Question 1

Marks: 1/1

Books Priced at $10 and Above

Write an SQL query to retrieve the book titles and customer names for the books priced at $10 and above that have been ordered. Use the `goodreads` and `orders` tables

Your answer

```
-- Write your query below
SELECT g.book_title, o.customer_id
FROM goodreads g
JOIN orders o ON g.book_id = o.book_id
WHERE g.price >= 10;
```

****

Question 2

Marks: 1/1

Total Quantity of Spare Parts for a Vehicle Model

We have a table named `SpareParts` that contains information about various spare parts, including the quantity available and the vehicle model they belong to.

Now, write a SQL query to get the total quantity of all spare parts for a particular vehicle model.

Sample Output:

TotalQuantity
35

Your answer

```
-- Write your query below
SELECT SUM(Quantity) AS TotalQuantity
FROM spareparts
WHERE Model = 'four';
```

****

Question 3

Establishing Foreign Key Relationship in MySQL

We do have two tables named `Department` and `Employee` tables. the task at hand is to create a foreign key relationship with respect to `Department` table with `Employee table`.

Note: Due to SQLite compatibility, we can't make use of `ALTER TABLE ADD FOREIGN KEY` constraint. Instead, create a new table with name "Employees" by enabling the Foreign Key constraint in the table creation and then Insert the respective values of "Employee" table into the "Employees" table and drop the table the "Employee" table.

Input:

Department Table:

```
+---------------+-----------------+
| department_id | department_name |
+---------------+-----------------+
| 1          | HR             |
| 2          | IT             |
| 3          | Marketing       |
+---------------+-----------------+
```

Employee Table:

```
+---------------+------------+------------+---------------+
| employee_id   | first_name | last_name  | department_id |
+---------------+------------+------------+---------------+
| 1          | John      | Doe     | 2          |
| 2          | Jane      | Smith   | 1          |
| 3          | Bob       | Johnson       | 3          |
| 4          | Alice           | Williams      | 1          |
+---------------+------------+------------+---------------+
```

Output:

Employees Table:

```
+---------------+------------+------------+---------------+
| employee_id   | first_name | last_name  | department_id |
+---------------+------------+------------+---------------+
| 1          | John      | Doe     | 2          |
| 2          | Jane      | Smith   | 1          |
| 3          | Bob       | Johnson       | 3          |
| 4          | Alice           | Williams      | 1          |
+---------------+------------+------------+---------------+
```

Your answer

```
CREATE TABLE new_employees (
    employee_id int primary key,
    first_name varchar(50) not null,
```

```
    last_name varchar(50) not null,
    department_id INT,
    FOREIGN KEY(department_id) REFERENCES departments(department_id)
);
INSERT INTO new_employees(employee_id, first_name, last_name, department_id)
values(1, 'John', 'Doe', 2),
    (2, 'Jane', 'Smith', 1),
    (3, 'Bob', 'Johnson', 3),
    (4, 'Alice', 'Williams', 1);

DROP TABLE employees;
ALTER TABLE new_employees RENAME TO employees;
```

****

Question 4

Marks: 1/1

Top Countries by Films Rented

Write a query to find the top 3 countries by the total number of films rented by customers living in those countries.

Sample Output:

Country_id	Country	rental_count
44	India	1572

Your answer

```
-- Write your query below
SELECT
    co.country_id,
    co.country,
    COUNT(r.rental_id) AS rental_count
FROM rental r
JOIN customer c ON r.customer_id = c.customer_id
JOIN address a ON c.address_id = a.address_id
JOIN city ci ON a.city_id = ci.city_id
JOIN country co ON ci.country_id = co.country_id
GROUP BY co.country_id, co.country
```

```
ORDER BY rental_count DESC
LIMIT 3;
**********************************************************************
```

Question 5

Show dishes with prices for a specific restaurant

We have two tables: `Restaurant` and `Dishes`. The `Restaurant` table contains information about different restaurants, including their names. The `Dishes` table contains information about various dishes, including their names, prices, and the restaurant they belong to.

