BUS211f1 Analyzing Big Data I

Fall 2018

Project 2: Sam's Club SQL Queries

- **Some questions here refer to modifying queries from the "Intro to Sam's Club Database"**
- **Review the questions below, and begin to sketch out a query design for each one before trying to code online**
- Connect to the University of Arkansas SQL site, either using Remote Desktop Access or via the Web.
- Using SQL, type in and execute queries in response to the questions below.
- For each query, **you will paste your code and at least part of the output into the template**.
- Your Group should upload your completed file via LATTE.

Group id: J
Member names: Shengchen Fu, Zimeng Wang, Qimo Li, Chen He

Introduction: An important phase in any data analytics project is to *understand the available data within the business context.* The first several queries here are basically an exploration of the Sam's Club database.

Query 1 (15 pts, 5 pts each) – Store Visits

a. How many store visits occur in our database? Just paste in your code and describe in one sentence.

```
SELECT COUNT(visit_nbr)
FROM store_visits;
```

	Count(Visit_Nbr)
	1007961

According to the description from Sam's club ERD, the variable "visit_nbr" records every time a member goes to the register and has their membership card scanned. Therefore, we can count the total number of "visit_nbr" from "store_visits" entity to get the result that we want. There are 1007961 store visits occur in the database.

b. How many members do we have in our database?

```
SELECT COUNT(membership_nbr)
FROM member_index;
```

Count(MEMBERSHIP_NBR)
5668375

According to the description from Sam's club ERD, the variable "membership_nbr" records the number assigned to the member upon joining the club. Therefore, we can count the total number

of "member_nbr" from "member_index" to get the result that we want. There are 5668375 members in our database.

c. How many members record a transaction in any store during our sample?

```
SELECT COUNT(distinct(membership_nbr))
FROM store_visits;
```

Count(Distinct(Membership_Nbr))
377746

Since we only want to record the member who has a transaction in a particular store, we count unique membership_nbr from "store_visits" entity to get the result we want. There are 377746 members record a transaction.

Query 2 (20 pts) – Item Scans

a. How many items are recorded in the Sam's Club database? ***(5 Pts)***

```
SELECT COUNT(item_nbr)
FROM item_desc;
```

	Count(Item_Nbr)
	432223

According to the description from Sam's club ERD, the variable "item_nbr" is the number assigned to every different item for sale. Therefore, we can count the total number of "item_nbr" from "item_desc" entity to get the result we want. There are 432223 items recorded in the Sam's Club database.

b. How many different items were bought during the available date range in our sample? ***(5 Pts)***

```
SELECT COUNT(DISTINCT(item_nbr))
FROM item_scan;
```

Count(Distinct(Item_Nbr))
34250

Records of items that were bought are in the item_scan table, and this dataset uses item_nbr to define an item. We calculate the number of distinct item_nbr, and the result is 34250, which means 34250 different items were bought during the available date range in our sample.

c. Given parts (a) and (b) from Queries 1 and 2 above, comment on the sampling of the item_scan and store_visits data? ***(2 Pts)***

The number of items recorded in the Sam's Club database is 432223, and 34250 different items sold during the available date range in our sample, which is 7.92% (34250/432223) from the general item number in the database.

There are 5668375 members in our database, and 377746 of them record a transaction in any store, which is 6.66% (377746/5668375) from the general member number in the database.

d. Which variable(s) *should* identify a single row of item scan in the database? Determine the number of unique rows identified by the variable(s)? Report discrepancies, if any. ***(8 Pts)***

1. Variables

```
SELECT COUNT(DISTINCT((visit_nbr||Item_Nbr)))
FROM item_scan;
```

Count(Distinct((Visit_Nbr||Item_Nbr)))
48178564

In store_visits table, the primary key is visit_nbr. Moreover, for the item_scans table, the primary key is not only visit_nbr but the combination of visit_nbr and item_nbr, which will define a transaction on a shopping trip. Using these two variables to identify a single row, the number of rows is 48178564, which is less than the number of total rows. The discrepancy exists.

