SQL

data: - a piece of useful information. / collection of raw information
database: - database is place where data is stored.

Processor

-to store the data we need to use database
-to communicate with database, we need a manager.
-that manager is DBMS (database management system)
- to communicate with DBMS we need a language, that language is SQL.
- by using SQL language ,we write queries.
- with the help of query, we can request the dbms to search for the data.
- once the query is submitted to dbms, it will start processing the query.
- the dbms will go inside database and it will search of the data inside database, if the data we are searching is it present inside database, dbms will take that data and display that data as output for the user.

Note :
SQL – structured query language . perilously called structured English query language.

Day 2

What is DBMS :-
-Is a program that stores retrieves and modifies data in the database on request.
- study of different techniques of design, development and maintenance of the database.

DBMS and it’s types:-

-One of the job of dbms is only storing the data but also arranging the data.
-To arrange the data we are having different types of dbms.
1.HDBMS

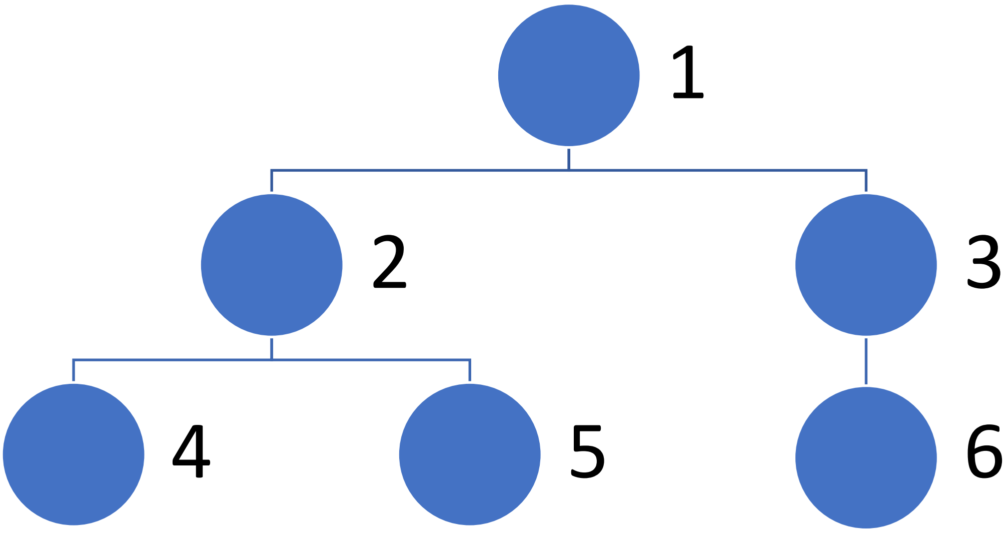
2. NDBMS

3.RDBMS

4.OBJECT ORIENTED DBMS

HDBMS (Hierarchal database management system) :-

in hdbms the data is going to be stored in hierarchal manner (tree type structure) .



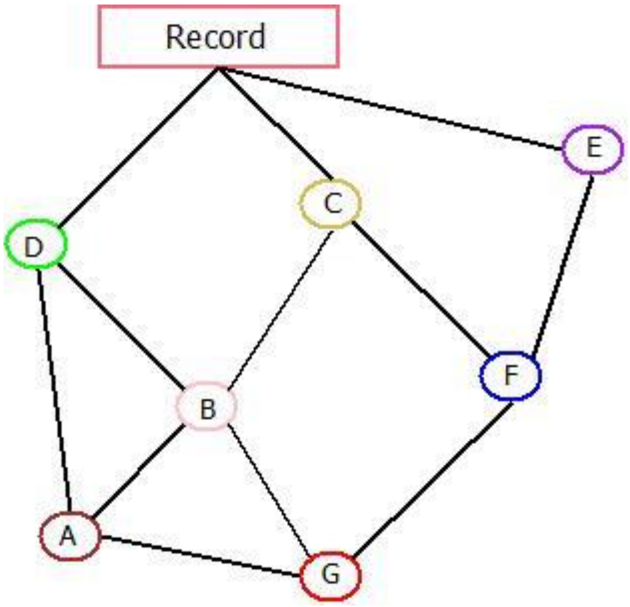
Day 3

Advantages (hdbms)	Disadvantages (hdbms)
If the data you are searching if it is present in first node then the time consumed for searching the data is less	If the data you are searching if it is present in last node then the time consumed for searching the data is more
	There is no guarantee that data will be present in the database.

After HDBMS
-to rectify the disadvantages in hdbms they came up with another dbms as Ndbms.

