## SQL Tasks:

1. Design a database schema to store experiment data (e.g., page variants, impressions, clicks, timestamps).

CREATE TABLE experiment\_data (

id INT PRIMARY KEY AUTOINCREMENT,

variant VARCHAR(10) NOT NULL,

impressions INT DEFAULT 0,

clicks INT DEFAULT 0,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

--------------------------------------------------------------------------------------------------

2. Write SQL queries to insert new experiment data.

INSERT INTO experiment\_data (variant, impressions, clicks)

VALUES ('A', 100, 10), ('B', 120, 12);

* This query inserts the experiment data for two variants, ‘A’ and ‘B’, with the respective number of impressions and clicks. It assumes we're recording data for multiple variants, which may change over time.

--------------------------------------------------------------------------------------------------

3. Implement SQL queries to calculate key metrics (e.g., CTR, confidence intervals) for each variant.

SELECT variant,

SUM(clicks) AS total\_clicks,

SUM(impressions) AS total\_impressions,

(SUM(clicks) \* 100.0 / NULLIF(SUM(impressions), 0)) AS ctr

FROM experiment\_data

GROUP BY variant;

 **CTR (Click-through Rate):** This is calculated by dividing the total clicks by the total impressions and multiplying by 100 to get the percentage.

 The NULLIF function ensures we don't divide by zero when there are no impressions.

 The SUM function aggregates clicks and impressions across the dataset per variant.

--------------------------------------------------------------------------------------------------

4. Use SQL to retrieve and aggregate historical data for the Multi-Armed Bandit algorithm.

SELECT variant,

SUM(impressions) AS total\_impressions,

SUM(clicks) AS total\_clicks,

(SUM(clicks) \* 100.0 / NULLIF(SUM(impressions), 0)) AS ctr

FROM experiment\_data

WHERE timestamp >= NOW() - INTERVAL '7 days'

GROUP BY variant;

This query calculates the metrics for each variant within the last 7 days, providing historical data for a Multi-Armed Bandit algorithm. It focuses on recent data to adjust weights dynamically based on performance.

--------------------------------------------------------------------------------------------

## Bonus:

- Add support for multiple variants (beyond just A/B testing) in both the database schema and queries.

SELECT

variant,

(SUM(clicks) \* 1.0 / NULLIF(SUM(impressions), 0)) AS ctr,

(SUM(clicks) \* 1.0 / NULLIF(SUM(impressions), 0) + SQRT(2 \* LN(SUM(impressions)) / NULLIF(SUM(impressions), 1))) AS ucb\_score

FROM experiment\_data

GROUP BY variant

ORDER BY ucb\_score DESC;

 **CTR**: This is the same calculation for click-through rate as before.

 **UCB Score (Upper Confidence Bound)**: This is an implementation of the Multi-Armed Bandit algorithm where we calculate the ucb\_score for each variant. This score helps in determining which variant to explore next based on its performance and the uncertainty in the impressions data (using logarithmic scaling and square root for exploration).

 This query works for multiple variants and orders them by the UCB score to prioritize variants that are performing better or have higher uncertainty.

-------------------------------------------------------------------------------------------------------