```
/* 1.Daily Order count and Total amount */
SELECT
 DATE1 AS ORDER_DATE,
 COUNT(*) AS CNT,
 SUM(ORDERAMOUNT) AS TOTAL_AMOUNT
FROM
 ORDERS
GROUP BY
 ORDER_DATE;

/* 2 Daily Order count and order amount of cancelled orders */
SELECT
 O.DATE1 AS DATE,
 COUNT(ID) AS CANCELLED_ORDER_CNT,
 SUM(QUANTITY * UNITPRICE) AS CANCELLED_AMOUNT
FROM
 ORDERS AS O
 JOIN ORDER_ITEMS AS OI ON O.ID = OI.ORDERID
WHERE
 STATUS = 'Cancelled'
GROUP BY
 DATE;

/*3 payment method distribution */
UPDATE ORDERS
SET
 PAYMENT_METHOD = 'phonepe'
```

```
WHERE
 PAYMENT_METHOD = 'phonpe'
SELECT
 PAYMENT_METHOD,
 COUNT(*) AS ORDERS,
 SUM(ORDERAMOUNT)
FROM
 ORDERS
GROUP BY
 PAYMENT_METHOD;

/* 4 How many customers are from Bangalore */
UPDATE CUSTOMER
SET
 CITY = 'Bangalore'
WHERE
 CITY = 'Banglore'
SELECT
 COUNT(*) AS CUSTOMERS_FROM_BANGALORE
FROM
 CUSTOMER
WHERE
 CITY = 'Bangalore';

/* 5 How Many Orders Came From pune */
SELECT
 COUNT(*) AS ORDERS_FROM_PUNE
```

```
FROM
 ORDERS AS O
 JOIN CUSTOMER AS C ON O.CUSTOMERID = C.ID
WHERE
 C.CITY = 'Pune';

/*6 when and fromwhich city was the first order for each category has been placed ?
no order for any category then display it as NA */
SELECT
 I.CATEGORY AS CATEGORY,
 COALESCE(MIN(O.DATE1)::TEXT, 'N/A') AS FIRST_ORDER_DATE,
 COALESCE(C.CITY, 'N/A') ORDER_FROM
FROM
 ORDER_ITEMS AS OI
 RIGHT JOIN ITEMS AS I ON OI.ITEMID = I.ID
 LEFT JOIN ORDERS AS O ON OI.ORDERID = O.ID
 LEFT JOIN CUSTOMER AS C ON O.CUSTOMERID = C.ID
GROUP BY
 I.CATEGORY,
 C.CITY
 /* 7 which seller sells in multiple categories */
SELECT
 S.NAME,
 COUNT(I.CATEGORY) AS CATEGORY_CNT
FROM
 SELLER AS S
 LEFT JOIN ITEMS AS I ON S.SELLERID = I.SELLERID
```

```
GROUP BY
 S.NAME
 /* 8find the primary category for each seller.primary category is the one which
 cotribute the highest sales for the seller */
WITH
 QUANTIFIED AS (
  SELECT
   S.SELLERID AS SELLER_ID,
   S.NAME AS SELLER_NAME,
   I.CATEGORY AS CATEGORY,
   SUM(OI.QUANTITY) AS TOTAL_QUANTITY
  FROM
   SELLER AS S
   LEFT JOIN ITEMS AS I ON S.SELLERID = I.SELLERID
   LEFT JOIN ORDER_ITEMS AS OI ON I.ID = OI.ITEMID
  GROUP BY
   SELLER_ID,
   SELLER_NAME,
   CATEGORY
 )
SELECT
 SELLER_ID,
 SELLER_NAME,
 CATEGORY AS PRIMARY_CATEGORY
FROM
 (
  SELECT
```

```
  SELLER_ID,
  SELLER_NAME,
  CATEGORY,
  TOTAL_QUANTITY,
  DENSE_RANK() OVER (
   PARTITION BY
    SELLER_NAME
   ORDER BY
    TOTAL_QUANTITY DESC
  ) AS RNK
 FROM
  QUANTIFIED
) AS R
WHERE
 RNK = 1
 /* 9 which customer has ordered more frequently than others ?.
 hint: find out the average order gap for each customers and check who has the lowest
 average order gap */
WITH
 ORDER_GAPS AS (
  SELECT
   CUSTOMERID,
   ROUND(AVG((NEXT_ORDER - ORDER_DATE)::INT), 2) AS ORDER_GAP
  FROM
   (
    SELECT
     O.CUSTOMERID AS CUSTOMERID,
```

```
    DATE1 AS ORDER_DATE,
    LEAD(DATE1) OVER (
     PARTITION BY
      CUSTOMERID
     ORDER BY
      DATE1
    ) AS NEXT_ORDER
   FROM
    ORDERS AS O
   GROUP BY
    O.CUSTOMERID,
    ORDER_DATE
   ORDER BY
    O.CUSTOMERID
  ) AS N
 WHERE
  NEXT_ORDER IS NOT NULL
 GROUP BY
  CUSTOMERID
)
SELECT
 C.NAME
FROM
 CUSTOMER C
 RIGHT JOIN ORDER_GAPS AS OG ON C.ID = OG.CUSTOMERID
WHERE
 ORDER_GAP = (
```

```
 SELECT
  MIN(ORDER_GAP)
 FROM
  ORDER_GAPS
)
 /* 10 what is the percentage sales contribution of each item to their respective categories? */
WITH
 SUMMARY AS (
  SELECT
   CATEGORY,
   ID,
   NAME,
   TOTAL_QUANTITIES,
   SUM(TOTAL_QUANTITIES) OVER (
    PARTITION BY
     CATEGORY
   ) AS CAT_SUM
  FROM
   (
    SELECT
     I.CATEGORY AS CATEGORY,
     I.ID AS ID,
     I.NAME AS NAME,
     SUM(OI.QUANTITY) AS TOTAL_QUANTITIES
    FROM
     ORDER_ITEMS AS OI
     JOIN ITEMS AS I ON OI.ITEMID = I.ID
```

```
    GROUP BY
      I.CATEGORY,
      I.ID,
      I.NAME
    ) AS S
   GROUP BY
    CATEGORY,
    ID,
    NAME,
    TOTAL_QUANTITIES
 )
SELECT
 CATEGORY,
 ID AS PRODUCT_ID,
 NAME AS PRODUCT_NAME,
 ROUND((TOTAL_QUANTITIES * 100.0 / CAT_SUM), 2) AS CAT_PER_SOLD
FROM
 SUMMARY
GROUP BY
 CATEGORY,
 PRODUCT_ID,
 PRODUCT_NAME,
 TOTAL_QUANTITIES,
 CAT_SUM;

/* 11 how many jaipur/mumbai customers have purchased in mobile category */
SELECT DISTINCT
```

```
 COUNT(*) AS CUSTOMER_MUMBAI
FROM
 ORDER_ITEMS AS I
 JOIN ORDERS AS O ON I.ORDERID = O.ID
 JOIN CUSTOMER AS C ON O.CUSTOMERID = C.ID
 JOIN ITEMS AS IT ON I.ITEMID = IT.ID
WHERE
 C.CITY = 'Mumbai'
 AND IT.CATEGORY = 'Mobile';

/* 12) Top 5 products ordered? */
SELECT
 ITEMID,
 I.NAME AS PRODUCTS_ORDERED,
 SUM(QUANTITY) AS TOTAL_QUANTITY
FROM
 ORDER_ITEMS AS OI
 LEFT JOIN ITEMS AS I ON OI.ITEMID = I.ID
GROUP BY
 ITEMID,
 I.NAME
ORDER BY
 TOTAL_QUANTITY DESC
LIMIT
 5;

/*13) which product has received the highest total sales? */
```

```
SELECT
 I.NAME AS PRODUCT,
 SUM(OI.QUANTITY) AS TOTAL_SALES
FROM
 ORDER_ITEMS AS OI
 JOIN ITEMS AS I ON OI.ITEMID = I.ID
GROUP BY
 PRODUCT
ORDER BY
 TOTAL_SALES DESC
LIMIT
 1;

/* 14) Top 5 categories in terms of sales? */
SELECT
 I.CATEGORY,
 SUM(QUANTITY) AS TOTAL_SALES
FROM
 ORDER_ITEMS AS OI
 JOIN ITEMS AS I ON OI.ITEMID = I.ID
GROUP BY
 I.CATEGORY
ORDER BY
 TOTAL_SALES DESC;

/* 15) Top 5 products in each category in terms of numbers of units sold ? */
WITH
```

```
 RANKED AS (
  SELECT
   CATEGORY,
   PRODUCT_NAME,
   UNITS_SOLD,
   DENSE_RANK() OVER (
    PARTITION BY
     CATEGORY
    ORDER BY
     UNITS_SOLD DESC
   ) AS RNK
  FROM
   (
    SELECT
     I.CATEGORY AS CATEGORY,