NDBMs (Networking database management system)

- in Ndbms the data is going to be stored in hierarchal manner along with networking structure.



for example: -

Day 4

Advantages	Disadvantages
All the disadvantages in hdbms are rectified	It become complex with large about of data (when the no of users are increasing the complexity of the database will be increased)

RDBMS (Relational database management system)

entity

Attributes		

Here table is called as ENTITY

Column name as ATTRIBUTES

Information we contain in table is DATA

for example: -
Here the table name is student (entity)

Sid	Sname	contactno
1	Raja	123
2	Rani	234
3	Sweety	345
4	Fruity	567

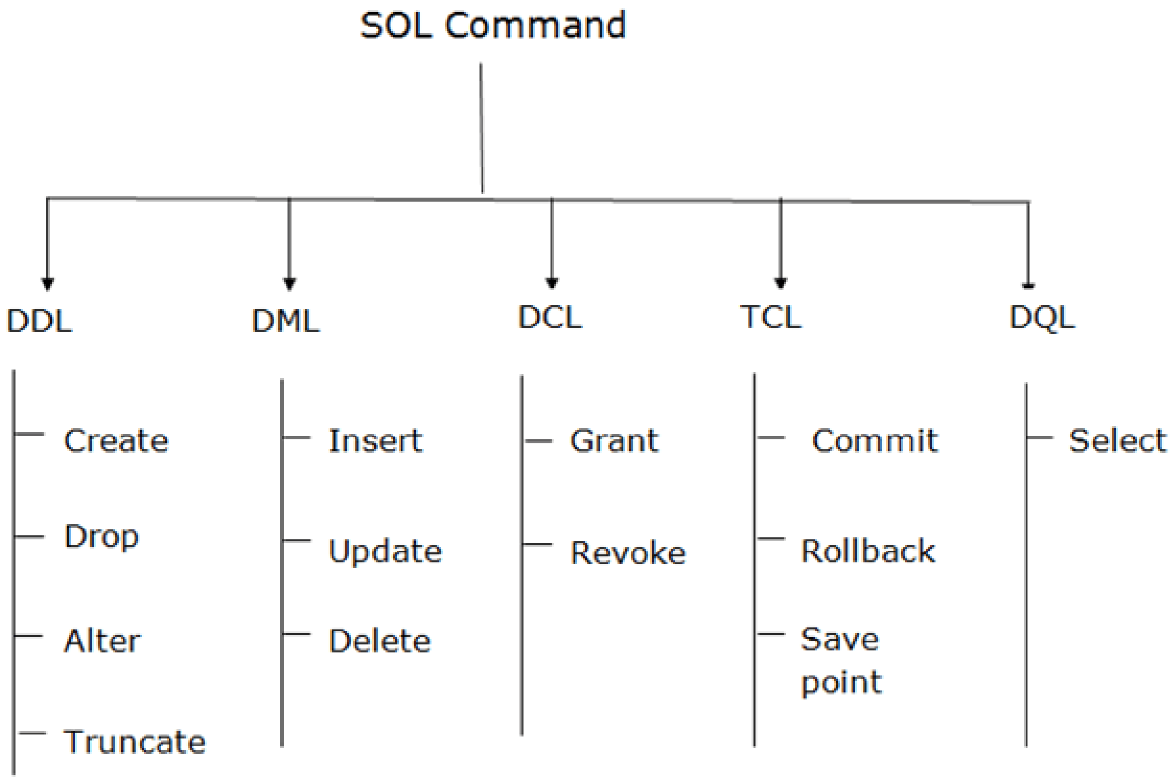
SQL summary
there are five types of statements used in SQL language/ subsets of SQL

DDL- Data definition language
DML – Data manipulation language
DTL- Data transaction language
DCL- data session control language
DQL -Data query language

DDL: - CREATE, ALTER, RENAME, TRUNCATE, DROP
DML: - INSERT, UPDATE, DELETE

DTL: - COMMIT, SAVEPOINT, ROLLBACK

DQL: - SELECT



DDL (data definition language)
with the help of the DDL we can create or delete a table

DML (data manipulation language)

With the help of DML language we can insert data or we can update the data or we can delete the data.

DTL (data transaction language)
DTL language is used to save the data inside the database permanently.

DCL (data control language)

With the help of DCL language we can give the permission or we can take back the given permission from the other users.

DQL (data query language)

With the help of DQL language we can search the data that is present inside database.

Emp table

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMN	DEPTNO

Dept table

DEPTNO	DNAME	LOC

Query example
select * from dept;

Select * from Emp;
(this will display all the data in both the tables)

Questions
Display any one column from emp table?

Ans – select column_ from table_name;

Exp- select empno from emp;

Display any two-column from emp table?

Ans – select empno, ename from emp;

Display any one column from dept table?

Ans – select deptno from dept;

Display any two- column from dept table?

Ans – select deptno, loc from dept;

Display all the employee names?

-Select ename from emp;

Display employee names and salaries of all the employees?

