1. Find department wise minimum salary empname and maximum salary empname .

Source:

|  |  |  |
| --- | --- | --- |
| EMP\_NAME | DEPT\_ID | SALARY |
| Siva | 1 | 30000 |
| Ravi | 2 | 40000 |
| Prasad | 1 | 50000 |
| Sai | 2 | 20000 |
| Anna | 2 | 10000 |

Target:

|  |  |  |
| --- | --- | --- |
| DEPT\_ID | MIN\_SAL\_EMP\_NAME | MAX\_SAL\_EMP\_NAME |
| 1 | Siva | Prasad |
| 2 | Anna | Ravi |

**Ans:**

with cte\_min\_sal as (

select EMP\_NAME,DEPT\_ID,SALARY,row\_number() over(partition by DEPT\_ID order by SALARY ) as min\_sal\_rno from emps\_tbl

),

cte\_max\_sal as (

select EMP\_NAME,DEPT\_ID,SALARY,row\_number() over(partition by DEPT\_ID order by SALARY DESC) as max\_sal\_rno from emps\_tbl

)

select a.DEPT\_ID,a.EMP\_NAME as MIN\_SAL\_EMP,b.EMP\_NAME as MAX\_SAL\_EMP from cte\_min\_sal a join cte\_max\_sal b on a.dept\_id=b.dept\_id

and min\_sal\_rno=1 and max\_sal\_rno=1

1. **Mask first 12 digits of card number. (** [**https://www.youtube.com/watch?v=sU--ZZabrMM**](https://www.youtube.com/watch?v=sU--ZZabrMM) **)**

**Source :**

|  |
| --- |
| CARD\_NUMBER |
| 1234567812345670 |
| 2345678923456780 |
| 3456789034567890 |

**Target:**

|  |  |
| --- | --- |
| CARD\_NUMBER | NEW\_CARD\_NUMBER |
| 1234567812345670 | \*\*\*\*\*\*\*\*\*\*\*\*5678 |
| 2345678923456780 | \*\*\*\*\*\*\*\*\*\*\*\*6789 |
| 3456789034567890 | \*\*\*\*\*\*\*\*\*\*\*\*7890 |

**Ans:**

select CARD\_NUMBER,repeat('\*',12)||RIGHT(CARD\_NUMBER,4) as new\_card\_number from cards;

1. Select employee names with same salary.

( <https://www.youtube.com/watch?v=sU--ZZabrMM> )

Source:

|  |  |  |
| --- | --- | --- |
| EMPLOYEE\_ID | ENAME | SALARY |
| 3 | Bob | 60000 |
| 4 | Diana | 70000 |
| 5 | Eve | 60000 |
| 6 | Frank | 80000 |
| 7 | Grace | 70000 |
| 8 | Henry | 90000 |

Target:

|  |  |
| --- | --- |
| ENAME | SALARY |
| Bob | 60000 |
| Eve | 60000 |
| Diana | 70000 |
| Grace | 70000 |

**Ans:**

**Approach: 1**

with cte as (

select \*,count(\*) over(partition by SALARY) as s\_cnt from employee\_3

)

select ename,salary from cte where s\_cnt>1 order by salary

1. Imagine Zomato is encountered an issue where each order’s item is mistakenly swapped with next one. Our task as a data analysts is to fix this error and ensure that every order Id is paired with the correct item.

Also, if the last item has an odd order ID, it should remain as the last item in the corrected data.

**(** [**https://www.youtube.com/watch?v=0dCqt4jpMCs**](https://www.youtube.com/watch?v=0dCqt4jpMCs) **)**

**Source :**

|  |  |
| --- | --- |
| ORDER\_ID | ITEM |
| 1 | Chow Mein |
| 2 | Pizza |
| 3 | Veg Nuggets |
| 4 | Paneer Butter Masala |
| 5 | Spring Rolls |
| 6 | Veg Burger |
| 7 | Paneer Tikka |

Target :

|  |  |
| --- | --- |
| ORDER\_ID | ITEM |
| 1 | Pizza |
| 2 | Chow Mein |
| 3 | Paneer Butter Masala |
| 4 | Veg Nuggets |
| 5 | Veg Burger |
| 6 | Spring Rolls |
| 7 | Paneer Tikka |

Ans:

with cte as (

select \* from z\_orders where order\_id not in (select max(order\_id) from z\_orders)

),

cte2 as (

select a.order\_id,b.item from cte a join cte b on a.order\_id-1=b.order\_id and mod(a.order\_id,2)=0 and mod(b.order\_id,2)=1

union

select a.order\_id,b.item from cte a join cte b on a.order\_id=b.order\_id-1 and mod(a.order\_id,2)=1 and mod(b.order\_id,2)=0

union

select order\_id,item from (

select \*,mod(order\_id,2) as l\_row from z\_orders where order\_id in (select max(order\_id) from z\_orders)

