3. Database Question (SQL Query)

Consider two tables Customers and CustomerOrders as shown below:

Customers

Id	Name
1	A
2	B
3	C

CustomerOrders

Id	Date	Qty
1	2014-01-13	10
1	2014-01-11	15
1	2014-01-12	20
2	2014-01-06	30
2	2014-01-08	40

Write a SQL query to extract the value of customers with most recent top 2 orders. The output should be as below:

Id	Name	Date	Qty
1	A	2014-01-13	10
1	A	2014-01-12	20
2	B	2014-01-08	40
2	B	2014-01-06	30

Describe your query.

Answer: Assuming that the CustomerOrders table is huge, I would index the Date Column and then run a Window function in SQL to get the desired output in a single query. This returns every customer with their most recent 2 orders. (The question is maybe asking only for 2 customers, I think, which I've done on the next query)

```
SELECT JOIN.id,
       JOIN.NAME ,
       JOIN.date ,
       JOIN.qty
       (
                  SELECT    cust.id ,
                            cust.NAME ,
                            custord.date ,
                            custard.qty ,
```

```
                        dense_rank() OVER (partition BY custord.id
ORDER BY custord.date DESC) AS d_rank
                FROM    customers cust
                LEFT JOIN customerorders custord
                ON      cust.id = custord.id ) AS
JOIN
WHERE  d_rank <=2
```

Answer 2: Following query only gives the most recent 2 customers, not all customers. I believe the following might be faster than using window functions on the whole table in this case, (if we index Date column)

```
SELECT final.id ,
       final.NAME ,
       final.date ,
       final.qty
FROM   (
                SELECT     cust_max.id,
                           cust_max.NAME,
                           custord.date,
                           custord.qty ,
                           Dense_rank() OVER (partition BY
cust_max.id ORDER BY custord.date DESC) AS d_rank
                FROM       customerorders
AS custord
                INNER JOIN
                           (
                                    SELECT   id,
                                             Max(date) AS max_date
                                    FROM     customerorders
                                    GROUP BY id,
                                    order by max_date
```

```
                                                    /* max(Date)*/
                                                    limit 2) AS cust_max
                        ON          custord.id = cust_max.id ) AS final
WHERE  final.d_rank <=2
```

Explaining query 2 from the inner query: Innermost query (cust_max) gets the Id of 2 customers who have the latest purchase. (could have stored just the Id in a table variable as well).

These 2 Ids are inner-joined with CustomerOrders table to get the order date and quantity, for only these 2 customers. Also, window function RANK() is used to get just the 2 latest records for these customers. Could use CTE as well, But I believe the performance will be similar.

4. You are given a project to track, record, maintain and visualize the data for Realtime air quality dataset.

Data:
https://opendatanepal.com/dataset/c3eff9e4-7783-4904-9e10-b3820b30041c/resource/f715980c-0897-4899-a9a5-fac8ca05122d/download/ratnapark.csv

a) Consume the data, prepare table(s) for storage, store the data.

b) Visualize the data using appropriately.

c) Explain 3Vs with respect to the above data.

Answer: Please find Colab notebook here: Colab notebook

5. You have an existing system that you have to **scale up data** in terms of both efficiency for existing clients and be capable to handle 100X number of clients. What are the steps that you would take?

Answer: I would do the following in the preferred order. Order could be re-ordered based on the number of clients and number and nature of queries to the database

1. *Index the most queried columns, Use the automated or manual cached-queries functionality. Try to use as much database optimization functionality provided in the software.*
2. *Run automated ETL processes during off-business times (e.g. at Night), to create a group of tables that are consumed by a specific team of users. This reduces traffic in the same database or server. Also low risk of somebody modifying the main table.*
3. *Normalize the tables, and use views to join these tables if necessary.*
4. *Establish a data collection and maintaining team, that is responsible for the technical aspects of collecting all data in a single system if possible. Any client looking for data relevant to them, contacts this team.*
5. *Increase Processing, RAM or storage of the server.*
6. *Buy a cloud provider's pay as you go server, if it saves resources and capital.*