-Select ename, sal from emp;

Display all the department name?

-Select dname from dept;

Display location for all the departments?

-Select loc from dept;

Display employee number and employees' names from employee table?

-Select empno, ename from emp;

ALIASES

SQL aliases are used to give a table, or a column in a table, a temporary name. **Aliases** are often used to make column names more readable. An **alias** only exists for the duration of the query.

Syntax- select SAL as salary from emp;

Syntax - select SAL as “Monthly Salary” from emp;

Note: -

(“” quotes necessary when we have space in new attribute name and if we want column in particular case (upper or lower) use “ “)

Questions

Display employee name as name for all the employees.

-select ename as name from emp;

Display SAL as salary, hiredate as joining date for all the employees?

-Select sal as Salary, hiredate as “joining date” from emp;

Display employee name, employee number, SAL as monthly salary, department number for all the employees?

-select ename, empno, sal as “monthly salary”, deptno from emp;

(note - *set pagesize 200 linesize 200* , this command can change size of data in editor. Please note its editor command not sql)

SQL Literals: -

A **literal** is an explicit numeric, character, string, or Boolean value not represented by an identifier. For example, TRUE, 786, NULL, 'tutorialspoint' are all **literals** of type Boolean, number, or string. PL/**SQL**, **literals** are case-sensitive.

It is a data, we are having four types some of them are, number literal -50, character/string/ text literal- ‘hello’ ,date literal- ‘dd-mm’yy’, integer .

If literal is used in select statement, same literal we will displayed for each and every row, present in the table.

Syntax -

Select 10 from emp; - example for number

Select ‘hi’ from dept; - example for character

Concatenation operator (||)

The CONCAT() function adds two or more strings together.
Note: See also Concat with the + operator and CONCAT_WS().

Concat operator is used to merge literal and data (column), by doing this we can increase readability of the sentence.

Syntax – select ‘hello’ || ename from emp;

Question –

Display all the employee details in the below format. Smith is clerk.

- select ename || ' is '|| job from emp;

Assignment-

1.Write a query to display name of the employee.

-*select ename from emp;*

2. Write a query to display ename and salary of all employee.

-*select ename, sal from emp;*

3. Write a query to display department name and its location for all the department.

-*select dname, loc from dept;*

4. Write a query to display name, salary, commission and date of joining of every employee.

-*select ename, sal, comm, hiredate from emp;*

5. Write a query to display employee name, his designation and his manager’s employee number.

-*select ename, job, mgr from emp;*

6. Write a query to display employee name and his department number.

-*select ename, deptno from emp;*

7. write a query to display all the details of each and every employee of the company.

-*select * from emp;*

8. Write a query to display all the details of department present in company.

-*select * from dept;*

9. Write query to display employee name his employee number, his salary and also his annual salary for every employee.

-*select ename, empno, sal, sal*12 as “annual salary “from emp;*

10. write query to display employee name, for employee number and salary with the hike of 30% for every employee.

- *select ename, empno, sal+ sal*30/100 from emp;*

11. write query to display the details of employee in the following format-

a. Employee x earns a monthly salary of rupees Y

- *Select ‘Employee ‘||ename ||’ earns monthly salary of rupees ‘|| sal from emp;*

b. Employee x earns a monthly salary of rupees, Y and annual salary of rupees Z

- *Select ‘Employee ‘||ename ||' earn monthly salary of rupees,' ||sal|| ' annual salary of rupees '|| sal*12 from emp;*

c. The department x bearing department number y is located in z.

- *select ‘the department’ ||dname || ‘bearing department number‘|| deptno||‘ is located in ‘|| loc from dept;*

12. write a query to display name of the employee with annual salary with monthly bonus of 200 and provide suitable alias name.

- select ename, (sal*200)*12 “annual salary with monthly bonus” from emp;

13. write a query to display name, salary along with annual salary of employee with yearly bonus of 100

- *select ename, sal, sal*12+ 1000 as “annual salary “from emp;*

Operators: -

Arithmetic operators- +, -, *, /

Relational operators- <, >, =, <=, >=, !=

Logical operators- and, or, not

Special operators- like, between and, in is null.

Use of where in sql: -

If you trying to get particular row from table, we use condition with the help of ‘where’.

For example, I am trying to find all the information about smith from emp table but only need smith data not any other employee information, in this case I’ll need to use ‘where’.

Syntax – *select * from emp where ename=’SMITH’;*

(Note – as ename is character literal we must write smith in quotes ‘ ‘.

And smith in uppercase as it is in emp table.)

