# BUS211f1 Analyzing Big Data I

Fall 2018

## Project 2: Sam’s Club SQL Queries

Read through the Sam’s Club and Teradata SQL Assistant materials on our LATTE site .

* + NOTE: **The Introduction to Teradata OLAP Capabilities** and the **Intro to Sam’s Club Database** will be very helpful for this assignment. You of course should refer to DataCamp’s SQL course notes, and might also get help from <https://www.w3schools.com/sql/default.asp>.
* **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.

***EXPECT to encounter syntax or logical errors, and allow sufficient time to correct them. Implication: START EARLY!!***

ALSO note: If a query generates a table longer than 2000 rows, the SQL Web Assistant will truncate the table at 2000 rows. This is not an error or a problem and is fine for the purposes of this assignment. The History window will show how many Rows were returned in the query even if it is more than 2000.

Group id: A

Member names: Nan Cheng, Dun Lin, Shengzhe Xu, Mengyang Yao

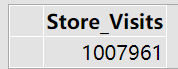
**Introduction: An important phase in any data analytics project is to *understand the available data within the business context*. The first several queries here are basically an exploration of the Sam’s Club database.**

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

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

SELECT count(\*) AS Store\_Visits

FROM store\_visits

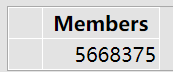


Since in the store\_visits entity, each row stands for a unique store visit, so we only have to count the rows to get the answer, which is 1007961.

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

SELECT count(\*) AS Members

FROM member\_index



Similarly, in the member\_index entity, each row contains the information of a unique member, so we only have to count the rows to get the answer, 5668375.

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

SELECT count(distinct Membership\_Nbr) AS Members\_With\_At\_Least\_One\_Transaction

FROM store\_visits



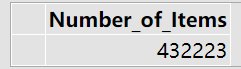
Since the same member can record multiple store visits, so we count the distinct membership number in the store\_visit entity, which yields the answer, 377746

## Query 2 (20 pts) – Item Scans

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

SELECT count(\*) AS Number\_of\_Items

FROM item\_desc



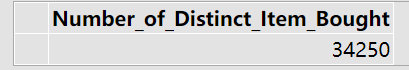
As is mentioned above, in item\_desc entity, each row stands for a unique item. Therefor we count the rows and get the answer, 432223.

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

SELECT count(distinct item\_Nbr) AS Number\_of\_Distinct\_Item\_Bought

FROM item\_scan

WHERE transaction\_Date IS NOT NULL



There are 34250 different items were bought.

1. 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)***

Only 6.66% of all the members came to store in sample period. Categories of items bought consist 7.9% of all. It is not a sample covering a big part of the database.

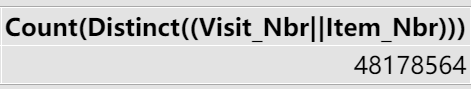
1. Which variable(s) *should* identify a single row of item scan in the database? Determine the number of unique rows identified by the variable(s)? Report discrepancies, if any. ***(8 Pts)***

Since in a single visit we can have multiple scans, we cannot use visit\_nbr to represent each row simply because one visit\_nbr will consist of a number of scan results, where the number here equals how many different items are scanned in this visit. So we can use the following:

Visit\_Nbr||Item\_Nbr: which means the combination of visit and item, which technically should be unique in the item\_scan entity.

SELECT count (distinct visit\_nbr||item\_Nbr)

FROM item\_scan



There are 48178564 different records, while 48204709 rows of item scans. There is 26145 of discrepancy, which we think the reason is as follows.In the common situation, the item\_nbr under a certain visit\_nbr should be totally different to each other, which means they should all be unique. In some particular visits, a same item appears more than once, which causes the discrepancy. We verified it with the following code:

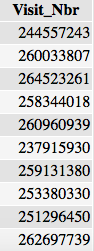
SELECT visit\_nbr

FROM item\_scan

GROUP BY visit\_nbr

HAVING count(item\_nbr)>count(distinct item\_nbr)

The code will give us a 24731 rows table, part of which is shown below. The listed Visit\_Nbrs all have repeated item\_nbrs under them!! Some of them even repeat a item more than twice!!



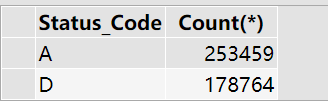
## Query 3 (10 pts)

1. 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 STATUS\_CODE and a count.) ***(4 Pts)***

SELECT Status\_Code, count(\*)

FROM item\_desc

GROUP BY Status\_Code



There are two status codes totally: 253459 items under status A, 178764 items under status D.