2. Discrepancy

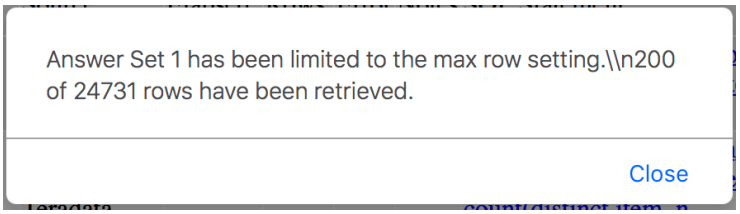
```
SELECT COUNT(*)-COUNT(DISTINCT((visit_nbr||Item_Nbr))) AS discrepancy
FROM item_scan;
```

discrepancy
26145

By subtracting the number in the last question from the total number of rows in the item_scan table, we can see there are at most 26145 rows that have the same value in the combination of visit_nbr and item_nbr. It means the same item_nbr is used more than once with the same visit_nbr.

3. Provement of reason of discrepancy

```
SELECT visit_nbr
FROM item_scan
GROUP BY visit_nbr
HAVING COUNT(item_nbr) > COUNT(DISTINCT item_nbr);
```



Answer Set 1

Visit_Nbr
244557243
260033807
264523261
258344018
260960939
237915930
259131380
253380330

There are 24731 visit_nbr that match more than 1 observation of a certain item_nbr.

Query 3 (10 pts)

a. We know from the documentation that there are multiple status codes for items, which are only indicated by a one-letter code. How many items are for each status code? (HINT: Make a 2-column table, listing each STORE_TYPE and a count.) ***(4 Pts)***

```
SELECT status_code,COUNT(*)
FROM item_desc
GROUP BY status_code;
```

Status_Code	Count(*)
A	253459
D	178764

In item_desc table, first we can know there are 2 values of status_code variable which are active represented by A and deactive represented by D. 253459 items, which is 58.6% of total items in item_desc are active.

b. Determine the **total number of item scans per status_code** in the database. Your result should be a 3-column table listing the status code, the number of scans for items for that code, total number of visits for that type. Again, what does this tell you about the sampling and the item_desc table? ***(6 Pts)***

```
SELECT status_code,COUNT(DISTINCT visit_nbr) AS
TOTAL_VISIT,COUNT(ITEM_NBR) AS TOTAL_SCAN
FROM(
SELECT  visit_nbr,item_scan.ITEM_NBR, status_code
FROM item_scan
LEFT JOIN item_desc
ON item_scan.ITEM_NBR=item_desc.ITEM_NBR) AS SUBQUERY
GROUP BY status_code
```

Status_Code	TOTAL_VISIT	TOTAL_SCAN
A	7417790	41174279
?	3534896	7030430

This result shows the sampling of the item_scan table in terms of the status_code, which means whether an item is active or inactive. More than 85% (41174279/(41174279+7030430)) of item_scan records were from items that included in the item_desc table with an 'A' in status_code. Also, these visits averagely had 5.6 items per visit.
Besides, 15% of observations were from other unknown status. Perhaps these observations' item_nbr are not included in the item_desc. Therefore the status_code of these items were missing. Also, these visits averagely had 2.0 items.

```
SELECT status_code,COUNT(DISTINCT visit_nbr) AS
TOTAL_VISIT,COUNT(ITEM_NBR) AS TOTAL_SCAN
FROM(
SELECT  visit_nbr,item_scan.ITEM_NBR, status_code
FROM item_scan
FULL JOIN item_desc
ON item_scan.ITEM_NBR=item_desc.ITEM_NBR) AS SUBQUERY
GROUP BY status_code
```

Status_Code	TOTAL_VISIT	TOTAL_SCAN
A	7417790	41174279
D	0	0
?	3534896	7030430

This result shows the sampling of the item_scan table in terms of the status_code, which means whether an item is active or inactive. More than 85% (41174279/(41174279+7030430)) of item_scan records were from items that included in the item_desc table with an 'A' in status_code. Also, these visits averagely had 5.6 items per visit.
Besides, 15% of observations were from another unknown status. Perhaps these observations' item_nbr are not included in the item_desc. Therefore the status_code of these items were missing. Also, these visits averagely had 2.0 items.
From the second table with three rows, we can understand the sampling difference between item_scan and item_desc. It shows that they do have overlap in these two entities, but each one also has a part that the other one doesn't have. In item_scan, there are no inactive(D) items being scanned. Meanwhile, some item_nbr of items purchased in item_scan are not included in item_desc.

Query 4 (10 pts) – Top 20 Categories

Get the top 20 categories in terms of number of transactions or total dollar sales. Look up the 3 top-earning categories, and describe them in a sentence. Include the code for the category exploration here, and summarize the results of the exploration in the description.