Questions: -

1. Display all the information of Scott?
   *Select * from emp where ename =’SCOTT’;*

2. Display Adam Job?
   Select job from emp where ename=’ ADAMS’;

3. Display miller’s salary and department
   select sal, depno from emp where ename= ‘MILLER’;

4. Display employee name of all the clerks
   Select ename from emp where job= ‘CLERKS’;

5. Display all the details of the employee who are working in department number 20?
   *select * from emp where deptno=20;*

6. Display all the employee name whose salary is more than 1500?
   select ename from emp where sal>1500;

7. Display employee name, job, salary of all the employees whose employee number is 7902?
   select ename, job, sal from emp where empno=7902;

8. Display all the employee’s information who joined on 30-dec-1981?
   *select * from emp where hiredate='30-DEC-1981';*

9. Display all the analyst information?
   *select * from emp where job='ANALYST';*

10. Display all the employee names who are working under 7698 managers.

select ename from emp where MGR=7698;

11. write a query to select all the salary of employee of the salary is greater than 1500.
   *select * from emp where sal <1500;*

12. write a query to display employee name, his hire date, his salary and annual salary only if his annual salary greater than 10000.

    *Select ename, hiredate, sal, sal*12 from emp where (sal*12)>10000;*

13. Write query to display all the details of employee if the department number is 20

    *Select * from emp where deptno=20;*

14. Write query to display all the details of the employee if designation is manager.

    *Select * from emp where job= 'MANAGER';*

15. Write a query to display all the details of employee only if they were hired after the year 1995

*Select * from emp where hiredate>'31-DEC-1995';*

Note: - SQL language is not case sensitive but the data present inside the table are case sensitive.

Example – select, from, where, table name, column name, operator all are non-case sensitive.

Data present inside the table is always case sensitive.

Distinct: -

Distinct is used to display unique data present inside the table.

A	A
B	B
A	Deleted
C	C
B	Delete
D	D
E	E
D	Delete
F	F

Example –

Select distinct(job) from emp;

Output: -

JOB

CLERK

SALESMAN

PRESIDENT

MANAGER

ANALYST

Questions

1.Display employee information whose comm is 1400.

*Select * from emp where comm=1400;*

2. Display employee information of clerk who works for department 20.

*Select * from emp where job=’CLERK’ and deptno=20;*

3. Display department number 20 employee who gets more than 1500

Select deptno, sal from emp where sal <1500 and deptno=20;

4. Display employee names of department number 10,30.

Select ename from emp where deptno= 10 OR deptno=30;

5. Display manager and analyst information.

*select * from emp where job= 'MANAGER' or job='ANALYST';*

6. Display employee whose earning is between 1000 and 2000 and include the same salaries also.

*Select * from emp where sal>=1000 and sal<=2000;*

7. Display employee name who joined in the year 1981.

Select ename from emp where hiredate>= ’01-JAN-1981 and hiredate<=’31-DEC-1981’;

Special operators: -

Between-one operator: -

When we have any rang of values we can use between- and operator.

Syntax- column name between lower range and higher ranger.

Example- select * from emp where sal between 1000 and 2000;

In operator: -

If we have multiple OR condition for a signal column then we can use IN operator.

Syntax: - column name in (data, ….)

Example – select * from emp where job in (‘MANAGER’, ‘ANALYST’);

Is null operator: -

This operator is use to get all null values in the column.

Is null operator is used to find out the null data that is present inside the table.

Syntax: - column name is null;

Example: - select * from emp where comm is null;

Not operator: -

Not operator cannot be used solo, it needs to club with any other operator. Like Not between and, not in, is not null, Not like.

Example: -

Select * from emp where comm is not null;

Like operator: -

The **SQL** Server **LIKE** is a logical **operator** that determines if a character string matches a specified pattern. A pattern may include regular characters and wildcard characters. The **LIKE operator** is used in the WHERE **clause** of the SELECT, UPDATE, and DELETE statements to filter rows based on pattern matching.

Like operator is a wild card operator which is used to perform wildcard operations.

We have two wild card operators

%- represents 0-n character.

_ - represents signal character.

Syntax – column name like ‘data’;

Example – ename like ‘S%’; (display all the name starts with S)

ename like ‘%S’;(display all the name ends with S)

ename like ‘_L%’;(display all the name whose 2nd letter is L)

ename like ‘%E_’; (display all the name whose last 2nd letter is E) example: - select * from emp where ename like ’S%’;

select * from emp where ename like ‘%S’;

select * from emp where ename like ‘_L%’;

select * from emp where ename like ‘%E_’;

Questions: -

1. Write a query to display Ename, salary, annual salary with the like of 25% only if he is clerk and his date of hire is after 16th June 1982.

*Select ename, sal, (sal*12)+25/100 from emp where job=’CLERK’ and hiredate>’16-JUN-1982’;*

2. Write a query to display all the details of employee only if annual salary is greater than 15000 and his commission is greater than his salary.

*Select * from emp where (sal*12)> 15000 and comm>sal;*

3. Write a Query to display all the details of department only if department number is 100 or name of department is RESEARCH.