     I.NAME AS PRODUCT_NAME,
     SUM(QUANTITY) AS UNITS_SOLD
    FROM
     ORDER_ITEMS AS OI
     JOIN ITEMS AS I ON OI.ITEMID = I.ID
    GROUP BY
     I.CATEGORY,
     I.NAME
   ) AS S
 )
SELECT
 CATEGORY,
```

```
 PRODUCT_NAME,
 UNITS_SOLD
FROM
 RANKED
WHERE
 RNK <= 5
ORDER BY
 CATEGORY,
 UNITS_SOLD DESC;

/* 16) Highest number of orders comes from which city? */
SELECT
 C.CITY,
 COUNT(O.ID) ORDERS_CNT
FROM
 ORDERS AS O
 JOIN CUSTOMER AS C ON O.CUSTOMERID = C.ID
GROUP BY
 C.CITY
ORDER BY
 ORDERS_CNT DESC
LIMIT
 1;

/* 17) How many sellers sell iphone? ( be it any model) */
SELECT DISTINCT
 S.NAME
```

```
FROM
 ITEMS AS I
 JOIN SELLER AS S ON I.SELLERID = S.SELLERID
WHERE
 TRIM(I.NAME) ILIKE '%iphone%';

/* 18) Which seller has not achieved any order till date? */
WITH
 NO_ORDER AS (
  SELECT
   I.SELLERID AS SELLERID,
   I.ID AS ITEMID
  FROM
   ORDER_ITEMS AS OI
   RIGHT JOIN ITEMS AS I ON OI.ITEMID = I.ID
  WHERE
   OI.ITEMID IS NULL
 )
SELECT
 S.NAME AS SELLER_NAME,
 N.ITEMID AS NOT_ORDERED_ITEMID
FROM
 SELLER AS S
 RIGHT JOIN NO_ORDER AS N ON S.SELLERID = N.SELLERID
 /* 19 ) which seller do not sell in which category? */
SELECT
 S.SELLERID,
```

```
 C.CATEGORY
FROM
 SELLER S
 CROSS JOIN (
  SELECT DISTINCT
   CATEGORY
  FROM
   ITEMS
 ) C
WHERE
 NOT EXISTS (
  SELECT
   1
  FROM
   ITEMS I
  WHERE
   I.SELLERID = S.SELLERID
   AND I.CATEGORY = C.CATEGORY
 )
ORDER BY
 S.SELLERID,
 C.CATEGORY
 /* 20 ) which seller has highest number of listings? */
SELECT
 S.SELLERID,
 S.NAME,
 COUNT(*) AS LISTINGS
```

```
FROM
 ITEMS AS I
 JOIN SELLER AS S ON I.SELLERID = S.SELLERID
GROUP BY
 S.SELLERID,
 S.NAME
ORDER BY
 LISTINGS DESC
LIMIT
 1;

/* 21 ) What is the average order value in each category? */
SELECT
 CATEGORY,
 ROUND(AVG(TOTAL_SALES), 2) AVG_ORDER_VALUE
FROM
 (
  SELECT
   I.CATEGORY AS CATEGORY,
   SUM(QUANTITY * UNITPRICE) TOTAL_SALES
  FROM
   ORDER_ITEMS AS OI
   JOIN ITEMS AS I ON OI.ITEMID = I.ID
  GROUP BY
   I.CATEGORY
 ) S
GROUP BY
```

```
 CATEGORY
ORDER BY
 AVG_ORDER_VALUE DESC;

/* 22) Which payment method is widely used by customers? */
SELECT
 PAYMENT_METHOD,
 COUNT(*) TOTAL_ORDERS
FROM
 ORDERS
GROUP BY
 PAYMENT_METHOD
ORDER BY
 TOTAL_ORDERS DESC
LIMIT
 1;

/* 23) List of employees who always transact in cash? */
SELECT
 C.NAME
FROM
 ORDERS AS O
 JOIN CUSTOMER AS C ON O.CUSTOMERID = C.ID
WHERE
 C.NAME NOT IN (
  SELECT DISTINCT
   C.NAME
```

```
 FROM
  ORDERS AS O
  JOIN CUSTOMER AS C ON O.CUSTOMERID = C.ID
 WHERE
  PAYMENT_METHOD NOT ILIKE 'cash'
)
/* 24) On which day highest number of orders have been ordered? */
SELECT
 DATE1 AS DATE,
 COUNT(*) NUM_ORDERS
FROM
 ORDERS
GROUP BY
 DATE
ORDER BY
 NUM_ORDERS DESC
LIMIT
 1;

/* 25) How many orders are there in which customer's city and seller's city are same? ( This is called

Regional Utilization (RU), Where demand and supply are from the same region) */
SELECT
 C.CITY,
