#SQLPRACD1

SQL PRACTICE

DataLemur

Presented By: Subhajit Dutta

Q1. Given a table of candidates and their skills, you're tasked with finding the candidates best suited for an open Data Science job.
You want to find candidates who are proficient in Python, Tableau, and PostgreSQL.
Write a query to list the candidates who possess all of the required skills for the job.
Sort the output by candidate ID in ascending order.

Assumption: There are no duplicates in the candidate's table.

```
SELECT candidate_id FROM candidates

WHERE skill IN('Python','Tableau','PostgreSQL')

GROUP BY candidate_id

HAVING COUNT(skill)=3

ORDER BY candidate_id ASC;
```

Q2. Assume you are given the tables below about Facebook pages and page likes.
Write a query to return the page IDs of all the Facebook pages that don't have any likes.
The output should be in ascending order.

```
SELECT DISTINCT(p.page_id) FROM pages AS p
FULL OUTER JOIN page_likes AS pl
ON
  p.page_id=pl.page_id
WHERE pl.page_id IS NULL
ORDER BY p.page_id ASC;
```

Q3. Tesla is investigating production bottlenecks and they need your help to extract the relevant data. Write a query that determines which parts have initiated the assembly process but remain unfinished.

Assumptions:
parts_assembly table contains all parts currently in production, each at varying stages of the assembly process.
An unfinished part is one that lacks a finish_date.

```
SELECT DISTINCT(part) FROM parts_assembly

WHERE finish_date IS NULL;
```

Q4. Assume that you are given the table below-containing information on viewership by device type (where the three types are laptop, tablet, and phone). Define “mobile” as the sum of tablet and phone viewership numbers. Write a query to compare the viewership on laptops versus mobile devices.

Output the total viewership for laptop and mobile devices in the format of : "**laptop_views**" and "**mobile_views**".

```
SELECT
  SUM(CASE WHEN device_type='laptop' THEN 1 ELSE 0 END) AS laptop_views,
  SUM(CASE WHEN device_type IN ('tablet', 'phone') THEN 1 ELSE 0 END) AS mobile_views
FROM viewership;
```

Q5. Assume you are given the table below that shows job postings for all companies on the LinkedIn platform. Write a query to get the number of companies that have posted duplicate job listings.

Clarification:
Duplicate job listings refer to two jobs at the same company with the same title and description.

```
WITH CTE AS
(
SELECT
        company_id,  title,    description,    count(*)
FROM job_listings
GROUP BY company_id,title,description
HAVING COUNT(*)>1
ORDER BY company_id, title, description
)
SELECT
        COUNT(company_id) AS co_w_duplicate_jobs
FROM CTE;
```

Q6. Given a table of Facebook posts, for each user who posted at least twice in 2021, write a query to find the number of days between each user's first post of the year and the last post of the year in the year 2021. Output the user and number of the days between each user's first and last posts.

```
SELECT
        user_id,
        EXTRACT(DAYS FROM max(post_date)-min(post_date)) AS date_diff
FROM posts
WHERE
  post_date BETWEEN '01/01/2021' AND '12/31/2021'
GROUP BY user_id
HAVING EXTRACT(DAYS FROM max(post_date)-min(post_date))>0
ORDER BY user_id ASC;
```

Q7. Write a query to find the top 2 power users who sent the most messages on Microsoft Teams in August 2022. Display the IDs of these 2 users along with the total number of messages they sent. Output the results in descending count of the messages.

Assumption:
No two users have sent the same number of messages in August 2022.

```
SELECT
        sender_id,
        COUNT(*) AS message_count
FROM messages
WHERE sent_date BETWEEN '08/01/2022' AND '08/31/2022'
GROUP BY sender_id
ORDER BY COUNT(*) DESC
LIMIT 2;
```

Q8. You are given the tables below-containing information on Robinhood trades and users. Write a query to list the top three cities that have the most completed trade orders in descending order.

Output the city and number of orders.

```
SELECT
        city,
        COUNT(*) AS TOTAL_ORDERS
FROM trades AS t
INNER JOIN users AS u
ON u.user_id = t.user_id
WHERE STATUS='Completed'
GROUP BY city
ORDER BY COUNT(*) DESC
LIMIT 3;
```

Q9. Given the reviews table, write a query to get the average stars for each product every month.

The output should include the month in numerical value, product id, and average star rating rounded to two decimal places. Sort the output based on the month followed by the product id.

```
SELECT
        EXTRACT(MONTH FROM submit_date),
        PRODUCT_ID,ROUND(AVG( stars ),2) AS avg_stars
FROM reviews
GROUP BY product_id, EXTRACT(MONTH FROM submit_date )
ORDER BY EXTRACT(MONTH FROM submit_date ), product_id;
```

Q10. Assume you have an events table on app analytics. Write a query to get the app’s click-through rate (CTR %) in 2022. Output the results in percentages rounded to 2 decimal places.

