TABLE TABLENAME ADD COLUMNNAME DATA TYPE

KEYWORD KEYWORD

eg→

alter table student add chem number

UPDATE →

SYNTAX→

1) UPDATE TABLENAME SET FORMULA

KEYWORD KEYWORD

2) UPDATE TABLENAME SET FORMULA WHERE CONDITION

1) EG→ UPDATE STUDENT SET total=math+phy+cs+chem.

2) eg→ update student set stud_name='ankita mondal' where stud_name='ankita'

>0 & <40 → FAIL

>40 & <60 →C

>60 & <80 →B

>80 & <90 →A

>90 →O

select syntax :-

1) select column names from tablename

keyword keyword

all column = *

2) select column names from tablename where condition

_MS_Access_DB

>0 &&<120 --Fail

>120 && <200 -- C

>200 && <280 --B

>280 && <320 -- A

>320 && <360 A+

>360 AA

>90 City = 'Bihar'

>80 && <90 city="Bangalore'

>70&& <80 city='Delhi'

>50 && <70 city='Mumbai'

>30 && <50 city='kolkata'

<30 city='chennai'

| Emp_id | Emp_name | Basic_salry | Da | Ta | Hra | Gross_salary | PF | NET SALARY |
|---|---|---|---|---|---|---|---|---|

By using Query Table Create :- DDL

```
Create table emp
(
    Emp_id text,
    Emp_name text,
    Basic_salary number,
    DA number,
    TA number,
    HRA number,
    Gross_Salary number
);
```

INSERT QUERY :-

INSERT INTO TABLE NAME VALUES(FIELD1,FIELD2,----)

EG→

INSERT INTO EMP VALUES('E002','ABINASH',20000)

UPDATE SYNTAX :-

1) UPDATE TABLENAME SET FORMULA

2) UPDATE TABLENAME SET FORMULA WHERE CONDITION

DA = 7.5% ON BASIC SALARY

TA = 10% ON BASIC SALARY

HRA = 15% ON BASIC SALARY

GROSS=(BASIC+TA+HRA-DA)

PF=25% ON BASIC SALARY

net salary =gross salary -pf

ALTER →

ALTER TABLE TABLENAME ADD COLUMNNAME DATATYPE

1) CHANGE THE NAME OF THOSE EMPLOYEE WHOSE IS SNEHA AND ID IS E004

SELECT SYNTAX :-

1) SELECT COLUMNNAME FROM TABLE NAME

2) SELECT COLUMNNAME FROM TABLE NAME WHERE CONDITION



* = ALL COLUMN

<=10000 PEON

>10000 TO <=20000 CLERK

>20000 TO <=40000 OFFICE SUPERINTENDENT

>40000 TO <=50000 EXECUTIVE

>50000 TO <=60000 SR. EXECUTIVE

>60000 TO <=80000 AST. BRANCH MANAGER

>=80000 TO <=100000 BRANCH MANAGER

>100000 DGM =DEPUTY GENERAL MANAGER

RANGE >= <= (BETWEEN)

>= GREATER THAN

<= LESS THAN

<> NOT EQUAL

IN

NOT IN

LIKE

1) SHOW/DISPLAY THE RECORDS OF THOSE EMPLOYEE WHOSE POST IS “CLERK”.

→ RECORDS = ALL COLUMN

2) SHOW/DISPLAY THE EMP_NAME,NET_SALARY,PF,DESIGNATION OF THOSE EMPLOYEE EXCEPT “CLERK”.

3) SHOW THE RECORDS OF THOSE EMPLOYEE WHOSE EMP_ID IS BETWEEN E004 AND E007

4) ADD A NEW COLUMN CITY

5) UPDATE THE CITY AS BIHAR WHOSE DESIGNATION IS PEON

6) SHOW THE NAME, DESIGNATION,CITY OF THOSE EMPLOYEE WHOSE CITY IS KOLKATA / WHO ARE LIVE IN KOLKATA.

7) SHOW THE NAME, NET_SALARY,DESIGNATION,CITY OF THOSE EMPLOYEE WHOSE CITY IS EITHER KOLKATA OR MUMBAI / WHO ARE LIVE IN KOLKATA OR MUMBAI. → or, in

8) SHOW THE RECORDS OF THOSE EMPLOYEE WHOSE CITY Neither KOLKATA NOR MUMBAI / WHO ARE NOT LIVE IN KOLKATA OR MUMBAI. → <> , NOT IN

AND → SAME ROW CHECKING(IF ALL THE CONDITION IS TRUE THEN WE WILL GET THE OUTPUT)

OR → DIFFERENT ROW CHECKING(IF ANY ONE OF THE CONDITION IS TRUE THEN WE WILL GET THE OUTPUT)

LIKE operator → It is an operator which is used to check a particular letter or more than one letter within a given string(text).

WILD CARD CHARACTER

_(UNDERSCORE) , % → THEORY

_ → to check a single character.

% →To check multiple character.

1) show the list of employee whose name start with S?

2) show the name,netsalary,designation,city of those employee whose name ends with A?

3) show the list of employee whose name ends with AN?

4) show the list of employee whose name start with S & ends with 4 letters?

5) show the list of employee whose name ends with AN **& begin with 3 letter** / **contains 5 letter**?

6) show the list employee whose name contains 'O'.

*O*

→ O*

→*O

→*O*

7) Show the records of those employee whose name does not contain/does not belong with A

8) Show the records of those employee whose name does not start with A.

9) show the list of employee whose name does not end with AN **& begin with 3 letter** /does not **contain 5 letter**?

