Q1

SELECT

DATE\_FORMAT(purchase\_time, '%Y-%m') AS month,

COUNT(\*) AS purchase\_count

FROM

transactions

WHERE

refunded = 0

GROUP BY

DATE\_FORMAT(purchase\_time, '%Y-%m');

**Explanation:**

* We use DATE\_FORMAT to extract the year and month from the purchase\_time column.
* We filter out refunded purchases by checking refunded = 0.
* We group by the month and count the number of purchases.

Q2

SELECT

store\_id,

COUNT(\*) AS order\_count

FROM

transactions

WHERE

DATE\_FORMAT(purchase\_time, '%Y-%m') = '2020-10'

GROUP BY

store\_id

HAVING

COUNT(\*) >= 5;

**Explanation:**

* We filter transactions that occurred in October 2020 using DATE\_FORMAT.
* We group by store\_id and count the number of orders.
* We use HAVING to filter stores that have at least 5 orders.

Q3

SELECT

store\_id,

MIN(TIMESTAMPDIFF(MINUTE, purchase\_time, refund\_time)) AS shortest\_interval\_min

FROM

transactions

WHERE

refunded = 1

GROUP BY

store\_id;

**Explanation:**

* We calculate the time difference in minutes between purchase\_time and refund\_time using TIMESTAMPDIFF.
* We filter only refunded transactions (refunded = 1).
* We group by store\_id and find the minimum interval.

Q4

WITH FirstOrder AS (

SELECT

store\_id,

MIN(purchase\_time) AS first\_purchase\_time

FROM

transactions

GROUP BY

store\_id

)

SELECT

t.store\_id,

t.gross\_transaction\_value

FROM

transactions t

JOIN

FirstOrder fo

ON

t.store\_id = fo.store\_id

AND t.purchase\_time = fo.first\_purchase\_time;

**Explanation:**

* We use a CTE (FirstOrder) to find the first purchase time for each store.
* We join the transactions table with the CTE to get the gross\_transaction\_value of the first order.

Q5WITH FirstPurchase AS (

SELECT

buyer\_id,

MIN(purchase\_time) AS first\_purchase\_time

FROM

transactions

GROUP BY

buyer\_id

)

SELECT

i.item\_name,

COUNT(\*) AS order\_count

FROM

transactions t

JOIN

FirstPurchase fp

ON

t.buyer\_id = fp.buyer\_id

AND t.purchase\_time = fp.first\_purchase\_time

JOIN

items i

ON

t.item\_id = i.item\_id

GROUP BY

i.item\_name

ORDER BY

order\_count DESC

LIMIT 1;

**Explanation:**

* We use a CTE (FirstPurchase) to find the first purchase time for each buyer.
* We join the transactions table with the CTE and the items table to get the item name.
* We group by item\_name and count the number of orders, then order by the count in descending order to get the most popular item.

Q6

ALTER TABLE transactions

ADD COLUMN refund\_processable BOOLEAN;

UPDATE transactions

SET refund\_processable = CASE

WHEN TIMESTAMPDIFF(HOUR, purchase\_time, refund\_time) <= 72 THEN 1

ELSE 0

END

WHERE refunded = 1;

**Explanation:**

* We add a new column refund\_processable to the transactions table.
* We update the column based on the condition that the refund must happen within 72 hours of the purchase time.
* Q7
* WITH RankedPurchases AS (
* SELECT
* buyer\_id,
* purchase\_time,
* ROW\_NUMBER() OVER (PARTITION BY buyer\_id ORDER BY purchase\_time) AS purchase\_rank
* FROM
* transactions
* WHERE
* refunded = 0
* )
* SELECT
* buyer\_id,
* purchase\_time
* FROM
* RankedPurchases
* WHERE
* purchase\_rank = 2;
* **Explanation:**
* We use a CTE (RankedPurchases) to rank purchases for each buyer using ROW\_NUMBER.
* We filter out refunded transactions (refunded = 0).
* We select only the second purchase (purchase\_rank = 2).
* Q8
* WITH RankedTransactions AS (
* SELECT
* buyer\_id,
* purchase\_time,
* ROW\_NUMBER() OVER (PARTITION BY buyer\_id ORDER BY purchase\_time) AS transaction\_rank
* FROM
* transactions
* )
* SELECT
* buyer\_id,
* purchase\_time
* FROM
* RankedTransactions
* WHERE
* transaction\_rank = 2;
* **Explanation:**
* We use a CTE (RankedTransactions) to rank transactions for each buyer using ROW\_NUMBER.
* We select only the second transaction (transaction\_rank = 2).