*Select * from dept where deptno=10 or dname=’RESEARCH’;*

4. Write a query to display employee name and his employee number and his hired date and salary with reduction of 20% only if his salary is greater than 2500 of his department number is 10.

```
Select ename, empno, hiredate, sal-20/100 from emp where sal >2500 and deptno=10;
```

5. Write a query to display all the details of employee, if they have joined the company before 1985 or his salary is greater than 2500.

```
Select * from emp where hiredate<’31-DEC-1985’ or sal>2500;
```

6. Write a Query to display all the details of employee who belongs to department number is 10 or 20 and salary greater than 2000.

```
Select * from emp where deptno in(10,20)and sal>2000;
```

7. Write a query to display all the details of employee he is working as a salesman or analyst and department number either 10 or 20 salary should be greater than 2000.

```
Select * from emp where job in (‘SALEMAN’,‘ANALYST’) and deptno in (10, 20) and sal >2000;
```

8. Write a Query to display all the details of employee who is working as SALESMAN and ANALYST.

*Select * from emp where job in(’ SALESMAN’,‘ANALYST’);*

9. Write Query to display all the details of employee he working either salesman or clerk and salary greater than 3000 or not equal to 2500 and department number either 30,10, 40.

*Select * from emp where job in(’SALEMAN’ ,’CLERK’)and (sal>3000 or sal not in(2500)) and deptno in(10,30,40);*

10. Write a Query to display employee name, job, commission, depart number, existing salary, new salary with the reduction of 35% only if job not equal to check and depart number either 10, 20, 30. Hire date greater than 1983 and omission should be less than salary.

Select ename, job,comm,deptno,sal,(sal-35/100) from emp where job!=’CHEAK’ and deptno in(10,20,30) and hiredate >’31-DEC-198’ and comm<sal;

11. Write a Query to display all the details of employee if number either 10, 20, 30.

*Select * from emp where deptno in (10,20,30)*

12. Write a Query to display all the employee who belong to department number 10 or 20 and comm between 1000 and 3000 and were hired during 1980 to 1983.

*Select * from emp where deptno in(10,20) and comm between 1000 and 3000 and hiredate between ‘01-JAN-1980’ and ‘31-DEC-1983’;*

13. Write a query to display all the details of employee if he is not having the reporting manager.

*Select * from emp where MGR is null;*

14. Write Query to display all the employee who don’t take comm.

*select * from emp where comm is null;*

15. Write a Query to select a name which begins with ‘A’.

Select ename from emp where ename like ‘A%’;

16. Write a Query to select the name which begins with ‘A’ and ends with ‘A’.

Select ename from emp where ename like ‘A%A’;

17. Write a Query to select a name which is having at least 2 ‘a’ in it.

Select ename from emp where ename like’%A%A%’;

18. Write a query to select the name which starts with ‘M’.

Select ename from emp where ename like ‘M%’;

19. Write a Query to match name to select string which is having substring MAD.

Select ename from emp where ename like ‘%MAD%’;

Functions: -

Functions. "A **procedures** or **function** is a group or set of **SQL** and PL/**SQL** statements that perform a specific task.". The major difference between a **procedure** and a **function** is, a **function** must always return a value,

-functions are used to perform operations.

- there are two types of functions, they are :

Single row functions

Multi row functions

Single row functions:

These single row functions will take each row as input and produce output for each row.

Like that it will create a corresponding output for every row.

(take each input at a time and perform operation to produce output)

Explanation

Examples of single row function

Length: - length function is used to get a length of the string.

Syntax- length(data)

Select ename, length(ename) from emp;

SMITH- 5

MILLER- 6

Lower: - lower function is used to convert text into lower case.

Syntax- lower(data)

Select ename, lower(ename) from emp;

SMITH- smith

MILLER- miller

Initcap: -

This function actually used to get first letter of word uppercase and remaining in lowercase.

Initcap- inticap(data)

Select ename, inticap(ename) from emp;

SMITH- Smith

MILLER- Miller

Concat: -

Concat function is used to combine two strings. we can only use two string not more than that.

Concat (ip1, ip2)

Concat (smith, clerk)- smithcleak

Concat (miller, Allen)- millerallen

Syntax- select ename, job, contact (ename, job) from emp;

Nested functions: -

If function is written inside another function it is known as nested function.

Fun(fun(data, data),data)

Example: -Select length(lower(ename)) from emp;

Replace: -

The REPLACE() function replaces all occurrences of a substring within a string, with a new substring.

Note: The search is case-insensitive.

REPLACE(*string*, *old_string*, *new_string*)

String- required, the original string

Old_string- requires. The string to be replace

New_string- required. The new replacement string.

