**Single Table Queries**

Select \* from emp; 🡪 Retrieves all columns and rows of emp table.

Select job from emp; 🡪 Retrieves only job column data of emp table.

Select distinct job from emp; 🡪 Hide Duplicate and show unique rows in job column of emp table.

Select \* from emp WHERE job = ‘MANAGER’ ; 🡪 Filter the records and will only fetch the data which satisfied the condition job = ‘MANAGER’ .

Select \* from emp WHERE job = ‘MANAGER’ AND name = ‘ALLEN’; 🡪 Filter the records and will only fetch the data which satisfied the condition job = ‘MANAGER’ AND name =’ALLEN’.

**Operators: =,!=, >, <, >=, <=**

Select \* from emp where job != ‘MANAGER’ AND sal > 3000 AND deptno = 20 🡪 Fetch the data which satisfied all the given conditions.

**If we want fetch details where condition can satisfy either of values we should use OR.**

Select \* from emp where job = ‘MANAGER’ OR job = ‘SALESMAN’; 🡪 Fetch the records when job column has manager value or salesman value.

**Exercise: Return names of the employees where job is not manager nor salesman and sal >=2000**

Select ename from emp where job != ‘MANAGER’ AND job != ‘SALESMAN’ AND sal >=2000;

**Exercise: Return names,hiring dates of the employees where location in Dallas or Chicago.**

Select ename, hiredate from emp where deptno = 20 or deptno = 30;

**IN Clause: To reduce the number of times to write deptno for OR Operator, we can use IN Clause**

Select ename,hiredate from emp where deptno **IN** (20,30)

Select ename, hiredate from emp where deptno **NOT IN** (20,30) 🡪 Returns all the rows except the depno column values with 20,30 .

**BETWEEN OPERATOR: TO filter data in the given range we use BETWEEN. we can use on numbers, dates and Textual data.(INCLUSIVE)BETWEEN 1000 AND 2000 (1000 and 2000 included)**

Select \* from EMP WHERE hiredate BETWEEN ‘05/01/1981’ AND ‘12/09/1982’; 🡪 Fetches records where hiredate is between the given range.

Select \* from EMP WHERE sal NOT BETWEEN 600 AND 1500 🡪 Fetches records which will not come under the given range condition for sal column.

select \* from emp where comm is NULL; 🡪 Returns the records where comm column has null value.

Select \* from emp where comm is NOT NULL 🡪 Returns the records where comm Column has values.

**USE Parenthesis () if we want to execute some conditions together.**

select \* from emp where (comm = 0 OR comm is NULL ) AND sal BETWEEN 1101 AND 4999 AND sal <> 3000;

**LIKE Operator: Wild Cards (%, \*, )**

**Aliasing:-**

Select ename Employee\_Name, comm Commision, sal Salary from emp; 🡪 Get all records for the columns we want.(We will get alias names instead of original column names… Alias names we can use underscores instead of space, if we want space then give alias names in Double Quotes(“Employee Name”).

**Sql Specific for Aliasing :-** select ename **AS** Employee\_Name,comm **AS** “Commision” from emp;

**Concatenating String with Records:-**

Select ‘Hello, My Name is ‘ || ename from emp; 🡪 We will get Ex:- Hello, My Name is Srikanth with the record name srikanth in ename. We can use filter conditions too.

select ename || ' makes ' || sal || ' per month' from emp**;**

select ename || ‘makes’ || sal || ‘per month’ as “Empl Sal Per Month” from emp;

**ORDER BY:**

Select \* from emp order by ename; [ascending by default]

Select \* from emp order by sal desc; [records with sal column descending order]

Select \* from emp order by deptno, sal; [First sql will get records with order by deptno ascending order, then in that if deptno contain with 10 value three times then in that records sal will be in ascending order, and deptno 20 value comes 5 times then sal values will be in ascending order]

**Questions for this assignment**

Next Page…..



