1. **Write a SQL Query to fetch all the duplicate records in a table.**

Step 1 : apply row no. partition by name

Step 2 : and then just fetch the records where row no > =2

With temp as

(Select \*, row\_num(partition by name) as rn

From  table )

Select \* from temp

Where rn >=2

          —-------------------------------------------\*\*\*\*\*--------------------------------------------------------

Select count(id) as total, name

From table

Group by name

Having count(id) as total >=2

**2. Write a SQL query to fetch the second last record from employee table.**

Step 1 : apply row\_num to id in desc

Step 2 : fetch the rn = 2

Select \*, row\_num ( order by id desc) as rn

From table

Where row\_num (order by id desc) as rn >= 2

**3. Write a SQL query to display only the details of employees who either earn the highest salary or the lowest salary in each department from the employee table.**

Step 1 department wise salary min max use group by

Step 2  join the main table with department and min

Step 3 join the main table with department and max

With temp as

(Select min(salary), max(salary), department

From table

Group by department)

Select temp.min\_sal, temp.max\_sal, \*.table

From table join temp on table.DEPARTMENT = temp.department

With temp as (select min(salary) as min\_salary, max(salary) as max\_salary,department

from table

group by department)

With temp as (Select row\_num() over (partition by department order by salary)

From table)

Select \*

From temp

Where row in ( select max(row) from temp group by department)

Union

Select min(row)

From temp

**4. From the doctors table, fetch the details of doctors who work in the same hospital but in different specialties.**

STEP 1 : SELF JOIN THE TABLE

STEP 2 : AND THEN PUT FILTERS

SELECT

FROM TABLE T1 JOIN TABLE T2

ON T1.ID <> T2.ID AND T1.HOSPITAL = T2.HOSPITAL AND T1.

**5. From the login\_details table, fetch the users who logged in consecutively 3 or more times.**

STEP 1 : USE LAG WINDOWS FUNCTION COMPARE IT TO 2 CONSECUTIVE RECORDS

SELECT \*

FROM

(SELECT CASE WHEN USERNAME = LAG ( USERNAME ) AND

  USERNAME = LAG ( USERNAME, 2 ) THEN USERNAME

ELSE NULL END AS USER) X

WHERE X.USER IS NOT NULL

**6. From the students table, write a SQL query to interchange the adjacent student names.**

STEP : WE WILL USE CASE FOR FILTERING EVEN AND ODD

STEP 2: FOR EVEN WE USE LEAD AND FOR ODD WE WILL USE EVEN

SELECT ID, STUDENT\_NAME,

CASE WHEN ID % 2 = 0 THEN LEAD(STUDENT\_NAME)

ELSE

LAG(STUDENT\_NAME) END AS NEW\_STUDENT\_NAME

FROM TABLE;

select id,student\_name,

case when id%2 <> 0 then lead(student\_name,1,student\_name) over(order by id)

when id%2 = 0 then lag(student\_name) over(order by id) end as new\_student\_name

from students;

**7. From the weather table, fetch all the records when London had extremely cold temperature for 3 consecutive days or more**

STEP 1 : USE LEAD FUNCTION COMPARE IT TO 2 FOLLOWING REORDS

SELECT \*, CASE WHEN TEMPERATURE <= 0 AND

    LEAD (TEMPERATURE) < 0 AND

    LEAD (TEMPERATURE,2) < 0  THEN

CITY ELSE NULL END

FROM TABLE

select id, city, temperature, day

from (

    select \*,

        case when temperature < 0

              and lead(temperature) over(order by day) < 0

              and lead(temperature,2) over(order by day) < 0

        then 'Y'

        when temperature < 0

              and lead(temperature) over(order by day) < 0

              and lag(temperature) over(order by day) < 0

        then 'Y'

        when temperature < 0

              and lag(temperature) over(order by day) < 0

              and lag(temperature,2) over(order by day) < 0

        then 'Y'

        end as flag

    from weather) x

where x.flag = 'Y';

**8. From the following 3 tables (event\_category, physician\_speciality, patient\_treatment), write a SQL query to get the histogram of specialties of the unique physicians who have done the procedures but never did prescribe anything.**

STEP 1 :

SELECT PS.SPECIALITY, COUNT(DISTINCT PS.SPECIALITY)

FROM PATEINT\_TREATEMENT PT JOIN EVENT\_CATEGORY EC

ON PT.EVENT\_NAME = EC.EVENT\_NAME

JOIN PHYSIAN\_SPECIALITY PS ON PT.PHYSIAN\_ID = PS.PHYSIAN\_ID

WHERE EC.CATEGORY = ‘PROCEDURE

**9. Find the top 2 accounts with the maximum number of unique patients on a monthly basis.**

STEP 1 : ARRANGE THE ID ON THE BASIS OF MONTH AND USE COUNT TO COUNT THE NO OF UNIQUE PATIENT

SELECT MONTH(DATE), ID, COUNT( DISTINCT PATEINT\_ID)

FROM TABLE

GROUP BY 1,2

LIMIT 2;

select a.month, a.account\_id, a.no\_of\_unique\_patients

from (

select x.month, x.account\_id, no\_of\_unique\_patients,

row\_number() over (partition by x.month order by x.no\_of\_unique\_patients desc) as rn

from (

select pl.month, pl.account\_id, count(1) as no\_of\_unique\_patients

from (select distinct to\_char(date,'month') as month, account\_id, patient\_id

from patient\_logs) pl

group by pl.month, pl.account\_id) x

     ) a

where a.rn < 3;

**10. SQL Query to fetch “N” consecutive records from a table based on a certain condition**

Note: Write separate queries to satisfy following scenarios:

10a. when the table has a primary key

10b. When table does not have a primary key

10c. Query logic based on data field