Example- select ename, replace(ename,’A’,’X’) from emp;

select ename, replace(ename, ‘A') from emp;

replace(‘java’,’a’,’b’)-jbvb

replace(‘java’,’x’,’y’)- java

replace(‘java’,’a’)- java

Write a query to select name which is having substring NA at least twice

Select ename from emp where ename like ‘%NA%NA%’;

Write a query to display all the employee’s whose name is having ‘R’ as the second last character.

Select ename from emp where ename like ‘%R_’;

List all the employee whose name starts with ‘s’

Select ename from emp where ename like ‘S%’;

List all the employee whose name is having letter L as second character

Select ename from emp where ename like ‘_L%’;

List all the employee whose name is having at least 2 L in it.

Select ename from emp where ename like ‘%L%L%’;

List all the employee whose name having letter ‘E’ as the last but one character

Select ename from emp where ename like ‘%E_’;

List all the employee whose name is having letter ‘R’ in the 3rd position.

Select ename from emp where ename like ‘__R%’;

SUBSTRING FUNCTION

Substring function is used to diplay particular part of string.

Syntax; substr(ip1,ip2,ip3)

Ip1- data

Ip2- position to start

Ip3 – no of character

Substr(‘developer’,1,3) – dev

Substr(‘developer’,3,3) – vel

Substr(‘developer’,5,1) – l

Substr(‘developer’,-3,3) – per

Substr(‘developer’,5) – loper

Substr(‘developer’,-2)- er

D(1,-9)	**E**(2,-8)	**V**(3,-7)	**E**(4,-6)	**L**(5,-5)	**O**(6,-4)	**P**(7,-3)	**E**(8,-2)	**R**(9,-1)

Example to display first chartere of every emplyoee name

Select ename, substr(ename,1,1) from emp

Question

Display first letter of every employee name

Select ename, substr(ename,1,1) from emp;

Display last letter of every employee name

Select ename, substr(ename,-1,1) from emp;

Display last letter of every employee name without using -1?

Select ename, substr(ename,length(ename),-1) from emp;

Display first three characters of every job

Select job, substr(job,1,3) from emp;

Write a query to remove A character from every employee name

select ename, replace(ename, ‘A') from emp;

Write a query to count no of A characters present in every employee name.

Select ename, length(ename)- length(replace(ename,’A’))from emp;

Display employee names whose name is having 4 characters.

Select ename from emp where length(ename)=4;

Display employee names and jobs whose job is starting with man.

Select ename, job from emp where Substr(job,1,3)='MAN';

Display all the employee details who joined in the month of feb.

Select * from emp where substr(hiredate,4,3)='FEB';

Display employee names whose name is having last but one character as E.

Select ename from emp where substr(ename, -2,1)='E';

Instring function: -

Syntax instr(ip1,ip2, ip3,ip4)

Ip1 – data

Ip2- character to search

Ip3- position to start

Ip4- occurrence

Instring function is used to display the position of character.

Instr('developer', 'e', 1,1) output: - **2**

Instr('developer', 'e',3,1) output: - 4

Instr('developer', 'e',1, 3) output: - 8

Instr('developer', 'e',4,1) output: - 4

Instr('developer', 'e',5,2) output: - 0

Instr('developer', 'e',-4,1) output: - 4

Instr('developer', 'e',-3,2) output: - 2

Instr('developer', 'vel',1,1) output: - 3

Instr('developer', 'vell',1,1) output: - 0

Instr('developer', 'e') output: - 2

D(1,-9)	**E**(2,-8)	**V**(3,-7)	**E**(4,-6)	**L**(5,-5)	**O**(6,-4)	**P**(7,-3)	**E**(8,-2)	**R**(9,-1)

Example: Select ename, instr(ename, 'A', 1, 1) from emp;

NVL function: -

Nvl(column name, value)

It is used to replace null data with zero. For example, we have null values in comm column in emp table. So, we can use NVL to replace values with o.

Select comm, nvl(comm, 0) from emp;

Q. Display me total salary of the all emp incuding comm.

Select comm, sal, sal+NVL(comm,0) from emp;

Nvl2 function: -

Nvl2 (column name, value1, value2)

If column having null data it will replace it with value2 and if it having any data it replaces with value1.

With the help of this function you can replace null data and not-null data also.

Example: Select comm, sal, sal+NVL2(comm, comm,0) from emp;

sysdate: -

If you want to display todays date (date of your system)

Example: - select sysdate from emp;

Systimestamp: -

This will display date, time and time zone.

Example: -Select systimestamp from emp;

Trunc function: -

If you give decimal value as input it will remove decimal and give integer value as output.

from example trunc(72.3) – 72

trunc(75.8)- 75

Round function: -

If you give decimal value as input ,it will round off decimal value and give integer value as output.

round(72.3) -72

round(75.6) -76

round(73.5) -74

round(79.8)- 80

Mod function: -

Mod fun is consisting of inputs. Mod function is used to display reminder of this values.