1.Using number of transactions.

```
SELECT TOP 20 category_nbr, COUNT(*) AS nbr_transactions
FROM item_scan
INNER JOIN item_desc
```

```
ON item_scan.item_nbr = item_desc.item_nbr
GROUP BY category_nbr
ORDER BY COUNT(*) DESC;
```

Category_Nbr	nbr_transactions
44	3650411
41	2584154
1	2535360
76	2494700
42	2323437
38	1942264
13	1827331
46	1753455
58	1629795
56	1453022
4	1448468
40	1258361
2	1246057
43	1226445
79	1202436
48	1185881
70	861115
45	817103
54	794406
77	756705

The second column shown in the table below represents the number of transactions. According to this table, we know that the three top-earning categories are category number 44, 41, and 1 with the number of transactions separately are 3.65 million, 2.58 million and 2.54 million.

2. Using total dollar sales

```
SELECT TOP 20 category_nbr,sum(total_scan_amount) AS total_dollar_sales
FROM item_scan
INNER JOIN item_desc
ON item_scan.item_nbr = item_desc.item_nbr
GROUP BY category_nbr
ORDER BY total_dollar_sales DESC;
```

Category_Nbr	total_dollar_sales
44	92130337.11
76	65574631.99
45	62264450.04
1	60954018.02
41	57800307.48
42	49663072.94
13	45234458.22
4	38876954.99
46	36715729.18
38	35954360.98
58	32277480.43
2	29372465.98
40	29112200.52
56	26811635.50
43	25750634.15
31	23953131.34
48	22113302.85
54	22031629.39
70	21170730.16
79	20443214.23

The second column shown in the table below represents the number of transactions. According to this table, we know that the three top-earning categories are category number 44, 76, and 45 with total dollar sales amount are separately are 92.13 million, 65.57 million and 62.26 million.

Query 5 (15 pts) – Sam's Club Membership

a. Find the category-subcategory combination(s) for which the sub-category description includes the phrase "Membership".

```
SELECT category_nbr, sub_category_nbr, sub_category_desc
FROM sub_category_desc
WHERE sub_category_desc LIKE '%Membership%'
```

Category_Nbr	Sub_Category_Nbr	Sub_Category_Desc
84	87	MEMBERSHIP FEES
73	99	MEMBERSHIP

b. Find the total transaction amount and number of transactions for items in these category 73, sub-category 99. These are annual fees paid by members. What's the annual fee per member?

```
SELECT COUNT(*) AS nbr_of_transactions, SUM(total_scan_amount) AS
total_transaction_amount, total_transaction_amount/nbr_of_transactions AS
annual_fee_per_member
FROM item_scan
WHERE item_nbr IN
(SELECT item_nbr FROM item_desc WHERE category_nbr = 73 AND
sub_category_nbr = 99)
```

	nbr_of_transactions	total_transaction_amount	annual_fee_per_member
	9580	143700.00	15.00

According to the description from Sam's Club ERD, the total_scan_amount is the total number of items scanned per visit number. Therefore, we can aggregate the total scan amount of each item scan that meets the category and subcategory requirements to get the total transaction amount which is 143700.00. Also, we can get number of transactions by counting the number of observations that satisfy the requirements and there are 9580 transactions for membership fee. Last, we divide the total transaction amount by the number of transactions to get the annual fee per member. The annual fee per member is 15.

c. Add in membership information to the table from (b), and display total membership paid for all of the membership types. Which membership type has the highest revenue?

```
SELECT member_type, sum(S.total_scan_amount) AS total_transaction_amount,
COUNT(S.visit_nbr) AS number_of_transactions
FROM member_index AS m
LEFT JOIN store_visits AS sv
ON m.membership_nbr = sv.membership_nbr
JOIN item_scan AS S
ON sv.visit_nbr = s.visit_nbr
JOIN item_desc AS D
ON S.item_nbr = D.item_nbr
WHERE category_nbr = 73 AND sub_category_nbr = 99
GROUP BY member_type
ORDER BY sum(S.total_scan_amount) DESC;
```

MEMBER_TYPE	total_transaction_amount	number_of_transactions
V	13140.00	876
W	3060.00	204
X	525.00	35
A	45.00	3
G	15.00	1

According to the description from Sam's Club ERD, member_type contains the membership information we need. Therefore, we can find the corresponding member types for the items in category 73, sub-category 99. The result above shows that membership type "V" has the highest revenue among all of the membership types.

Query 6 (10 pts; 5 pts each) – Store Sales

a. Find the top 10 stores that generate the highest membership dues. Your table should contain the store number, store name, city, state and total membership dues collected for the top 10 stores in the descending order. Your final query table will have 5 columns and 10 rows of data.

```
SELECT TOP 10 iscan.store_nbr, si.store_name, si.city, si.state,
SUM(iscan.total_scan_amount) as membership_dues
FROM item_scan AS iscan,  store_information AS si, item_desc AS idesc
WHERE iscan.store_nbr = si.store_nbr
```

```
AND iscan.item_nbr =  idesc.item_nbr
AND idesc.category_nbr = 73
AND idesc.sub_category_nbr = 99
GROUP BY iscan.store_nbr, si.store_name, si.city, si.state
ORDER BY SUM(iscan.total_scan_amount) DESC
```

Store_Nbr	Store_Name	City	State	membership_dues
39	Extreme Retailers FLORENCE, SC	FLORENCE	SC	8235.00
150	Extreme Retailers YPSILANTI, A	YPSILANTI	MI	4695.00
6	Extreme Retailers ATLANTA, WA	ATLANTA	GA	4380.00
17	Extreme Retailers CICERO, TX	CICERO	KS	2670.00
123	Extreme Retailers ST CLAIRSVIL	ST CLAIRSVILLE	OH	2295.00
143	Extreme Retailers WARWICK, CA	WARWICK	TX	2130.00
57	Extreme Retailers INVER GROVE	INVER GROVE HTS.	MN	1830.00
147	Extreme Retailers WICHITA FALL	WICHITA FALLS	TX	1785.00
136	Extreme Retailers TULSA, CO	TULSA	OK	1725.00
141	Extreme Retailers WACO, CA	WACO	TX	1710.00

The table above shows the top 10 stores that generate the highest membership dues. We consider the total transactions amount as the membership dues and get the result that NO. 39 store from City Florence in South Carolina has the highest membership dues.

b. Generate a similar list of the 10 stores that generate the highest sales. You should sum up the total revenue for each store (excluding sales tax) and list out the Store_Nbr, Store-name, City, State, and Total_Sales for the top 10 stores. Your final query table will have 5 columns and 10 rows of data. Is there any overlap between the stores in part (a) and (b)?

```
SELECT TOP 10 sv.store_nbr, si.store_name, si.city, si.state, SUM(total_visit_amt -
sales_tax_amt) as Total_Sales
FROM store_visits as sv
INNER JOIN store_information as si
ON sv.store_nbr = si.store_nbr
GROUP BY sv.store_nbr, si.store_name, si.city, si.state
ORDER BY SUM(total_visit_amt - sales_tax_amt) DESC;
```

Store_Nbr	Store_Name	City	State	Total_Sales
18	Extreme Retailers CINCINNATI,	CINCINNATI	OH	6332969.57
19	Extreme Retailers CINCINNATI,	CINCINNATI	OH	5455432.15
28	Extreme Retailers DALLAS, TX	DALLAS	TX	5437168.31
15	Extreme Retailers CHESAPEAKE,	CHESAPEAKE	VA	5193743.27
27	Extreme Retailers CRYSTAL LAKE	CRYSTAL LAKE	IL	5125138.17
24	Extreme Retailers CONCORD, TX	CONCORD	OH	5052911.90
21	Extreme Retailers CLARKSVILLE,	CLARKSVILLE	TN	4679453.39
22	Extreme Retailers CLEARWATER,	CLEARWATER	FL	4646626.57
29	Extreme Retailers DAYTONA BEAC	DAYTONA BEACH	FL	4128089.03
16	Extreme Retailers CHESTERFIELD	CHESTERFIELD	MO	3864734.89

The table above shows the top 10 stores that generate the highest sales. We consider the total sales as the sum of total value of the entire transaction (total_visit_amt) minus the sum of sales tax charged for total visit (sales_tax_amt) and get the result that the NO. 18 store from City Cincinnati in Ohio State has the highest sales. Comparing the results in (a) and (b), there's no overlap between the stores in the two tables.

Query 7 – Investigating Vendors (10 pts)

(10 pts) In this query, we focus on the volume of products from different Vendors (suppliers) in two states: Kansas (KS) and Texas (TX). In your query, you'll create a new column called Total Units, which is the sum of item_quantity.

