test

February 25, 2023

[ ]:
```
%load_ext sql
%sql sqlite:///Data.db
```

```
The sql extension is already loaded. To reload it, use:
  %reload_ext sql
```

[ ]:
```
'Connected: @Data.db'
```

0.1 Question 1

Which brand saw the most dollars spent in the month of June?

[ ]:
```
%%sql
SELECT  b.NAME, ROUND(sum(i.TOTAL_FINAL_PRICE),2) as "Total Sales in June"
FROM Brands b,  Items i
Where i.MODIFY_DATE LIKE '%-06-%'
AND (b.BRAND_CODE = i.BRAND_CODE
OR b.BARCODE = i.BARCODE)
GROUP BY b.NAME
ORDER BY sum(i.TOTAL_FINAL_PRICE) DESC
LIMIT 1;
```

```
 * sqlite:///Data.db
Done.
```

[ ]:
```
[('Keebler', 179943.7)]
```

0.2 Question 2

Which user spent the most money in the month of August?

[ ]:
```
%%sql
SELECT u.ID as "USER ID", ROUND(sum(i.TOTAL_FINAL_PRICE),2) as "Total Spends in␣
 ↪August"
FROM Items i, Receipts r, User u
WHERE i.MODIFY_DATE LIKE '%-08-%'
AND i.REWARDS_RECEIPT_ID = r.ID
AND r.USER_ID = u.ID
```

```
GROUP BY u.ID
ORDER BY sum(i.TOTAL_FINAL_PRICE) DESC
LIMIT 1;
```

```
 * sqlite:///Data.db
Done.
```

[ ]:
```
[('609ab37f7a2e8f2f95ae968f', 157719.27)]
```

0.3 Question 3

What user bought the most expensive item?

There my be the case there are two user purchased the same expensive item. The fllowing code output one user

[ ]:
```
%%sql
SELECT u.ID as "User that purchased the most expensive item", u.STATE as␣
 ↪"State", (e.TOTAL_FINAL_PRICE/e.QUANTITY_PURCHASED) as "Price of the most␣
 ↪expensive item"
FROM Receipts r, User u, (SELECT *
FROM Items
ORDER BY (TOTAL_FINAL_PRICE/QUANTITY_PURCHASED) DESC
LIMIT 1) as e
Where e.REWARDS_RECEIPT_ID = r.ID
AND r.USER_ID = u.ID;
```

```
 * sqlite:///Data.db
Done.
```

[ ]:
```
[('617376b8a9619d488190e0b6', 'NY', 31005.99)]
```

To output all the users that purchased the most expensive item

[ ]:
```
%%sql
SELECT u.ID as "User that purchased the most expensive item", u.STATE as␣
 ↪"State", (e.TOTAL_FINAL_PRICE/e.QUANTITY_PURCHASED)  as "Price of the most␣
 ↪expensive item"
FROM Receipts r, User u, (SELECT *
FROM Items
ORDER BY (TOTAL_FINAL_PRICE/QUANTITY_PURCHASED) DESC
LIMIT 1) as e, Items i
Where (i.TOTAL_FINAL_PRICE/i.QUANTITY_PURCHASED) = (e.TOTAL_FINAL_PRICE/e.
 ↪QUANTITY_PURCHASED)
AND i.REWARDS_RECEIPT_ID = r.ID
AND r.USER_ID = u.ID;
```

```
 * sqlite:///Data.db
Done.
```

[ ]:
```
[('617376b8a9619d488190e0b6', 'NY', 31005.99),
 ('617376b8a9619d488190e0b6', 'NY', 31005.99)]
```

0.4 Question 4

What is the name of the most expensive item purchased?

[ ]:
```
%%sql
SELECT ORIGINAL_RECEIPT_ITEM_TEXT as "Most expensive item", (TOTAL_FINAL_PRICE/
 ↪QUANTITY_PURCHASED) as "Price of the most expensive item", DESCRIPTION as␣
 ↪"Description"
FROM Items
ORDER BY (TOTAL_FINAL_PRICE/QUANTITY_PURCHASED) DESC
LIMIT 1;
```

```
 * sqlite:///Data.db
Done.
```

[ ]:
```
[('STRBCKS IC CF BL', 31005.99, 'Starbucks Iced Coffee Premium Coffee Beverage
Unsweetened Blonde Roast Bottle 48 Oz 1 Ct')]
```

I used the ORIGINAL_RECEIPT_ITEM_TEXT to find the name of the most expensive item. The following code output all the elements of the most expensive item.

[ ]:
```
%%sql
SELECT *
FROM Items
ORDER BY (TOTAL_FINAL_PRICE/QUANTITY_PURCHASED) DESC
LIMIT 1;
```

```
 * sqlite:///Data.db
Done.
```

[ ]:
```
[('62c6300d0a72315a3e1b202e', '0', 'deab389b6f6ecfef51d595ab07c40e51',
'Starbucks Iced Coffee Premium Coffee Beverage Unsweetened Blonde Roast Bottle
48 Oz 1 Ct', '048500201831', 'STARBUCKS', 1.0, 31005.99, '310059.90', 'STARBUCKS
21 OZ OR LARGER MULTISERVE', 'STRBCKS IC CF BL', '2022-07-07T07:11:50.648Z')]
```

0.5 Question 5

How many users scanned in each month?

0.5.1 Scenario 1

We can calculate one user multiple times in one month. For example, a single user scanned 3 times in June, then the number of users scanned in June is 3.