Example: -

Mod(20,2) – 0

Mod(25,2) – 1

Question:

1. Display employee name, sal, no of days working in the company.
   Select ename, sal, (sysdate- hiredate) as "No of working days" from emp;

2. Display the employee names who had worked more than 12000 days.
   Select ename from emp where (sysdate- hiredate) >12000;

3. Display employee names, sal, yearsvof experience for all the all the employees.
   Select ename, sal, round(((sysdate-hiredate) /365)) as "years of experience" from emp;

Order by statement: -

Order by statement is used to arrange the data based on column name or condition.

Example: -

Select * from emp order by ename;

Select * from emp order by ename desc;

Select * from emp order by deptno, job;

Select * from emp order by sal*12

Select ename, sal, sal*12 as “ annual salary” from emp order by “annual salary” ;

Questions: -

Display list of employees joined on 01-may-81, 03-dec-81, 17-dec-81, 19-jan-80 in ascending of seniority.

*Select * from emp where hiredate in (‘01-MAY-81’, ’17-DEC-81’, ’19-JAN-80’) order by hiredate;*

Display employees in the ascending order of designation who joined after second half of 1981.

*select * from emp where hiredate>=’01-JUL-1981’ order by job;*

List all the employees who does not belong to department no 20.

*Select * from emp where deptno !=20;*

Multi-row functions : -

Select max(sal) from emp; (to display maximum salary of emp)

Select min(sal) from emp; (to display minimum salary of employee)

Select avg(sal) from emp; (to display average salaries of employees)

Select sum(sal) from emp; (to display sum salaries of all the employees)

Select count(comm) from emp; (to display count of input values(rows))

Questions: -

1. Display no of analysts.

   select count(JOB) from emp where JOB='ANALYST';

2. Display average salary among all the employees working in department no 20.

   select avg(sal) from emp where deptno= 20;

3. Display least salary earned by manger.

   select min(sal) from emp where job='MANAGER';

4. Display total salary earned by employees who joined in the year 81.

Select sum(sal) from emp where hiredate between '01-JAN-81' and '31-DEC-81';

5. Display no of employee in department number 10.
   Select count(empno) where deptno=10;

6. Display no of employee present in each department.
   Select deptno, count(*) from emp group by deptno;

Group by: -

Group by statement is used to dived a table based on condition or column name.

1. Display no of employees working in each Job.
   Select job, count(*) from emp group by Job;

2. Display maximum salary earned in each job.
   Select Job, max(sal) from emp group by Job;

3. Display no of employees in each job, select only those where no of employees is 3 and above.

   Select job, count(*) from emp group by Job having count(ename)>=3;
   (here we are using having keyword which is similarly to where but can be used after dividing the table)

4. Display total salary of each department only if the total salary exceeds 9000?

   Select deptno, sum(sal) from emp group by deptno having sum(sal)>9000;

Joins:

Joins are used to merge two or more tables.

Equi join-

For joining of two tables, if we use equal to (=) operator then it represents as Equi-join.

Syntax: - select * from emp, dept where emp.deptno=dept.deptno;

Questions: -

Display employee name and department name for all the employee.

Select ename, dname from emp, dept where emp.deptno=dept.deptno;

Display employee name, location and department number for all the employees.

Select ename, loc, emp.deptno from emp, dept where emp.deptno=dept.deptno;

Display ename and dname of all the salesman.

Select ename, dname from emp, dept where Job='SALESMAN' and emp.deptno=dept.deptno;

Non Equi Join: -

For joining of two tables , if we doesn't use equal to(=) operator then it's called as Non equi join.

Syntax: -

Select * from emp, salgrade where sal between losal and hisal;

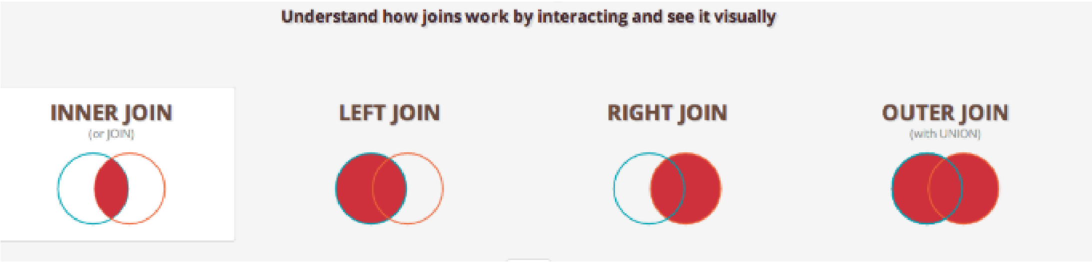
Cross join: -

While joining two table if we don't give any join condition then it performs cross join.