Write a new query that sums the "item_quantity" for all items supplied by vendors to Sam's Clubs in Kansas, and lists the 10 vendors with the highest sales. Note that within our database, Vendor names have been coded as numbers, such as "Vendor_3313". Your result table should have two columns: Vendor Name and Total Units.

```
SELECT TOP 10 idesc.vendor_nbr, SUM(iscan.item_quantity)
FROM item_desc AS idesc, item_scan AS iscan, store_information AS si
WHERE idesc.item_nbr = iscan.item_nbr
AND iscan.store_nbr = si.store_nbr
AND si.state = 'KS'
GROUP BY idesc.vendor_nbr
ORDER BY SUM(iscan.item_quantity) DESC;
```

Vendor_Nbr	Sum(Item_Quantity)
6539	129065.00
11475	62160.00
14262	40581.00
15460	31747.00
3577	28199.00
17795	27861.00
4767	26481.00
11477	24088.00
297	17632.00
10161	16802.00

Next, run the same query but this time analyze sales in Texas to find the top 10 vendors in that state.

```
SELECT TOP 10 idesc.vendor_nbr, SUM(iscan.item_quantity)
FROM item_desc AS idesc, item_scan AS iscan, store_information AS si
WHERE idesc.item_nbr = iscan.item_nbr
AND iscan.store_nbr = si.store_nbr
AND si.state = 'TX'
GROUP BY idesc.vendor_nbr
ORDER BY SUM(iscan.item_quantity) DESC;
```

Vendor_Nbr	Sum(Item_Quantity)
6539	994715.00
11475	250855.00
11477	237400.00
3577	175187.00
14262	167599.00
17795	118366.00
15460	89048.00
6622	88987.00
4723	85416.00
15456	76441.00

Which vendors, if any, are in the top 10 list in both states?

As shown in the two tables above, Vendor_6539, Vendor_11475, Vendor_14262, Vendor_15460, Vendor_3577, Vendor_17795, and Vendor_11477 are in the top 10 list in both states.

Final Reflections (10 pts)

When you have completed the project, please reflect on some of the longer-lasting lessons of this experience. Most teams will make some discoveries or gain key insights either about SQL, SQL shortcuts, or the nature of data structures. Perhaps you noticed something important about this particular business. Write a thoughtful paragraph describing the team's most noteworthy and valuable discovery or insight.

After finishing this project, we have gained some insights into different dimensions:

1. Data structures

In the real world, the database such as Sam's Club has a complicated data structure. We at first need to understand the relationship between attributes and entities. Therefore, it is necessary to draw an ERD diagram before using SQL to retrieve any meaningful dataset. To fully understand the relationship, we can use different types of JOIN in SQL to explore the sampling and relationship between entities in the database. For example, we did 'LEFT JOIN' and 'FULL JOIN' in query 3 to examine the sample difference between 2 entities.

Also, through the process of manipulating data, we realized that relational databases could greatly reduce data redundancy and improve efficiency. We can easily extract the data we need for analyzing by submitting a query that follows the basic logical relationships. The data are stored more efficiently in SQL than in "flat files".

2. Interesting Discovery in this particular Business

Through our analysis in query 6 about store sales, we discovered that the high total sales do not necessarily guarantee high sales of membership dues, because there's no overlap between stores that generate the highest membership fee and the stores that generate the most top total sales. However, stores need to focus on acquiring more membership dues, because membership dues are the main source of long-term stable revenues for stores.

Another exciting discovery came from query 7. Among the top 10 vendors of each state, Kansas and Texas have 7 vendors in common. Since Kansas and Texas are both located in the central part of the U.S., the customers may have similar preferences when making purchase decisions

result from similar living habits and similar climate conditions. Also, these brands might have high reputations among residents in the central U.S.

3. SQL

Sam's Club database contains numerous amount of data, and it's impossible for us to store all the data in our laptops to do data manipulation. However, SQL allows us to do data management through a remote server and to work on the same database from different ends. Also, with the language and commands SQL provided for "queries", we can retrieve data from where we store them more efficiently.

Furthermore, we found that SQL could be more efficient if it could offer a short description for each error code instead of only showing the error code. It will save much trouble going back and forth looking up the error code, which will increase the efficiency of the debugging process.

Because of the complicated logical relationships between different tables in a SQL dataset, sometimes it's not easy to use the proper query to retrieve the data. During this case, we learned to figure out the appropriate query step by step. For example, in query 5 (b), the data we need is spread in two tables, and we need to construct a query to get from one table the attributions of items whose item numbers are regulated by information in another table. So, we first write a query to get the required item numbers and then use it as a subquery to limit the output of another query. The way of problem-solving practices our ability of logical thinking and give us a better intuition of how to construct a complex query in SQL.