Sample SQL Queries

Prepared by: Jenna (KimHuong) Nguyen

E-mail: hn13@brandeis.edu/ hj.nguye1@gmail.com

Find all users that have never placed a sales transaction

Outer Join

```
SELECT user.id, user.name
FROM user
LEFT JOIN sale
ON user.id = sale.user_id
WHERE sale.id IS NULL
```

Subquery or anti-join

```
SELECT user.id, user.name
FROM user
WHERE user.id NOT IN
             (SELECT user.id
               FROM user
               INNER JOIN sale
               ON user.id = sale.user_id)
```

Display only items that has been purchased

Subquery or semi-join

- SELECT user.id,
- FROM user
- LEFT JOIN sale
- ON user.id = sale.user_id
- WHERE EXISTS
  - (SELECT user.id
  - FROM sale
  - INNER JOIN item
  - ON sale.id = item.sales.id)

Show all the sale_id ordered on February 01st, 2017, and user’s names for each sale transaction?

INNER JOIN

- SELECT s.id, s.date, u.name
- FROM sale s
- INNER JOIN user u
- ON u.id = s.user_id
- WHERE s.date = ‘2017-02-01’

Subquery

- SELECT s.id, s.date,
    - (SELECT u.name
    - FROM user u
    - WHERE u.id = s.user_id)
- FROM sale s
- WHERE s.date = ‘2017-02-01’

List all the users’ names and count all the sale_id they have placed in a descending order?

```
SELECT u.name,
          (SELECT COUNT(*)
          FROM sale s
          WHERE u.id = s.user_id) AS CountofOrders
FROM user
ORDER BY CountofOrders DESC
```

List users’name and all items for their last order?

```
SELECT user.name, sale.date, item.id
FROM (user
          INNER JOIN sale
          ON user.id = sale.user_id
          INNER JOIN item
          ON sale.id = item.sale_id)
WHERE sale.date = (SELECT MAX(date)
                            FROM sale s2
                            WHERE s2.user_id = user.id)
```

Rank sales rating based on sold amount on the range: Low: <500, Medium: >=500 and < 1500, High >=1500?

```
WITH sale_item AS
        (SELECT SUM(amount) AS total_sale
        FROM sale
        WHERE (sale.id = item.sale_id)
        GROUP BY date)

SELECT
SUM (CASE WHEN total_sale < 500 THEN 1 ELSE 0 END) AS Low_Sale,
SUM (CASE WHEN total_sale > = 500 AND total_sale < 1500 THEN 1 ELSE 0 END)
AS Medium_Sale,
SUM (CASE WHEN total_sale > 1500 THEN 1 ELSE 0 END) AS High_Sale
FROM sale_item
```

Ranked users spending in 2017 by device_type and total daily sales

```
 u.id, u.name d.device, s.sum(amount) AS total_daily_sale
RANK() OVER(PARTITION BY date, device_type) ORDER BY sum(amount) DESC AS user_rank
FROM (device d
        INNER JOIN user_device
        ON d.id = user_device.device_id
        INNER JOIN user u
        ON user_device.user_id = u.id
        INNER JOIN sale s
        ON user.id = s.user_id
        INNER JOIN item
        ON s.id = item.sale_id")
WHERE s.date BETWEEN ‘2017-01-01’ AND ‘2017-02-27’
GROUP BY u.id, u.name, d.device_type, sale.date
```

How many transactions took place in each of the Dallas, Texas (CITY=’DALLAS’) stores on December 24, 2004 (SALEDATE = ‘2004-12-24’)? What was the dollar value of sales in each store on that date? Your result should be a 3-column table showing the STORE number, transaction count, and SUM of AMT

```
SELECT k.store, count(distinct k.trannum) as
Transactions, sum(k.amt) as TotalSales
FROM strinfo s
INNER JOIN trnsact k ON s.store = k.store
WHERE k.saledate = '2004-12-24'
AND s.city='DALLAS'
GROUP BY k.store
ORDER BY sum(amt) desc
```

Ohio has a large number of Dillard’s locations (21 in all). Ohio is also a large and diverse state. Each store naturally does a different volume of business and local customers demand different products. While this allows local store managers to exercise their judgment about their own clientele, corporate management monitors the financial results at each store. Initially, the sales manager wants to rank the stores in Ohio from the lowest dollar volume of total sales to the highest during the year 2005. In your query, remember to include only Purchases (stype = ‘P’)

```
SELECT s.store, s.city, sum(amt) as TotalSales
FROM trnsact t
INNER JOIN strinfo s ON t.store = s.store
WHERE state = 'OH'
AND stype = 'P'
AND t.saledate BETWEEN '2005-01-01' AND '2005-12-31'
GROUP BY city, s.store
ORDER BY TotalSales
```

Compare the sales performance of two popular brands: Nine West and Nina. Devise another query to compare (state by state) total revenue from the two brands in Dillard’s stores. These totals should be aggregated and broken out by state. Your result should be a table listing all states with Dillard’s stores, and total Nina sales amounts for all 29 states, then Nine West sales amounts for all 29 states. In other words, the top 29 rows should show Nina sales, and the next 29 Nine West sales. In the database, the brands are identified as NINA FOO and NINE WES. As with the other queries, be sure to include Purchases only (not returns)

```
SELECT s.state, u.brand, sum(t.amt), count(distinct s.store)
FROM trnsact t
INNER JOIN skuinfo u ON t.sku = u.sku
INNER JOIN strinfo s ON t.store = s.store
WHERE t.stype = 'P'
AND u.brand = 'nina foo'
OR u.brand = 'nine wes'
GROUP BY s.state, u.brand
ORDER BY u.brand, s.state
```