Muslim Bayemirov:

1. Find the average rating of cryptocurrencies that are in the watchlists of premium users with a Gold subscription and have a 24-hour price change greater than 5.

```
MariaDB [Group-26]> SELECT AVG(c.Rating) AS Average_Rating
    -> FROM Cryptocurrency c
    -> JOIN Watchlist_Crypto wc ON c.CryptoID = wc.CryptoID
    -> JOIN Premium_User_Watchlist puw ON wc.WatchlistID = puw.WatchlistID
    -> JOIN Premium_User pu ON puw.UserID = pu.UserID
    -> JOIN Gold_Subscription_User gu ON pu.UserID = gu.UserID
    -> JOIN PremiumCrypto pc ON c.CryptoID = pc.CryptoID
    -> WHERE pc.Change_In_24_Hours > 5;
+----------------+
| Average_Rating |
+----------------+
|           NULL |
+----------------+
1 row in set (0.001 sec)
```

2. Find the total sales volume of cryptocurrencies owned by Basic Users, grouped by their subscription start year.

```
MariaDB [Group-26]> SELECT SUBSTRING(pu.Subscription_Start_Date, 1, 4) AS Subscr
iption_Start_Year,
    ->        SUM(c.Sales_Volume) AS Total_Sales_Volume
    -> FROM Cryptocurrency c
    -> JOIN Basic_User_Crypto buc ON c.CryptoID = buc.CryptoID
    -> JOIN Basic_User bu ON buc.UserID = bu.UserID
    -> JOIN Premium_User pu ON bu.UserID = pu.UserID
    -> GROUP BY Subscription_Start_Year;
+------------------------+--------------------+
| Subscription_Start_Year | Total_Sales_Volume |
+------------------------+--------------------+
| 2023                   |          100000.00 |
+------------------------+--------------------+
1 row in set (0.001 sec)
```

3. Find the number of premium users with a Platinum subscription who have a watchlist containing more than 3 cryptocurrencies.

```
MariaDB [Group-26]> SELECT COUNT(pu.UserID) AS Premium_Platinum_Users_With_Watch
list
    -> FROM Premium_User pu
    -> JOIN Platinum_Subscription_User psu ON pu.UserID = psu.UserID
    -> JOIN Premium_User_Watchlist puw ON pu.UserID = puw.UserID
    -> JOIN (SELECT WatchlistID, COUNT(*) AS CryptoCount FROM Watchlist_Crypto G
ROUP BY WatchlistID) wc ON puw.WatchlistID = wc.WatchlistID
    -> WHERE wc.CryptoCount > 3;
+--------------------------------------+
| Premium_Platinum_Users_With_Watchlist |
+--------------------------------------+
|                                    0 |
+--------------------------------------+
1 row in set (0.001 sec)
```

Serzhan Kenesbek:

1. Find Premium_User(s) who have an average price of cryptocurrencies in their watchlist greater than $1,000, and list the watchlists along with the average price (using GROUP BY and HAVING clauses).

```
MariaDB [Group-26]> SELECT w.WatchlistID, AVG(c.Price) AS Average_Price
    -> FROM Watchlist w
    -> JOIN Watchlist_Crypto wc ON w.WatchlistID = wc.WatchlistID
    -> JOIN Cryptocurrency c ON wc.CryptoID = c.CryptoID
    -> JOIN Premium_User pu ON w.UserID = pu.UserID
    -> GROUP BY w.WatchlistID
    -> HAVING AVG(c.Price) > 1000.0;
+-------------+---------------+
| WatchlistID | Average_Price |
+-------------+---------------+
|           1 |  26750.000000 |
+-------------+---------------+
1 row in set (0.001 sec)

MariaDB [Group-26]>
```

2. Find all Premium_User(s) and their associated Subscription_Type (using JOIN).

```
MariaDB [Group-26]> SELECT u.Username, pu.Subscription_Type
    -> FROM User u
    -> JOIN Premium_User pu ON u.UserID = pu.UserID;
+----------+-------------------+
| Username | Subscription_Type |
+----------+-------------------+
| user1    | Silver            |
| user3    | Gold              |
+----------+-------------------+
2 rows in set (0.001 sec)
```

3. Calculate the total sales volume of all cryptocurrencies in the Cryptocurrency table (using Aggregate function SUM()).

```
MariaDB [Group-26]> SELECT SUM(Sales_Volume) AS Total_Sales_Volume
    -> FROM Cryptocurrency;
+--------------------+
| Total_Sales_Volume |
+--------------------+
|          152000.00 |
+--------------------+
1 row in set (0.001 sec)
```

Batyikhan Ibragimov:

1. Retrieves the highest price among the cryptocurrencies in the Watchlist of each Premium User, grouping the results by the username of premium users.

```
MariaDB [Group-26]> SELECT u.Username, MAX(c.Price) AS Highest_Price
    -> FROM User u
    -> JOIN Premium_User pu ON u.UserID = pu.UserID
    -> JOIN Premium_User_Watchlist pw ON pu.UserID = pw.UserID
    -> JOIN Watchlist_Crypto wc ON pw.WatchlistID = wc.WatchlistID
    -> JOIN Cryptocurrency c ON wc.CryptoID = c.CryptoID
    -> GROUP BY u.Username;
+----------+---------------+
| Username | Highest_Price |
+----------+---------------+
| user1    |      50000.00 |
| user3    |          2.00 |
+----------+---------------+
2 rows in set (0.001 sec)
```

2. Retrieve the average price of cryptocurrencies in each watchlist.

```
MariaDB [Group-26]> SELECT w.WatchlistID, AVG(c.Price) AS AveragePrice
    -> FROM Watchlist w
    -> JOIN Premium_User_Watchlist pw ON w.WatchlistID = pw.WatchlistID
    -> JOIN Watchlist_Crypto wc ON w.WatchlistID = wc.WatchlistID
    -> JOIN Cryptocurrency c ON wc.CryptoID = c.CryptoID
    -> GROUP BY w.WatchlistID;
+-------------+--------------+
| WatchlistID | AveragePrice |
+-------------+--------------+
|           1 | 26750.000000 |
|           2 |     2.000000 |
+-------------+--------------+
2 rows in set (0.001 sec)
```

3. Calculate the total price of cryptocurrencies for each basic user, then grouping the results by the username.

```
MariaDB [Group-26]> SELECT u.Username, SUM(c.Price) AS Total_Price
    -> FROM User u
    -> JOIN Basic_User bu ON u.UserID = bu.UserID
    -> JOIN Basic_User_Crypto buc ON bu.UserID = buc.UserID
    -> JOIN Cryptocurrency c ON buc.CryptoID = c.CryptoID
    -> GROUP BY u.Username;
+----------+-------------+
| Username | Total_Price |
+----------+-------------+
| user1    |    50000.00 |
| user2    |     3500.00 |
+----------+-------------+
2 rows in set (0.001 sec)
```