1. Write a query that retrieves suppliers that work in either Georgia or California.

Select \* from suppliers where state in (‘Georgia’,’California’);

1. Write a query that retrieves suppliers with the characters "wo" and the character "I" or "i" in their name.

Select \* from suppliers where supplier\_name like ‘%wo%’ and (supplier\_name like “%I%” or supplier\_name like “%i%”);

1. Write a query that retrieves suppliers on which a minimum of 37,000 and a maximum of 80,000 was spent.

Select \* from suppliers where where total\_spent between (37000,80000);

1. Write a query that returns the supplier names and the state in which they operate meeting the following conditions:
   1. belong in the state Georgia or Alaska
   2. the supplier id is 100 or greater than 600
   3. the amount spent is less than 100,000 or the amount spent is 220,000

select supplier\_name,state from suppliers

where state in (‘Georgia’,’Alaska’)

AND (supplier\_id =100 or supplier\_id>600)

AND (total\_spent < 100,000 or total\_spent =220,000);

TRUE or FALSE Question:

1. The keywords such as SELECT and WHERE must always be capital in the SQL Query. **False**
2. The database works on first processing the filtering conditions and then processes the FROM condition. **False**
3. Having just the filter condition shown below in a SQL query will return all of the records from the table. **WHERE 1 = 1: True**
4. NULL can not be compared using an equal sign. **True**
5. The ORDER BY clause is processed before the FROM clause in a SQL statement and it's used to sort the columns in an ascending or descending fashion. **False**

**Structure:-**

1. **Select**
2. **From**
3. **Where**
4. **Group BY**
5. **Having**
6. **Order by**

**Single Row Functions:-** SRF’s will apply only on single record, i.e., concatenating two column records or three columns, etc…

1. **concat(‘This is’,ename):** select concat(‘My Name is ‘,ename) from emp;
2. **upper(ename):** select upper(ename) as Ename from emp;
3. **lower(ename):** select lower(ename) as ename from emp;
4. **select ‘hello’ from dual:** To get only one record o/p we use dual table;
5. **select concat(upper(ename),lower(‘ IS THE NAME’)) from emp:** function inside function;

select concat(concat(lower(ename),upper(' is the name ')),concat(lower(' and their job is: '),upper(job))) as Output from emp;

1. **initcap(‘my name is srikanth’):** select initcap(‘my name is srikanth’) from dual; O/P:- My Name Is Srikanth
2. **length(ename):** select ename,length(ename) from emp where length(ename) = 5;
3. **substr(ename,2,2):** first argument:- string or column, SecondArg: starting letter postion, ThirdArg:How many letters we want to show from selected starting position.

Ex: select substr(‘Srikanth’,2,4): O/p:- rika, select substr(‘Srikanth’,2): O/P:- rikanth

1. **lpad(‘Day’,6,’%’):** select lpad(‘Day’,6,’%’) from dual; O/P:- %%%Day
2. **rpad(‘Day’,6,’%’):** select rpad(‘Day’,6,’%’) from dual; O/P:- Day%%%
3. **lpad(‘Day’,6):** select lpad(‘Day’,6) from dual; O/P:- ‘ Day’
4. **ltrim(‘ddday’):** select ltrim(‘dddaydd’,’d’) from dual; O/p:- aydd
5. **rtrim(‘ddday’):** select rtrim(‘dddaydd’,’d’) from dual; O/p:- ddday
6. **trim(‘ddday’):** select trim(‘dddaydd’,’d’) from dual; O/p:- ay
7. **round(107.342,2):** O/P:- 107.34, round(107.3529,3): O/P:- 107.353,round(107.9):-108
8. **trunc(107.34,1):** O/P:- 107.3, trunc(107.222):- 107 :- it wont round off.
9. **Sysdate:** **OS current date** select sysdate from dual;
10. **Systimestamp:** OS current date and time
11. **Addmonths(‘11/13/1992’,4):**- **‘03/13/1993,** addmonths(‘11/13/1992’,-4):- ‘7/13/1992’
12. **Months\_between( ‘11/13/1992’, ‘7/13/1992’):- 4 (first arg – sec arg)**
13. **trunc(sysdate,’month’):- 14-01-2022 :-** It will give present year,date,first month
14. **trunc(sysdate,’year’):- 01-01-2022 :-** It will give present years first day and month.