 COUNT(O.ID) ORDERS_CNT
FROM
 ORDER_ITEMS AS OI
```

```
 JOIN ORDERS AS O ON OI.ORDERID = O.ID
 JOIN CUSTOMER AS C ON O.CUSTOMERID = C.ID
 JOIN ITEMS AS I ON OI.ITEMID = I.ID
 JOIN SELLER AS S ON I.SELLERID = S.SELLERID
WHERE
 S.CITY = C.CITY
GROUP BY
 C.CITY
 /* 26) which city supplies most number of products in Men's clothing ? Note : Supply comes from
the
 sellers. */
SELECT
 S.CITY,
 COUNT(DISTINCT OI.ITEMID)
FROM
 ORDER_ITEMS AS OI
 JOIN ITEMS AS I ON OI.ITEMID = I.ID
 JOIN SELLER AS S ON I.SELLERID = S.SELLERID
WHERE
 I.CATEGORY ILIKE 'Menclothing'
GROUP BY
 S.CITY;

/* 27) How many products would be there in an order on an average */
SELECT
 ROUND(SUM(N_ITEMS) / COUNT(*), 2) AVG_ITEMS_PER_ORDER
FROM
```

```
 (
  SELECT
   ORDERID,
   COUNT(ITEMID) N_ITEMS
  FROM
   ORDER_ITEMS
  GROUP BY
   ORDERID
 ) AS S
 /*28) which products combination has been ordered the highest ? */
WITH
 COMBINATIONS AS (
  SELECT
   OI1.ORDERID AS ORDER_ID,
   OI1.ITEMID AS FIRST_PRODUCT,
   OI2.ITEMID AS SECOND_PRODUCT
  FROM
   ORDER_ITEMS AS OI1
   JOIN ORDER_ITEMS AS OI2 ON OI1.ORDERID = OI2.ORDERID
   AND OI1.ITEMID <> OI2.ITEMID
   AND OI1.ITEMID < OI2.ITEMID
 )
SELECT
 FIRST_PRODUCT,
 I.NAME AS FIRST_NAME,
 SECOND_PRODUCT,
 I2.NAME AS SECOND_NAME,
```

```
 COUNT(*) CMBN_CNT
FROM
 COMBINATIONS AS C
 LEFT JOIN ITEMS AS I ON C.FIRST_PRODUCT = I.ID
 LEFT JOIN ITEMS AS I2 ON C.SECOND_PRODUCT = I2.ID
GROUP BY
 FIRST_PRODUCT,
 I.NAME,
 SECOND_PRODUCT,
 SECOND_NAME
ORDER BY
 CMBN_CNT DESC
LIMIT
 1;


/* 29) If the company decided to give cashback to customer, 5% cashback on first order, 10% on all
the
subsequent orders, calculate the total cashback each customer has received till date */
WITH
 CASHBACK AS (
  SELECT
   CUSTOMERID,
   ID,
   ORDERAMOUNT,
   ORDER_SERIES,
   CASE
    WHEN ORDER_SERIES = 1 THEN ORDERAMOUNT * 0.05
```

```
    ELSE ORDERAMOUNT * 0.1
   END AS CASHBACK_AMOUNT
  FROM
   (
    SELECT
     CUSTOMERID,
     ID,
     ORDERAMOUNT,
     DENSE_RANK() OVER (
      PARTITION BY
       CUSTOMERID
      ORDER BY
       ID ASC
     ) AS ORDER_SERIES
    FROM
     ORDERS
    ORDER BY
     CUSTOMERID,
     ID
   ) AS U
 )
SELECT
 CUSTOMERID,
 C.NAME,
 SUM(CASHBACK_AMOUNT) AS TOTAL_CASHBACK
FROM
 CASHBACK AS CB
```

```
 LEFT JOIN CUSTOMER AS C ON CB.CUSTOMERID = C.ID
GROUP BY
 CUSTOMERID,
 C.NAME
ORDER BY
 TOTAL_CASHBACK DESC;

/*30) Classify the customers into : a) Gold b) Silver C) Bronze Gold : total order value till date is
greater than 50000, Silver : total order value till date is between 30000 and 50000 Bronze : total
order
value till date is less than 20000 */
SELECT
 CUSTOMERID,
 TOTAL_ORDER,
 CASE
  WHEN TOTAL_ORDER > 50000 THEN 'Gold'
  WHEN TOTAL_ORDER BETWEEN 50000 AND 20000  THEN 'Silver'
  ELSE 'Bronze'
 END AS CUSTOMER_CLASS
FROM
 (
  SELECT
   CUSTOMERID,
   SUM(ORDERAMOUNT) AS TOTAL_ORDER
  FROM
   ORDERS AS O
  GROUP BY
```

CUSTOMERID

) U