1. 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)***

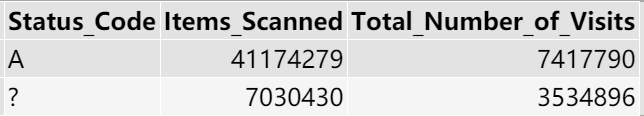
SELECT Status\_Code, count(\*) AS Items\_Scanned, count(distinct visit\_Nbr) AS Total\_Number\_of\_Visits

FROM item\_scan as s

LEFT JOIN item\_desc as d

ON s.item\_Nbr = d.item\_Nbr

GROUP BY Status\_Code



The reason for A and “?”, instead of A and D, shown in the result is that all the items of category D are not scanned.

For sampling, we may have selected a biased sample, which contains only items of status A, according to result.

Or there is a possibility that the status code in item description is not correctly classified. Category D may contain few categories of products available. “?” means many items can’t be found in item description, which means they are not belong to any status code.

## Query 4 (10 pts) – Top 20 Categories

Get the top 20 categories in terms of number of transactions or total dollar sales. Your result should be a 2-column table listing the category number, and the number of transactions/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.

SELECT TOP 20 CATEGORY\_NBR, COUNT(d.ITEM\_NBR) AS Number\_of\_Transactions

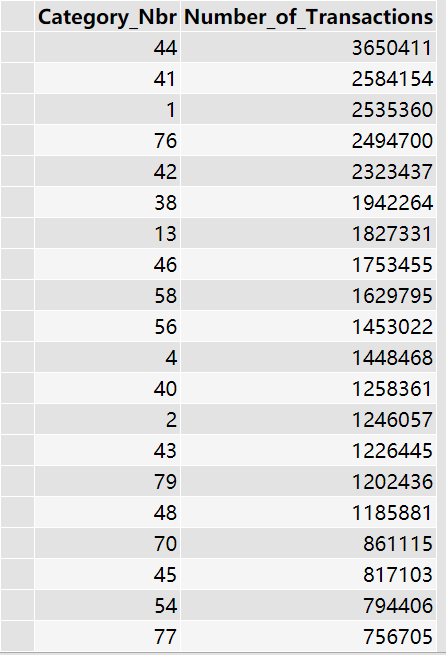
FROM ITEM\_SCAN AS s

INNER JOIN ITEM\_DESC AS d

ON s. ITEM\_NBR = d. ITEM\_NBR

GROUP BY CATEGORY\_NBR

ORDER BY Number\_of\_Transactions DESC



SELECT TOP 20 CATEGORY\_NBR, SUM(total\_scan\_amount) AS Total\_Dollar\_Sales

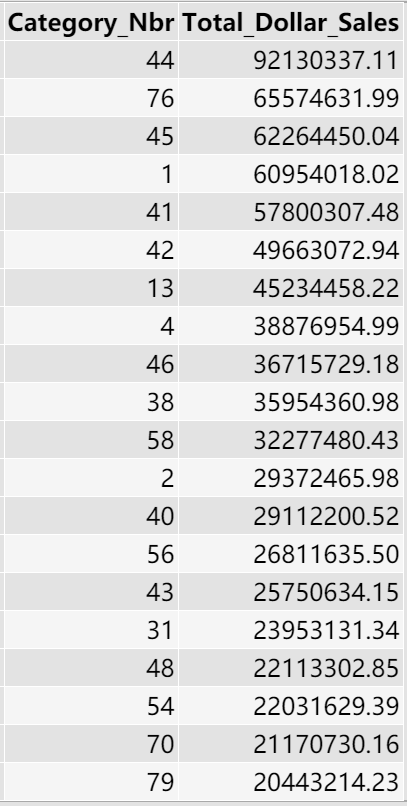
FROM ITEM\_SCAN AS s

INNER JOIN ITEM\_DESC AS d

ON s.ITEM\_NBR = d.ITEM\_NBR

GROUP BY CATEGORY\_NBR

ORDER BY Total\_Dollar\_Sales DESC



The three top categories in sales volume are 44,41,1, while the three top categories in sales revenue are 44,76,45. This may convey the information that the unit price of category 45 is high, because its number of transactions is among the lowest but the revenue is among the top. And the unit price of category 41 and 1 is low.

We think that the second way, which is to see the total revenue of each category is more reasonable because the target of sales is the most profitable items.