Now write a query to retrieve all the dishes with their respective prices that belong to the restaurant 'Punjabi Rasoi'.

Sample Output:

DishName	Price
Dal Makhni	120

Your answer

```
-- Write your query below
SELECT
    d.DishName,
    d.Price
FROM Dishes d
JOIN Restaurant r ON d.RestaurantId = r.Id
WHERE r.Name = 'Punjabi Rasoi';
**********************************************************************
```

Question 6

Marks: 1/1

Top Revenue Categories and Average Revenues

Write a query to determine the top 5 categories by total revenue and compare their average revenues to the overall average revenue of films in the database.

Sample Output:

Category_id	name	total_revenue	percentage_of_overall_avg_revenue

15	Sports	5314.21	126.141076
14	Sci-Fi	4756.98	112.914351

Your answer-- Write your query below

```
SELECT
    c.category_id,
    c.name,
    SUM(p.amount) AS total_revenue,
    (SUM(p.amount) / (
        SELECT AVG(total_cat_revenue)
        FROM (
            SELECT SUM(p2.amount) AS total_cat_revenue
            FROM category c2
            JOIN film_category fc2
                ON c2.category_id = fc2.category_id
            JOIN film f2
                ON fc2.film_id = f2.film_id
            JOIN inventory i2
                ON f2.film_id = i2.film_id
            JOIN rental r2
                ON i2.inventory_id = r2.inventory_id
            JOIN payment p2
                ON r2.rental_id = p2.rental_id
            GROUP BY c2.category_id
        ) sub
    )) * 100 AS percentage_of_overall_avg_revenue
FROM category c
JOIN film_category fc
    ON c.category_id = fc.category_id
JOIN film f
    ON fc.film_id = f.film_id
JOIN inventory i
    ON f.film_id = i.film_id
JOIN rental r
    ON i.inventory_id = r.inventory_id
JOIN payment p
    ON r.rental_id = p.rental_id
GROUP BY c.category_id, c.name
ORDER BY total_revenue DESC
LIMIT 5;
```

****

Question 7

Marks: 1/1

Employee Salaries Details

You are given two tables: Employees and Salaries. The Employees table contains the details of employees in a company, and the Salaries table contains their salary information.

Employees Table:

Salaries Table:

Write a query to find the average salary of employees in each department, but only include departments where the average salary is greater than 50000.

Table: Employees

Showing top 5 rows

EmpID	Name	DeptID	Position
1	Alice	1	Manager
2	Bob	2	Developer
3	Charlie	1	Analyst
4	David	2	Developer
5	Eva	3	HR

Table: Salaries

Showing top 5 rows

EmpID	Salary
1	60000

2 55000

3 45000

4 70000

5 48000

Your answer

```
-- Write your query below
SELECT e.DeptID, ROUND(AVG(s.Salary), 2) AS AvgSalary
FROM Employees e
JOIN Salaries s ON e.EmpID = s.EmpID
GROUP BY e.DeptID
ORDER BY AvgSalary DESC
LIMIT 2;
**********************************************************************************
```

Question 8

Top Films by Number of Rentals

Write a query to determine the top 3 films in terms of the number of rentals in each store.

Sample Output:

store_id	flim_id	title	rental_count
1	535	LOVE SUICIDES	20
2	55	BARBARELLA STREETCAR	18

Your answer

```
-- Write your query below
SELECT
  store_id,
  film_id,
  title,
  rental_count
FROM (
  SELECT
    i.store_id,
    f.film_id,
    f.title,
    COUNT(r.rental_id) AS rental_count,
    ROW_NUMBER() OVER (PARTITION BY i.store_id ORDER BY COUNT(r.rental_id) DESC)
AS rank
  FROM rental r
```

```
  JOIN inventory i ON r.inventory_id = i.inventory_id
  JOIN film f ON i.film_id = f.film_id
  GROUP BY i.store_id, f.film_id, f.title
) ranked
WHERE rank <= 3
ORDER BY store_id, rank;
```

****

Question 9

Marks: 1/1

Customers with Bank and Crypto Transactions

We have four tables: bank_transactions, bank_accounts, trading_accounts, and crypto_transactions. The bank_transactions table contains information about bank transactions, the bank_accounts table contains information about bank accounts, the trading_accounts table contains information about trading accounts, and the crypto_transactions table contains information about crypto transactions.

Your task is to write a query to determine how many customers have conducted both bank transactions and crypto transactions on the same day.

Sample Output:

TransactionID	BankSenderAccountID	BankReceiverAccountID	PaymentID	BankTransactionType	Amount	TransactionDate	BankAccountID	BankID	CustomerID	TotalBalance	Status	CreationDate	TradingAccountID	Balance	CryptoSenderAccountID	CryptoReceiverAccountID	AssetID	CryptoTransactionType	Quantity
10363	306052	57898	116	Payment	609.65	2023-02-15 21:46:5(	306052	32	26621	41684.94	Completed	2023-07-08 04:5(	68832	92897	68832	96708	88	Transfer	34.51

Your answer

```
-- Write your query below
SELECT DISTINCT *
FROM Bank_Transactions bt
JOIN Bank_Accounts ba ON bt.BankSenderAccountID = ba.BankAccountID
JOIN Trading_Accounts ta ON ta.BankAccountID = ba.BankAccountID
JOIN Crypto_Transactions ct ON ct.CryptoSenderAccountID = ta.TradingAccountID
WHERE DATE(bt.TransactionDate) = DATE(ct.TransactionDate);
```

****

Question 10

Marks: 1/1

Hypothetical Bank Data

Find the top 3 customers with the highest total transaction amount for each city. Details are given below.

Tables

1. Customers

2. CustomerID

Name

City

3. Accounts

AccountID

CustomerID

AccountType

Balance

4. Transactions

TransactionID

AccountID

TransactionDate

Amount

Your answer

```
WITH CustomerTransactionTotals AS (
    SELECT
        c.CustomerID,
        c.Name,
        c.City,
        SUM(t.Amount) AS TotalTransactionAmount
    FROM Customers c
    JOIN Accounts a ON c.CustomerID = a.CustomerID
    JOIN Transactions t ON a.AccountID = t.AccountID
    GROUP BY c.CustomerID, c.Name, c.City
),
RankedCustomers AS (
    SELECT
        CustomerID,
```

```
        Name,
        City,
        TotalTransactionAmount,
        ROW_NUMBER() OVER (PARTITION BY City ORDER BY TotalTransactionAmount
DESC) AS TransactionRank
    FROM CustomerTransactionTotals
)

SELECT
    CustomerID,
    Name,
    City,
    TotalTransactionAmount,
    TransactionRank
FROM RankedCustomers
WHERE TransactionRank <= 3
ORDER BY City, TransactionRank;

**********************************************************************************
```

Question 11

Oders_Customers

You have two tables, orders and customers, with the following structures:

orders table:

order_id (integer): The unique identifier for each order.

customer_id (integer): The unique identifier for each customer.

order_date (date): The date when the order was placed.

amount (decimal): The total amount of the order.

customers table:

customer_id (integer): The unique identifier for each customer.

customer_name (varchar): The name of the customer.

city (varchar): The city where the customer resides.

Write an SQL query to find the top 3 customers by total order amount in 2023 and list their names, total order amounts, and the number of orders they placed. Ensure the results are ordered by the total order amount in descending order.

Your answer

```
-- Write your query below
SELECT
    c.customer_name,
    SUM(o.amount) AS total_amount,
    COUNT(o.order_id) AS order_count
FROM
    orders o
JOIN
    customers c ON o.customer_id = c.customer_id
WHERE
    strftime('%Y', o.order_date) = '2023'
GROUP BY
    c.customer_id, c.customer_name
ORDER BY
    total_amount DESC
LIMIT 3;
```