) where L\_row=1

)

select \* from cte2 order by order\_id

**Approach:2**

with cte\_rcnt as (

select count(\*) as tot\_recs from z\_orders

),cte2 as (

select order\_id,item,tot\_recs,mod(order\_id,2) as mval,

case

when mod(cast(order\_id as int),2)=1 and order\_id<tot\_recs then cast(order\_id as int)+1

when mod(cast(order\_id as int),2)=1 and order\_id=tot\_recs then order\_id

when mod(cast(order\_id as int),2)=0 and order\_id<tot\_recs then cast(ORDER\_ID as int)-1 END

as NEW\_ORDER\_ID

from cte\_rcnt a cross join z\_orders

)

select NEW\_ORDER\_ID,ITEM from cte2 order by NEW\_ORDER\_ID

1. **find the top two highest-selling products within each category based on total spending.**

**(** [**https://www.youtube.com/watch?v=4KPPZLg\_QPA**](https://www.youtube.com/watch?v=4KPPZLg_QPA) **)**

**source:**

|  |  |  |  |
| --- | --- | --- | --- |
| CATEGORY | PRODUCT | USER\_ID | SPEND |
| appliance | refrigerator | 165 | 26 |
| appliance | refrigerator | 123 | 3 |
| appliance | washing machine | 123 | 19.8 |
| electronics | vacuum | 178 | 5 |
| electronics | wireless headset | 156 | 7 |
| electronics | vacuum | 145 | 15 |
| electronics | laptop | 114 | 999.99 |
| fashion | dress | 117 | 49.99 |
| groceries | milk | 243 | 2.99 |
| groceries | bread | 645 | 1.99 |
| home | furniture | 276 | 599.99 |
| home | decor | 456 | 29.99 |

Target:

|  |  |  |
| --- | --- | --- |
| CATEGORY | PRODUCT | TOT\_SPEND |
| appliance | refrigerator | 29 |
| appliance | washing machine | 19.8 |
| electronics | laptop | 999.99 |
| electronics | vacuum | 20 |
| fashion | dress | 49.99 |
| groceries | milk | 2.99 |
| groceries | bread | 1.99 |
| home | furniture | 599.99 |
| home | decor | 29.99 |

Ans:

with cte as (

select category,product,sum(spend) as TOT\_SPEND from ProductSpend group by category, product

), cte1 as (

select category, product, TOT\_SPEND, RANK() OVER(PARTITION by category order by TOT\_SPEND desc) as rnk

from cte )

select \* from cte1 where rnk <=2 order by category

**6) Find the customers who have purchased at least one product from every product category available.**

( <https://www.youtube.com/watch?v=yGww6dmMR0Q> )

Customer\_contracts

|  |  |  |
| --- | --- | --- |
| **CUSTOMER\_ID** | **PRODUCT\_ID** | **AMOUNT** |
| 1 | 1 | 1000 |
| 2 | 2 | 2000 |
| 3 | 1 | 1100 |
| 4 | 1 | 1000 |
| 7 | 1 | 1000 |
| 7 | 3 | 400 |
| 6 | 4 | 2000 |
| 1 | 5 | 1500 |
| 2 | 5 | 2000 |
| 4 | 5 | 2200 |
| 7 | 6 | 5000 |
| 1 | 2 | 2000 |

Products

|  |  |  |
| --- | --- | --- |
| **PRODUCT\_ID** | **PRODUCT\_CATEGORY** | **PRODUCT\_NAME** |
| 1 | Analytics | Azure Databricks |
| 2 | Analytics | Azure Stream Analytics |
| 3 | Containers | Azure Kubernetes Service |
| 4 | Containers | Azure Service Fabric |
| 5 | Compute | Virtual Machines |
| 6 | Compute | Azure Functions |

Ans:

with cte as(

select cc.customer\_id,cc.product\_id,p.product\_category

from customer\_contracts cc join products p on cc.product\_id=p.product\_id order by customer\_id, product\_id

)

select customer\_id,count(distinct PRODUCT\_CATEGORY) p\_cnt from cte group by customer\_id

having p\_cnt = (select count(distinct PRODUCT\_CATEGORY) from products)