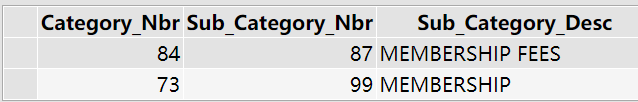
## Query 5 (15 pts) – Sam’s Club Membership

1. Find the category-subcategory combination(s) for which the sub-category description includes the phrase “Membership”. Your result should be a 3-column table with category number, sub-category number and sub-category description.

SELECT category\_Nbr, sub\_category\_Nbr, sub\_category\_desc

FROM sub\_category\_desc

WHERE sub\_category\_desc LIKE '%Membership%'



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

SELECT category\_Nbr, sub\_category\_Nbr,SUM(total\_scan\_amount) As Total\_Transaction\_Amount,COUNT(s.item\_Nbr) AS Number\_Of\_Transactions

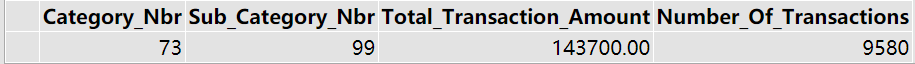
FROM item\_scan as s

INNER JOIN item\_desc as d

ON s.item\_Nbr= d.item\_Nbr

GROUP BY category\_Nbr, sub\_category\_Nbr

WHERE category\_Nbr = 73 AND sub\_category\_Nbr = 99



Annual fee per member: 143700/9580=15

1. Add in membership information to the table from (b), and display total membership paid for all of the membership types. Your result should be a 3-column table with membership type, total transaction amount and number of transactions. Which membership type has the highest revenue?

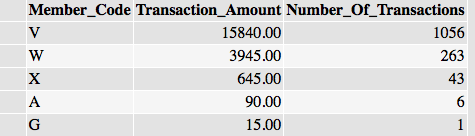
SELECT v.Member\_Code,SUM(s.total\_scan\_amount) as Transaction\_Amount, COUNT(s.item\_Nbr) AS Number\_Of\_Transactions

FROM item\_scan as s,item\_desc as d,store\_visits as v

WHERE s.Item\_Nbr=d.Item\_Nbr AND s.Visit\_Nbr=v.Visit\_Nbr AND d.Category\_Nbr=73 AND d.sub\_category\_Nbr=99

GROUP BY d.Category\_Nbr,d.sub\_category\_Nbr,v.Member\_Code

ORDER BY Transaction\_Amount desc



Member type V has the greatest revenue.

Note: As we inner joined three tables together, some information were lost. Every time when we are joining two tables, rows in one table that do not have a corresponding row, based on the joining criteria, in the second table, are deleted. When we join item\_scan and item\_desc together, some of the data lost but it is totally acceptable. But when we try to join item\_scan and store\_visits, so many rows, above 50% in the item\_scan entity, are lost. This means that for a majority records of transaction, we cannot locate their visit\_nbr in the store\_visits entity. This is why the sum of transaction amount and number of transactions of these 5 membership types in this table do not equal to the two numbers in the previous table.

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

1. 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.

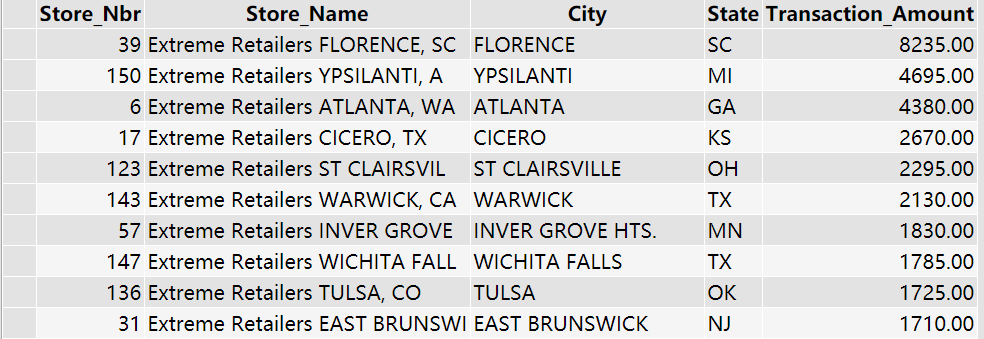
SELECT TOP 10 i.Store\_Nbr, i.Store\_Name, i.City, i.State, SUM(s.Total\_Scan\_Amount) as Transaction\_Amount

FROM item\_scan as s, store\_information as i, item\_desc as d

WHERE s.Store\_Nbr = i.Store\_Nbr AND s.Item\_Nbr = d.Item\_Nbr AND d.Category\_Nbr = 73 AND d.Sub\_Category\_Nbr = 99

GROUP BY i.Store\_Nbr, i.Store\_Name, i.City, i.State

ORDER BY Transaction\_Amount desc



1. 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)?