Notes:

Percentage of click-through rate = 100.0 * Number of clicks / Number of impressions
To avoid integer division, you should multiply the click-through rate by 100.0, not 100.

```
SELECT
        app_id,
        ROUND(SUM(CASE WHEN event_type='click' THEN 1 ELSE 0 END)*100.0/
        SUM(CASE WHEN event_type='impression' THEN 1 ELSE 0 END),2) AS CTR
FROM events
WHERE timestamp BETWEEN '01/01/2022' AND '12/31/2022'
GROUP BY APP_ID;
```

Q11. New TikTok users sign up with their emails and each user receives a text confirmation to activate their account. Assume you are given the below tables about emails and texts.
Write a query to display the ids of the users who did not confirm on the first day of sign-up, but confirmed on the second day.

Assumption:

action_date is the date when the user activated their account and confirmed their sign-up through the text.

```
SELECT
        user_id
FROM emails E
JOIN texts AS T
ON
 T.email_id=E.email_id AND
 signup_date!=action_date
WHERE EXTRACT(DAYS FROM action_date - signup_date)=1;
```

Q12. Your team at JPMorgan Chase is soon launching a new credit card, and to gain some context, you are analyzing how many credit cards were issued each month.

Write a query that outputs the name of each credit card and the difference in issued amount between the month with the most cards issued, and the least cards issued. Order the results according to the biggest difference.

```
SELECT
        card_name,
        MAX(issued_amount)-MIN(issued_amount) AS DIFFERENCE
FROM monthly_cards_issued
GROUP BY card_name
ORDER BY DIFFERENCE DESC;
```

Q13. You are trying to find the mean number of items bought per order on Alibaba, rounded to 1 decimal place.

However, instead of doing analytics on all Alibaba orders, you have access to a summary table, which describes how many items were in an order (item_count), and the number of orders that had that many items (order_occurrences).

```
SELECT
        ROUND(SUM(item_count::DECIMAL*order_occurrences)/
        SUM(order_occurrences),1) AS AVG_COUNT
FROM items_per_order;
```

Q14.CVS Health is trying to better understand its pharmacy sales, and how well different products are selling. Each drug can only be produced by one manufacturer.

Write a query to find the top 3 most profitable drugs sold, and how much profit they made. Assume that there are no ties in the profits. Display the result from the highest to the lowest total profit.

Definition:

cogs stands for **Cost of Goods Sold** which is the direct cost associated with producing the drug.

Total Profit = Total Sales - Cost of Goods Sold

```
SELECT
  drug,
  total_sales-cogs AS PROFIT
FROM pharmacy_sales
ORDER BY PROFIT DESC
LIMIT 3;
```

Q15. CVS Health is trying to better understand its pharmacy sales, and how well different products are selling. Each drug can only be produced by one manufacturer.

Write a query to find out which manufacturer is associated with the drugs that were not profitable and how much money CVS lost on these drugs.

Output the manufacturer, number of drugs and total losses. Total losses should be in absolute value. Display the results with the highest losses on top.

```
SELECT
  manufacturer,
  COUNT(*) AS DRUG_COUNT,
  SUM(ABS(total_sales-cogs)) AS DIFFERENCE
FROM pharmacy_sales
WHERE total_sales-cogs<=0
GROUP BY manufacturer
ORDER BY DIFFERENCE DESC;
```

Q16. CVS Health is trying to better understand its pharmacy sales, and how well different products are selling.

Write a query to find the total drug sales for each manufacturer. Round your answer to the closest million, and report your results in descending order of total sales.

Because this data is being directly fed into a dashboard which is being seen by business stakeholders, format your result like this: "$36 million".

```
SELECT
  manufacturer,
  CONCAT('$',CONCAT(ROUND((SUM(total_sales)/1000000),0),' million'))
FROM pharmacy_sales
GROUP BY manufacturer
ORDER BY SUM(total_sales) DESC;
```

Q.17 UnitedHealth has a program called Advocate4Me, which allows members to call an advocate and receive support for their health care needs – whether that's behavioural, clinical, well-being, health care financing, benefits, claims or pharmacy help.

Write a query to find how many UHG members made 3 or more calls. case_id column uniquely identifies each call made.

```
WITH CTE AS(
SELECT
  policy_holder_id,
  COUNT(case_id) AS TOTAL_CALL
FROM callers
GROUP BY policy_holder_id
ORDER BY TOTAL_CALL DESC)
SELECT COUNT(*) AS MEMBER_COUNT FROM CTE
WHERE TOTAL_CALL>=3;
```

Q18. UnitedHealth Group has a program called Advocate4Me, which allows members to call an advocate and receive support for their health care needs – whether that's behavioral, clinical, well-being, health care financing, benefits, claims, or pharmacy help.

Calls to the Advocate4Me call center are categorized, but sometimes they can't fit neatly into a category. These uncategorized calls are labeled "n/a", or are just empty (when a support agent enters nothing into the category field).

Write a query to find the percentage of calls that cannot be categorized. Round your answer to 1 decimal place.

```
SELECT
  ROUND(SUM(CASE WHEN call_category IS NULL OR call_category='n/a' THEN 1 ELSE 0 END)*100.0/
  COUNT(*),1)
FROM CALLERS;
```

#SQLPRACD1

THANK YOU!!!

DataLemur

Presented By: Subhajit Dutta