1. write a SQL query to identify sellers who did not make any sales in 2020.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table: ORDERS |  |  |  |  |
| **ORDER\_ID** | **SALE\_DATE** | **ORDER\_COST** | **CUSTOMER\_ID** | **SELLER\_ID** |
| 1 | 01-03-2020 | 1500 | 101 | 1 |
| 2 | 25-05-2020 | 2400 | 102 | 2 |
| 3 | 25-05-2019 | 800 | 101 | 3 |
| 4 | 13-09-2020 | 1000 | 103 | 2 |
| 5 | 11-02-2019 | 700 | 101 | 2 |

|  |  |
| --- | --- |
| Table: Sellers |  |
| SELLER\_ID | SELLER\_NAME |
| 1 | Daniel |
| 2 | Ben |
| 3 | Frank |

Ans:

Approach-1

select s.seller\_name from sellers s left join orders o on s.seller\_id=o.seller\_id

and date\_trunc(YEAR,o.sale\_date)='2020-01-01' where order\_id is null

|  |
| --- |
| SELLER\_NAME |
| Frank |

Approach-2

select \* from sellers where seller\_id not in (

select distinct seller\_id from orders where date\_trunc(YEAR,SALE\_DATE)='2020-01-01')

1. Write the query to find the matching combinations.( <https://www.youtube.com/watch?v=jERnTLuQFAQ>)

Source:

|  |
| --- |
| **TEAM\_NAME** |
| CSK |
| KKR |
| GT |
| DC |
| LSG |

Target:

|  |
| --- |
| **MATCHES** |
| CSK vs KKR |
| CSK vs GT |
| CSK vs DC |
| CSK vs LSG |
| KKR vs LSG |
| GT vs KKR |
| GT vs LSG |
| DC vs KKR |
| DC vs GT |
| DC vs LSG |

Ans:

select t1.team\_name||' vs '||t2.team\_name as MATCHES from teams t1

join

teams t2 on t1.team\_name < t2.team\_name

1. How will you re-write the below query with help of the snowflake syntax SELECT \* FROM ( SELECT i, p, o, ROW\_NUMBER() OVER (PARTITION BY p ORDER BY o) AS row\_num FROM qt ) WHERE row\_num = 1

ANS:

Approach : 1

SELECT i, p, o FROM qt QUALIFY ROW\_NUMBER() OVER (PARTITION BY p ORDER BY o) = 1

Approach: 2

SELECT i, p, o, ROW\_NUMBER() OVER (PARTITION BY p ORDER BY o) AS row\_num FROM qt QUALIFY row\_num = 1

select \* from PLICY at (offset=> 18\*60\*60)

create DATABASE DB\_BKP CLONE

with cte as

(

select EMP\_ID,REPLACE(EMAIL\_ID,'@@','@') as NEW\_EMAIL\_ID from EMPLOYEE

)

UPADTE EMPLOYEE SET EMAIL\_ID=CTE.NEW\_EMAIL\_ID where EMPLOYEE.EMP\_ID=CTE.EMP\_ID;

MERGE INTO EMP\_T tgt USING EMP\_S src

ON src.emp\_id=tgt.emp\_id

when matched

update set tgt.salary=src.salary where src.emp\_id=tgt.emp\_id

;

1. select \* from a left join b on a.id =b.a\_id and b.customer\_name ='ABC'

***Note: this ‘AND’ will function with the join. So , it returns all records from left side table* and only the matching records from right table.**

2. select \* from a left join b on a.id =b.a\_id where b.customer\_name ='ABC'

**Note: This will join tables a and b and generate a set of results. Then filter out the records as based on the conditions in where clause.**

1. select \* from a inner join b on a.id =b.a\_id and b.customer\_name ='ABC'

2. select \* from a inner join b on a.id =b.a\_id where b.customer\_name ='ABC'

|  |  |  |  |
| --- | --- | --- | --- |
| T1 |  |  |  |
| C1 | C2 |  |  |
| 1 | 2 |  |  |
| 2 | 3 |  |  |
| 3 | 4 |  |  |
|  |  |  |  |
| T2 |  |  |  |
| C1 | C2 |  |  |
| 1 | 2 |  |  |
| 2 | 3 |  |  |
| 0 | 1 |  |  |
|  |  |  |  |
|  |  |  |  |
| Select T1.\* from T1 Left join T2 | | |  |
| on T1.C1 = T2.C1 | |  |  |
| and T1.C2 < T2.C2 | |  |  |
|  |  |  |  |
| C1 | C2 | C1 | C2 |
| 1 | 2 |  |  |
| 2 | 3 |  |  |
| 3 | 4 |  |  |

|  |  |  |
| --- | --- | --- |
| T1 <-Table | |  |
| C1 <- Column | |  |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| T2 |  |  |
| C1 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| Select T1.\* from T1 Left join T2 | | |
| on T1.C1 = T2.C1 | |  |
| C1 <- Column | |  |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
|  |  |  |