Data sorting

→ ascending = choto theke boro
→ descending = boro theke choto
ascending = asc(keyword) by default sorted mode is ascending.
descending =desc(keyword)
order by (keyword) → is a keyword which is used to sort the data or any particular column in ascending or descending order.
order by can be used within the condition or without the condition.
10) show all records of employee in ascending order/ in order of Emp_Name.
11) show all records of employee in descending order of Emp_Name.
12) Show all records of employee in ascending order of emp_Name whose salary is greater than 50000.

13) Show all records of Employee whose Name start with 'A' in descending order of Emp_Name

14) show all records of Employee whose Name starts with S and salary greater than 50000 in ascending order of salary.

1) show all the records of employee in ascending order of designation

2) show all the records of employee who are either live in Mumbai or live in Kolkata in descending order of designation

3) show the all records of employee whose name contains 'O' & begins with one letter.

4) show the all records of employee whose name contains 'O'(not end with O or not begins with O)

5) SHOW EMP_NAME, NET_SALARY,DESIGNATION,DEPT_NAME & ADDRESS OF EMPLOYEE.

**(HERE EMP_ID OF EMP TABLE IS PK & EMP_ID OF DEPT TABLE IS TREATED AS FK)**

6) show emp_name,basic_salary,designation,city,d_name,emp_address,ph_no whose basic salary between 20k to 40k.

7) show emp_name,basic_salary,designation,city,d_name,emp_address,ph_no whose basic salary between 20k to 40k and who are live in either delhi or mumbai.

NOTE :- IF THERE IS NO CHECKING MENTIONED IN THE QUESTION FOR FETCHING RECORDS FROM 2 OR MORE TABLE, THEN PRIMARY KEY & FOREIGN KEY CHECK HAS TO BE THERE

'AS' KEYWORD IS USED TO TEMPORARILY CHANGE THE COLUMN NAME OR TABLE NAME

**DOCTOR & PATIENT**

1) SHOW DOCTOR_NAME, PATIENT_NAME, ADMIT_DATE, DISCHARGE_DATE,TOTAL_BILL FROM BOTH THE TABLE.

2) SHOW TOTAL PATIENT ADMITTED IN THE HOSPITAL.

3) SHOW THE PATIENT_NAME,BILL_AMOUNT FROM THOSE PATIENT WHOSE DOCTOR IS 'SAYANTAN' , 'SAYANTA'

4) show no. of patient under each doctor.

5) show no. of patient admitted in each date under each doctor.

6) show the highest bill among all patient from 26/09/20 to 30/09/20.

6) show the highest bill among all patient from 26/09/20 to 30/09/20 of each patient.

7) show no. of patient under each doctor but no_of_patient greater than or equal to 2.

8) show the lowest bill among all patient from 26/09/20 to 30/09/20 .

1) Show the Student_Name, Teacher_name,City,Total,Teacher_Address & salary from both the table.

2) Show the Student_Name,T_Name,Salary,City from both the table whose name start with 'S' and end with 'i'

3) Show the Student_Name,T_Name,Salary,City from both the table

whose total is 150 or more

4) Show the Student_Name,T_Name,Salary,City from both the table

whose total is 150 or more and ascending order of the teacher salary.

5) Show the Student_Name,T_Name,city from both the table whose city is either Chennai or delhi and whose teacher name starts with 'S'

in descending order of there teacher name.

Aggregate Function :-

1) max(), min(), avg(),count(),sum(),distinct

group by

Note :- Aggregate func. can't work after the where clause.

6) show the highest marks on Math form student table

7) show the lowest marks on Math form student table

Note : - beside the aggregate func. we can't include other column name

8) show the average marks on Math from student table

9) show total student in the student table.

1) show total no. student who are getting 80+ marks in chemistry.

2) show total no. student who are getting 80+ marks in chemistry and live in either Mumbai or Chennai.

3) show distinct city from student.

4) show distinct student name from student table.

5) show & count the total no. of distinct city from student table.

6) calculate total marks in chemistry.

7) calculate total marks in chemistry whose marks in chemistry is greater than 80.

8) calculate average in chemistry.

9) calculate average in chemistry whose marks in chemistry is greater than 80.

10) calculate average in math whose marks in chemistry is greater than 80.

11) Find out the Highest Marks in Chemistry.

12) Find out the Highest Marks & Lowest Marks in Chemistry.

13) count each individual city from student table.

14) count each individual student from student table.

1) COUNT HOW MANY STUDENTS ARE PRESENT IN THE "KOLKATA".

2) SHOW & COUNT HOW MANY STUDENTs ARE PRESENT WITHIN EACH INDIVIDUAL CITY.

3) SHOW & COUNT HOW MANY STUDENT with same/each name is present WITHIN student table.

4) count how many students are present within each class.

5) count how many students are present within each class and Sec.

6) count how many students are present within each class,Sec who are in class XI.

7) count how many students are present within each class,Sec who are in class XI or xii or b.tech.

8) count no. of student within each class & sec whose no. of student is greater than 2.

9) count no. of student within each s_name,class whose chem. marks is highest.

10) average of chem. is half yearly and annual exm.

11) count s_name,chem,exm within each s_name,chem. marks,exm whose avg in chem. for the year 2019 is >60

NOTE: - GROUP BY CAN BE USED AFTER WHERE CLAUSE.

BUT AFTER GROUP BY NO WHERE CLAUSE CAN BE USE.

note :- after where clause no aggregate func. can be use.

but, if we have to use aggregate func. after group by/ after group by we have to check any condition then "having" clause is use.

after "HAVING" clause we have use aggregate func. as a checking.