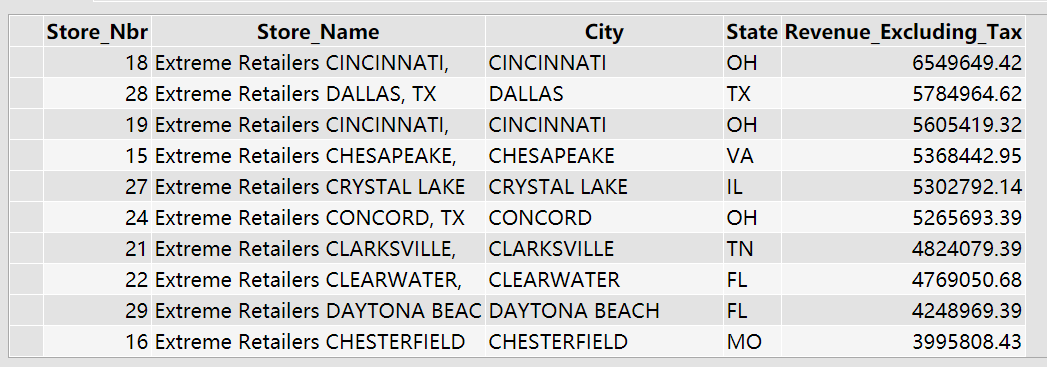
SELECT TOP 10 i.Store\_Nbr, i.Store\_Name, i.City, i.State, SUM(v.Total\_Visit\_Amt) as Revenue\_Excluding\_Tax

FROM store\_information as i, store\_visits as v

WHERE i.Store\_Nbr = v.Store\_Nbr

GROUP BY i.Store\_Nbr,i.Store\_Name, i.City, i.State

ORDER BY Revenue\_Excluding\_Tax desc



There is no overlap at all between the two tables. This is saying that stores with the highest membership dues are not necessarily the ones that have the highest revenue.

## 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.

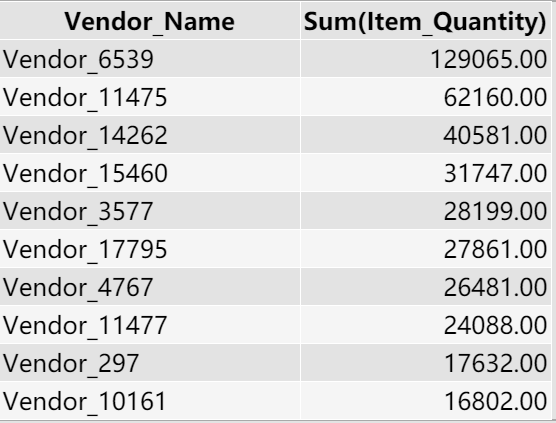
Write a new query that sums the “item\_quantity” for all items supplied by vendors to Sam’s Clubs in Kansas, and lists the 10 vendors with the highest sales. Note that within our database, Vendor names have been coded as numbers, such as “Vendor\_3313”. Your result table should have two columns: Vendor Name and Total Units.

SELECT TOP 10 vendor\_name, sum(item\_quantity)

FROM item\_scan, item\_desc, store\_information

WHERE item\_scan.item\_nbr = item\_desc.item\_nbr AND item\_scan.store\_nbr = store\_information.store\_nbr AND state = 'KS'

GROUP BY vendor\_name

ORDER BY sum(item\_quantity) DESC; 

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

SELECT TOP 10 vendor\_name, sum(item\_quantity)

FROM item\_scan, item\_desc, store\_information

WHERE item\_scan.item\_nbr = item\_desc.item\_nbr AND item\_scan.store\_nbr = store\_information.store\_nbr AND state = 'TX'

GROUP BY vendor\_name

ORDER BY sum(item\_quantity) DESC;

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

Vendor\_6539, Vendor\_11475, Vendor\_11477, Vendor\_3577, Vendor\_14262, Vendor\_17795, Vendor\_15460 are among the top 10s 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.

SQL is a useful tool looking up information but not as easy as we thought. It requires a strong and smooth logic flow in order to maximize efficiency and minimize errors, because errors in SQL, at least in teradata, is costly and hard to recognize.

As a result, It is essential to be familiar with shortcuts and syntax of SQL, as it will speed up the process significantly, for example, using aliases to shorten query, or minimizing number of tables we need to join. Moreover, we need to know what attributes each table includes and what they represent. We also need to deal with missing data, especially when it comes to joining multiple tables together.