**GROUPING FUNCTIONS:-----------**

**Group BY:-**

**It takes N number of Records and gives one record o/p: MAX, MIN, AVG, SUM, COUNT**

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

**If there is null in the record count function will not count that as a record.**

If we want individual job types avg salary we can do as follows. Otherwise we should write separate queries for individual job titles.

\*\*\* **To get MAX, MIN, AVG, SUM, COUNT of particular section of records with different groups. We use GROUP BY Clause.**

For each job title if we want avg or sum or min salary we should use group by.

Select job, max(sal) from emp GROUP BY job;

**Group By first groups the data with the similar record values for the given column and applies function on the bucket of similar values.**

\*\*\* **We cannot use Where Clause in Group Functions.**

**\*\*\*\*When we want to filter the query when we are using Group By Function use \*\*\*Having clause.**

**\*\*\* The columns we are fetching using select statement should also be after GROUP BY clause But all the columns are after GROUP BY Clause may not need to be after select statemnt.**

Select c1,c2,c3,max(\*) from emp GROUP BY c1,c2,c3,cn;

Select job, count(\*) as count from emp GROUP BY job HAVING count(\*)=2;

**\*\*\* We can group by n number of columns. Ex:- There are 6 managers with deptnos 10-2,20-3,30-1.**

**If we use group by for job and deptno and cont the records we will get**

**Manager 10 2**

**Manager 20 3**

**Manager 30 1**

**Sub Queries:- \*Try get single value from sub-query and map that result value to main table for filter.**

Queries in Queries is called as Sub-Queries.

Select \* from dept where deptno=(select deptno from deptno where deptno=30);

Select \* from emp where deptno=(select deptno from dept where loc=’CHICAGO’);

Select ‘\*’ from dual; 🡪 It will return single value. So we can use this subquery after select clause to get only one value.

Select ename,job,(select ‘Hello” from dual) as ‘Hello’ from emp;

**JOINS:**

Select e.ename,e.sal,d.loc from emp e,dept d where e.deptno=d.deptno and d.loc=’CHICAGO’;

**INNER JOIN:-** fetch data which is common in two tables.

Select e.ename,e.sal,d.loc from emp e INNER JOIN dept d ON e.deptno=d.deptno;

**RIGHT JOIN:-** Right table whole records and left table matching records with right table.

Select e.ename,e.sal,d.loc from emp e RIGHT JOIN dept d ON e.deptno=d.deptno;

**LEFT JOIN:-** Left table whole records and right table matching records with left table.

**EXISTS & NOT EXISTS:- Correlated Sub-Queries [outer table rec can be compared inside subquery)**

Select \* from emp where EXISTS (select ‘hi’ from dual);

Select \* from emp where NOT EXISTS (select ‘hi’ from dual);

Select e.\* from emp e Where EXISTS (select \* from dept d where e.deptno=d.deptno);

**SELF JOIN:- Nothing but inner join or left join or right join two same tables.**

**Ex:-** For emp table there is a **empno column** which contains emp id’s and there is **mgr column** which **contains employees Manager id’s** which will agin map to the same table with **empid’s.** So we can use self joins here.

**CASE STATEMENT:- like IF THEN ELSE Statement.**

Select ename, job, (CASE job WHEN ‘PRESIDENT’ THEN ‘BIG SHOT’ WHEN ‘ANALYST’ THEN ‘GOOD AT MATH’ ELSE ‘NO COMMENT’ END) as comments from emp;

select ename, sal, (CASE WHEN sal >= 3000 THEN 'BIG SHOT'WHEN sal < 3000 THEN 'HARD WORKING' END) Type from emp;