Cross join actually means all the records of first table will be merged with first record of two table similarly the merging continues till the end of all the records.

Syntax – select * from emp, dept;

(as emp consist of 14 and dept consist of 4 record it will give 56 records as output)


Understand how joins work by interacting and see it visually
INNER JOIN
(or JOIN)
LEFT JOIN
RIGHT JOIN
OUTER JOIN
(with UNION)

Inner join: -

Inner join will display common information that is present both the tables.

Select * from emp, dept where emp.deptno=dept.deptno;

Outer join: -

Left join: -

The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2). The result is NULL from the right side, if there is no match.

```
Select * from emp, dept where emp.deptno=dept.deptno(+);
```

Right join: -

The RIGHT JOIN keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match

```
Select * from emp, dept where emp.deptno(+)= dept.deptno;
```

Full outer join: -

The FULL OUTER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records.

Note: FULL OUTER JOIN can potentially return very large result-sets!

(FULL OUTER JOIN and FULL JOIN are the same)

Constraints: -

Constraints are restriction that are applied on the table.

Unique constraint: -

Unique constraint will not allow duplicate data but will allow null data.

For example: - below table we should have unique data in Student Id buut it will accept null value.

StudentId	Name	Contact No
1	Raja	123
2	Rani	234

3	Sweety	124
	Fruity	121

Not Null: - Not null constraint will not allow null data but it allow duplicate data.

PRIMARY KEY Constraint

The PRIMARY KEY constraint uniquely identifies each record in a table.

Primary keys must contain UNIQUE values, and cannot contain NULL values.

A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

(it is combination of unique and not null constraints. It is also known as unique identifier. In the hole table I can only one primary key)

FOREIGN KEY Constraint

A FOREIGN KEY is a key used to link two tables together.

A FOREIGN KEY is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another table.

The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.

(Foreign key constraint is used to build the relationships between tables. It is also known as referential integrity constraint. In a signal table you can have n no. of foreign keys.)

DDL (Data definition language)

Creating table

Syntax:

For parent table

SQL> create table Department(deptno number(4) primary key, dname varchar(15) not null, loc varchar(10));

Child table

SQL> create table emp420(empno number(4) primary key, ename varchar(30) not null, deptno references Department(deptno));

<u>Alter: -</u>

Alter is used to add, remove or modify column in existing table.

<u>Add column-</u>

we are trying to add column name remark in table emp420.

Syntax: - alter table table_name add(column_name datatype(size) constrain,..);0

SQL> alter table emp420 add(remark varchar(30) not null);

<u>Rename column: -</u>

We are trying to rename column name remark to rating

Syntax: -Alter table tablename column oldcolumn to newcolumn;

SQL> alter table emp420 rename column remark to rating;

Table altered.

Remove column: -

To remove column, we use this syntax.

Syntax: -

Alter table table_name drop column column_name;

SQL> alter table emp420 drop column rating;

Table altered.

Modify the datatype: -

This is used to change datatype of column.

Syntax: -Alter table table_name modify (column_name newdatatype(size));

SQL> alter table emp420 modify(remark number(2));

Table altered.

Renaming table: -

Syntax: - Rename old table name to new table name;

SQL> rename emp420 to emp520;

Table renamed.

Drop: -

We can using delete table using drop.

Syntax: - drop table tablename;

SQL> drop table emp520;

Table dropped.

Truncate: -

It is used to delete only data inside the table but table structure will remain as it is.

Syntax: - truncate table tablename;

Truncate table emp520;

DML (data manipulation language

1. Insert: - this is used to insert data inside the table.

Syntax Insert into table_name values(data);

insert into emp values(7945, 'MARK', 'CLERK', 7789, '17-JAN-95', 950,0, 20);

2. update: -

Syntax: -Update tablename Set columname=data;

SQL> update emp set deptno=40 where ename= 'MARK';

1 row updated.

3. Delete: -

Syntax- delete tablename where columnname=data;

SQL> delete emp where ename='MARK';

1 row deleted.

DCL: - DCL is short name of Data Control Language which includes commands such as GRANT and mostly concerned with rights, permissions and other controls of the database system.

Syntax:

Grant select on emp to hr; // here hr is another user such as scott

Revoke select on emp from hr;// to revoke permission

DTL: -

TCL(transaction Control Language) : TCL commands deals with the transaction within the database.

Examples of TCL commands:

- COMMIT– commits a Transaction.
- ROLLBACK– rollbacks a transaction in case of any error occurs.
- SAVEPOINT–sets a savepoint within a transaction.
- SET TRANSACTION–specify characteristics for the transaction.

Syntax: -

Commit – commit;

Save point- savepoint name;

Roll back- Rollback to name;