[ ]:
```
%%sql
SELECT COUNT(DATE_SCANNED) as "Number of receipts scanned in June"
FROM Receipts
WHERE DATE_SCANNED LIKE '%-06-%';
```

```
 * sqlite:///Data.db
Done.
```

[ ]:
```
[(5405,)]
```

[ ]:
```
%%sql
SELECT m1."Jan", m2."Feb", m3."Mar", m4."Apr", m5."May", m6."Jun", m7."Jul", m8.
  ↪"Aug", m9."Sep", m10."Oct", m11."Nov", m12."Dec"
FROM(SELECT COUNT(USER_ID) as "Jan"
     FROM Receipts
     WHERE DATE_SCANNED LIKE '%-01-%') as m1, (SELECT COUNT(USER_ID) as "Feb"␣
  ↪FROM Receipts WHERE DATE_SCANNED LIKE '%-02-%') as m2, (SELECT␣
  ↪COUNT(USER_ID) as "Mar" FROM Receipts WHERE DATE_SCANNED LIKE '%-03-%') as␣
  ↪m3, (SELECT COUNT(USER_ID) as "Apr" FROM Receipts WHERE DATE_SCANNED LIKE␣
  ↪'%-04-%') as m4, (SELECT COUNT(USER_ID) as "May" FROM Receipts WHERE␣
  ↪DATE_SCANNED LIKE '%-05-%') as m5, (SELECT COUNT(USER_ID) as "Jun" FROM␣
  ↪Receipts WHERE DATE_SCANNED LIKE '%-06-%') as m6, (SELECT COUNT(USER_ID) as␣
  ↪"Jul" FROM Receipts WHERE DATE_SCANNED LIKE '%-07-%') as m7, (SELECT␣
  ↪COUNT(USER_ID) as "Aug" FROM Receipts WHERE DATE_SCANNED LIKE '%-08-%') as␣
  ↪m8, (SELECT COUNT(USER_ID) as "Sep" FROM Receipts WHERE DATE_SCANNED LIKE␣
  ↪'%-09-%') as m9, (SELECT COUNT(USER_ID) as "Oct" FROM Receipts WHERE␣
  ↪DATE_SCANNED LIKE '%-10-%') as m10, (SELECT COUNT(USER_ID) as "Nov" FROM␣
  ↪Receipts WHERE DATE_SCANNED LIKE '%-11-%') as m11, (SELECT COUNT(USER_ID) as␣
  ↪"Dec" FROM Receipts WHERE DATE_SCANNED LIKE '%-12-%') as m12
```

```
 * sqlite:///Data.db
Done.
```

[ ]:
```
[(4222, 3830, 4767, 4882, 5627, 5405, 6058, 6191, 6355, 7305, 7512, 8447)]
```

0.5.2 Scenario 2

We need to calculate the number of users in each month by using the unique values of the USER_ID. For example, a user scanned 3 times in June, then the number of users in June is 1.

[ ]:
```
%%sql
SELECT m1."Jan", m2."Feb", m3."Mar", m4."Apr", m5."May", m6."Jun", m7."Jul", m8.
  ↪"Aug", m9."Sep", m10."Oct", m11."Nov", m12."Dec"
FROM(SELECT COUNT(DISTINCT USER_ID) as "Jan"
     FROM Receipts
```

```
    WHERE DATE_SCANNED LIKE '%-01-%') as m1, (SELECT COUNT(DISTINCT USER_ID)␣
↪as "Feb" FROM Receipts WHERE DATE_SCANNED LIKE '%-02-%') as m2, (SELECT␣
↪COUNT(DISTINCT USER_ID) as "Mar" FROM Receipts WHERE DATE_SCANNED LIKE␣
↪'%-03-%') as m3, (SELECT COUNT(DISTINCT USER_ID) as "Apr" FROM Receipts␣
↪WHERE DATE_SCANNED LIKE '%-04-%') as m4, (SELECT COUNT(DISTINCT USER_ID) as␣
↪"May" FROM Receipts WHERE DATE_SCANNED LIKE '%-05-%') as m5, (SELECT␣
↪COUNT(DISTINCT USER_ID) as "Jun" FROM Receipts WHERE DATE_SCANNED LIKE␣
↪'%-06-%') as m6, (SELECT COUNT(DISTINCT USER_ID) as "Jul" FROM Receipts␣
↪WHERE DATE_SCANNED LIKE '%-07-%') as m7, (SELECT COUNT(DISTINCT USER_ID) as␣
↪"Aug" FROM Receipts WHERE DATE_SCANNED LIKE '%-08-%') as m8, (SELECT␣
↪COUNT(DISTINCT USER_ID) as "Sep" FROM Receipts WHERE DATE_SCANNED LIKE␣
↪'%-09-%') as m9, (SELECT COUNT(DISTINCT USER_ID) as "Oct" FROM Receipts␣
↪WHERE DATE_SCANNED LIKE '%-10-%') as m10, (SELECT COUNT(DISTINCT USER_ID) as␣
↪"Nov" FROM Receipts WHERE DATE_SCANNED LIKE '%-11-%') as m11, (SELECT␣
↪COUNT(DISTINCT USER_ID) as "Dec" FROM Receipts WHERE DATE_SCANNED LIKE␣
↪'%-12-%') as m12
```

```
 * sqlite:///Data.db
Done.
```

[ ]:
```
[(97, 87, 89, 90, 88, 88, 88, 88, 88, 91, 93, 98)]
```

[ ]: