**Sr. Data Engineer Technical Interview**

**Coding Questions (60 minutes):**

1. Write a solution to report the customer ids from the Customer table that bought all the products in the Product table.

Table: Customer

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| customer\_id | int |

| product\_key | int |

+-------------+---------+

This table may contain duplicates rows.

customer\_id is not NULL.

product\_key is a foreign key (reference column) to Product table.

Table: Product

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| product\_key | int |

+-------------+---------+

product\_key is the primary key (column with unique values) for this table.

 Return the result table in **any order**.

The result format is in the following example.

**Example:**

**Input:**

Customer table:

+-------------+-------------+

| customer\_id | product\_key |

+-------------+-------------+

| 1 | 5 |

| 2 | 6 |

| 3 | 5 |

| 3 | 6 |

| 1 | 6 |

+-------------+-------------+

Product table:

+-------------+

| product\_key |

+-------------+

| 5 |

| 6 |

+-------------+

**Output:**

+-------------+

| customer\_id |

+-------------+

| 1 |

| 3 |

+-------------+

**Explanation:**

The customers who bought all the products (5 and 6) are customers with IDs 1 and 3.

**Answer**

select

customer\_id

from

Customer A

group by 1

having

count(distinct A.product\_key) = (select count(distinct product\_key) from Product)

1. Assume you're given a table containing data on Amazon customers and their spending on products in different category, write a query to identify the top two highest-grossing products within each category in the year 2022. The output should include the category, product, and total spend.

**product\_spend - Table:**

| **Column Name** | **Type** |
| --- | --- |
| category | string |
| product | string |
| user\_id | integer |
| spend | decimal |
| transaction\_date | timestamp |

**product\_spend - Example Input:**

| **category** | **product** | **user\_id** | **spend** | **transaction\_date** |
| --- | --- | --- | --- | --- |
| appliance | refrigerator | 165 | 246.00 | 12/26/2021 12:00:00 |
| appliance | refrigerator | 123 | 299.99 | 03/02/2022 12:00:00 |
| appliance | washing machine | 123 | 219.80 | 03/02/2022 12:00:00 |
| electronics | vacuum | 178 | 152.00 | 04/05/2022 12:00:00 |
| electronics | wireless headset | 156 | 249.90 | 07/08/2022 12:00:00 |
| electronics | vacuum | 145 | 189.00 | 07/15/2022 12:00:00 |

**Example Output:**

| **category** | **product** | **total\_spend** |
| --- | --- | --- |
| appliance | refrigerator | 299.99 |
| appliance | washing machine | 219.80 |
| electronics | vacuum | 341.00 |
| electronics | wireless headset | 249.90 |

**Explanation:**

Within the "appliance" category, the top two highest-grossing products are "refrigerator" and "washing machine."

In the "electronics" category, the top two highest-grossing products are "vacuum" and "wireless headset."

The dataset you are querying against may have different input & output - **this is just an example**!

**Answer**

with cte as

(

select

appliance,

product,

sum(**spend) as “total\_spend”**

**from**

**product\_spend**

**where year(transaction\_date) = ‘2022’**

**group by 1,2**

**),**

**cte\_2 as**

**(**

**select**

appliance,

product,

**total\_spend,**

row\_number () over (partition by appliance order by **total\_spend desc) as “row\_num”**

**from**

**cte**

**)**

**select**

**appliance,**

**product,**

**total\_spend**

**from**

**cte\_2**

**where**

**row\_num <= 2**

1. Please run this on your local IDE or <https://colab.google/> using data:

<https://docs.google.com/spreadsheets/d/1bEoqOWbhL5kcX4_KvBBukpkmlsgxp-gr7t9EIcwxi6M/edit?usp=sharing>

Extract Data:

* You are given a CSV file containing genomic data. The file has the following columns: GeneID, Chromosome, StartPosition, EndPosition, Sequence, GeneName, and VariantType. The Sequence column contains genomic sequences, and the VariantType can be 'SNP', 'Deletion', or 'Insertion'.

Transform Data:

* The Sequence field contains long strings of genomic sequences. Write a pseudo-code or describe a process to identify and count the occurrence of a specific short sequence (e.g., 'ATCG') within each genomic sequence.
* Transform the StartPosition and EndPosition into a new field named Length which is the difference between the two.

Load Data:

* Load the data into SQL table. Mention any specific SQL data types you would use for each column. Considering performance, explain how you would index the SQL table(s).

**System Design Questions (30 mins):**

Imagine you are charged with the development and execution of an all-encompassing data management system for the research and product development division of a biotechnology firm. This system is to include two specialized pipelines:

* The first pipeline's main function is to manage the transfer of extensive genomic and clinical datasets vital for research and product development. The challenges associated with this type of data include: (1) datasets of considerable breadth, (2) many data sources, (3) unstructured and inconsistent data, and (4) numerous data processing pipelines. This data is crucial for supporting comprehensive biomarker discovery, which is highly computationally demanding, and for developing machine learning models suitable for clinical environments.
* The second pipeline is designated for the oversight of data produced by bespoke web applications. These datasets are relatively small in scale and exhibit a high degree of structure, in stark contrast to those handled by the first pipeline.

In your role, you will be responsible for the supervision of the entire data lifecycle, guaranteeing the efficient movement, storage, and accessibility of both data types. Your efforts will serve the company's overarching research and development goals.

1. Could you describe the criteria and considerations that would guide your selection of a data platform, particularly for the initial stages of managing large-scale genomic data? How would you address the need for scalability and efficiency in data handling and transformation to support research initiatives? Please identify specific tools and technologies you would recommend and provide your rationale.

Answer

1. How would you approach the modification of the data pipeline to accommodate integration with a web application handling smaller dataset? What strategies would you implement to ensure a smooth and uninterrupted data flow, and to facilitate the integration of the R&D pipeline with the web application's data needs?
2. Can you outline your strategy for upholding industry regulatory compliance within both the R&D and web application data environments? Additionally, how would you approach the task of optimizing performance and scalability given the varied data requirements and the differing technology stacks in each domain?

ans

**Conceptual Questions (30 mins):**

Write shorts answers.

* 1. Can you describe the concept of data lakes and how they differ from traditional data warehouses?

ans :

**data lake**

- raw unfiltered ingested data.

* less conditional joins more quicker compared to data warehouse.
* supportive of analytical queries

**data warehouse**

- more structured categorized data format.

- more defined fields, doesn’t have unstructured data.

* supportive of transactions
* supportive of scd
  1. What are the benefits and drawbacks of using microservices for data processing tasks?

benefits

* external microservices, it can quicken your task execution and reliable tested product.
* support documentation available.

drawbacks

* opened sourced system, then there could be data integrity issue.
* consistency of the package builds (especially on the communication part)
  1. How do you approach data partitioning and indexing in a large-scale distributed database system?
* partitioning should be done on the a column that maintains the consistency of the data across nodes.
* Repartitioning can be done if the cluster size is large enough to be split into small number of nodes to ensure that the data shuffling problem is minimized.
* Indexing on high cardinality column.
  1. What strategies would you employ to handle schema evolution in a database without causing downtime for applications that use the database?
* Maintain multiple versions of the database schema concurrently. This allows you to roll out changes gradually without disrupting applications. New versions of applications can be gradually migrated to use the latest schema version.
* Implement schema validation mechanisms to ensure that applications can handle changes to the schema gracefully. This may involve versioning APIs or providing error handling for unexpected schema changes.
* Monitor the performance and behavior of applications after deploying schema changes. Have a rollback plan in place in case of any unexpected issues, allowing you to quickly revert to the previous schema version if necessary.