-- Your name and cohort here

Yurii Maisuradze, C402

/\*

Aggregate Queries

REQUIREMENT - Use a multi-line comment to paste the first 5 or fewer results under your query

             THEN records returned.

\*/

USE orderbook\_activity\_db;

-- #1: How many users do we have?

**SELECT COUNT(\*) AS totalUsers FROM User;**

7

Obraz zawierający tekst, zrzut ekranu, Czcionka, czarne

Opis wygenerowany automatycznie

-- #2: List the username, userid, and number of orders each user has placed.

**SELECT User.uname, User.userid, COUNT(Order.orderid) AS numberOfOrders FROM User JOIN `Order` ON User.userid = Order.userid GROUP BY User.userid;**

Obraz zawierający tekst, zrzut ekranu, czarne

Opis wygenerowany automatycznie

-- #3: List the username, symbol, and number of orders placed for each user and for each symbol.

-- Sort results in alphabetical order by symbol.

**SELECT User.uname, Order.symbol, COUNT(Order.orderid) AS numberOfOrders FROM User JOIN `Order` ON User.userid = Order.userid GROUP BY User.uname, Order.symbol ORDER BY Order.symbol ASC;**

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

-- #4: Perform the same query as the one above, but only include admin users.

**SELECT User.uname, Order.symbol, COUNT(Order.orderid) AS numberOfOrders FROM User JOIN `Order` ON User.userid = Order.userid JOIN UserRoles ur ON User.userid = ur.userid JOIN Role r ON ur.roleid = r.roleid WHERE r.name = 'admin' GROUP BY User.uname, Order.symbol ORDER BY Order.symbol ASC;**

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

-- #5: List the username and the average absolute net order amount for each user with an order.

-- Round the result to the nearest hundredth and use an alias (averageTradePrice).

-- Sort the results by averageTradePrice with the largest value at the top.

**SELECT User.uname, ROUND(AVG(ABS(Order.shares)), 2) AS averageTradePrice FROM User JOIN `Order` ON User.userid = Order.userid GROUP BY User.uname ORDER BY averageTradePrice DESC;**

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

-- #6: How many shares for each symbol does each user have?

-- Display the username and symbol with number of shares.

**SELECT User.uname, Order.symbol, SUM(Order.shares) AS totalShares FROM User JOIN `Order` ON User.userid = Order.userid GROUP BY User.uname, Order.symbol;**

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

-- #7: What symbols have at least 3 orders?

**SELECT Order.symbol, COUNT(Order.orderid) AS numberOfOrders FROM `Order`GROUP BY Order.symbol HAVING COUNT(Order.orderid) >= 3;**

Obraz zawierający tekst, zrzut ekranu, Czcionka, diagram

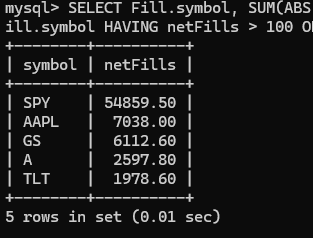
Opis wygenerowany automatycznie

-- #8: List all the symbols and absolute net fills that have fills exceeding $100.

-- Do not include the WLY symbol in the results.

-- Sort the results by highest net with the largest value at the top.

**SELECT Fill.symbol, SUM(ABS(Fill.share \* Fill.price)) AS netFills FROM Fill WHERE Fill.symbol != 'WLY' GROUP BY Fill.symbol HAVING netFills > 100 ORDER BY netFills DESC;**

****

-- #9: List the top five users with the greatest amount of outstanding orders.

-- Display the absolute amount filled, absolute amount ordered, and net outstanding.

-- Sort the results by the net outstanding amount with the largest value at the top.

**SELECT User.uname, SUM(ABS(Fill.share)) AS absoluteAmountFilled,SUM(ABS(Order.shares)) AS absoluteAmountOrdered,(SUM(ABS(Order.shares)) - SUM(ABS(Fill.share))) AS netOutstanding FROM User JOIN `Order` ON User.userid = Order.userid JOIN Fill ON Order.orderid = Fill.orderid GROUP BY User.uname ORDER BY netOutstanding DESC LIMIT 5;**

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie