Task 1:

Q1. Write a query to find the list customer_name, customer_id, different cards owned and balance left on each card for all Premium customers?

```
SELECT c.customer_name, c.id AS customer_id, cm.card_name, cc.balance_left
FROM customer c JOIN customer_cards cc ON c.id = cc.customer_id
JOIN card_master cm ON cc.card_id = cm.id
WHERE c.customer_type = 'Premium';
```

Q2. List the total renews of customers by different card types. Output should contain customer_id, card_type, card_name and total renewd_amount?

```
SELECT cr.customer_id, cm.card_type, cm.card_name, SUM(cr.renew_amount)
        AS total_renewed_amount
FROM card_renewals cr
JOIN customer_cards cc ON cr.customer_id = cc.customer_id AND cr.card_id = cc.card_id
JOIN  card_master cm ON cc.card_id = cm.id
GROUP BY cr.customer_id, cm.card_type, cm.card_name;
```

Q3. List total calls made between 4th April 2018 and 5th May 2018 for all premium customers whose call duration is greater than 10 minutes by day.Output should contain total calls , date?

```
SELECT  DATE(cr.call_datetime) AS call_date, COUNT(*) AS total_calls
FROM call_records cr
JOIN  customer c ON cr.customer_id = c.id
JOIN  customer_cards cc ON cr.customer_id = cc.customer_id AND cr.card_id = cc.card_id
JOIN  card_master cm ON cc.card_id = cm.id
WHERE  cm.card_type = 'Premium'
   AND cr.call_datetime BETWEEN '2018-04-04' AND '2018-05-05'
   AND TIMESTAMPDIFF(MINUTE, cr.call_datetime, cr.call_endtime) > 10
GROUP BY call_date;
```

Q4. List total calls, card used, total call duration, customer name, current card balance for all customers for the month of April 2018 by customer, card type ?

```
SELECT c.customer_name, cm.card_type,  SUM(1) AS total_calls, cc.card_id  AS card_used,

        SEC_TO_TIME(SUM(TIMESTAMPDIFF(SECOND, cr.call_datetime, cr.call_endtime))) AS
        total_call_duration, cc.balance_left  AS  current_card_balance

FROM customer c

JOIN customer_cards cc ON c.id = cc.customer_id

JOIN card_master cm ON cc.card_id = cm.id

LEFT JOIN call_records cr ON c.id = cr.customer_id

WHERE cr.call_datetime >= '2018-04-01' AND cr.call_datetime < '2018-05-01'

GROUP BY c.customer_name, cm.card_type, cc.card_id, cc.balance_left;
```

Task 2:

Q1. What are different types of joins?

Inner Join: - Retrieves records that have matching values in both tables.

Left Join: - Retrieves all records from the left table (Table1), and the matched records from the right table (Table2).

Right Join: - Retrieves all records from the right table (Table2), and the matched records from the left table (Table1).

Full Outer Join: - Retrieves all records when there is a match in either left or right table.

Cross Join: - Pairs each row from one table with every row from another table.

Self Join: - Join a table to itself, which is useful for comparing rows within the same table

Q2. Display all faq_id, questions for a given partner_id. What kind of join is it?

```
SELECT fq.id AS faq_id, fq.question
FROM Faq_questions fq
JOIN partner_questions pq ON fq.Id = pq.Faq_mapping_id
WHERE pq.Partner_id = <partner_id>;
```

(It's an **INNER JOIN** between two tables, Faq_questions and partner_questions)

Q3. Display questions,partner_id for a given category_id and all partners?

```
SELECT fq.question, pq.Partner_id
FROM Faq_questions fq
JOIN Category_faq_mapping cfm ON fq.Id = cfm.Faq_id
JOIN partner_questions pq ON cfm.Id = pq.Faq_mapping_id
WHERE cfm.Category_id = <category_id>;
```

Q4. Display question_id, question, answer_id, answer for all partners?

```
SELECT fq.id AS question_id, fq.question, ca.id AS answer_id,ca.answer
FROM Faq_questions fq
JOIN Category_faq_mapping cfm ON fq.Id = cfm.Faq_id
JOIN Category_answers ca ON cfm.Id = ca.faq_mapping_id
JOIN partner_questions pq ON cfm.Id = pq.Faq_mapping_id;
```

Task 3:

Q1. Get total orders and total order value per day for the month of March 2018 ?

```
SELECT Order_date,COUNT(Order_id) AS Total_orders, SUM(Order_value)
        AS   Total_order_value
FROM Orders
WHERE MONTH (Order_date) = 3 AND YEAR(Order_date) = 2018
GROUP BY Order_date
ORDER BY Order_date;
```

Q2. Get Total orders,order value per day for the month of March 2018 for each order type ?

```
SELECT Order_date,Order_type, COUNT(Order_id) AS Total_orders, SUM(Order_value)
        AS Total_order_value
FROM Orders
WHERE Order_date >= '2018-03-01' AND Order_date < '2018-04-01'
GROUP BY Order_date, Order_type
ORDER BY Order_date, Order_type;
```

Q3. Get average order_value for the month of March 2018 for all successful orders for each category?

```
SELECT Order_type, AVG (Order_value) AS Avg_order_value
FROM Orders
WHERE Order_date >= '2018-03-01' AND Order_date < '2018-04-01'
        AND Order_status = 'Success'
GROUP BY Order_type
ORDER BY Order_type;
```

Task 4:

> Table 1 (A flat table which contains the following information) Customer_name, cutsomer_email,category_name,order_value,order_date,order_id

```
CREATE TABLE AggregatedOrders AS
SELECT c.customer_name AS Customer_name,c.email AS Customer_email,
        cat.category_name AS Category_name, o.order_value AS Order_value,
        o.order_date AS Order_date, o.order_id AS Order_id
FROM Orders o
JOIN Customer c ON o.customer_id = c.customer_id
JOIN Categories cat ON o.category_id = cat.category_id;
```

> Table 2 (Total discount value availed by customer - Aggregated table) Customer_name,customer_email,total_discount

```
CREATE TABLE TotalDiscounts AS
SELECT c.customer_name AS Customer_name, c.email AS Customer_email,
        SUM (o.discount) AS Total_discount
FROM Orders o
JOIN Customer c ON o.customer_id = c.customer_id
GROUP BY c.customer_name, c.email;
```

> Table 3 (Month and category wise order_value - Aggregated table) Month,category_name,total_order_value

```
CREATE TABLE MonthCategoryOrders AS
SELECT DATE_FORMAT (order_date, '%Y-%m') AS Month,
        cat.category_name AS Category_name, SUM(o.order_value) AS Total_order_value
FROM Orders o
JOIN Categories cat ON o.category_id = cat.category_id
GROUP BY Month, Category_name;
```