**Team BRAMS**

**Members:** Bryan Vu, Rachel Pai, Autumn Nguyen,   
Marjorie Balaoro and Saurav Chhapawala  
CECS 323 Term Project: SQL Queries Output

**SQL Queries**

**-- 1. List the customers. For each customer, indicate which category he or she fall into, and**

**-- his or her contact information. If you have more than one independent categorization of customers, please indicate which category the customer falls into for all of the**

**-- categorizations**

SELECT Cust.CUSTOMERID AS 'CUSTOMER ID', Cust.CUSTOMERNAME AS 'CUSTOMER NAME', 'PRIVATE CUSTOMER, INDIVIDUAL' AS CATEGORY

FROM Customers Cust

INNER JOIN CustomerAccounts USING (customerID)

WHERE CustomerType = "Private"

UNION

SELECT c.customerID AS 'CUSTOMER ID', c.customerName AS 'CUSTOMER NAME', 'ANONYMOUS' AS CATEGORY

FROM Customers c

LEFT JOIN CustomerAccounts USING (customerID)

WHERE CustomerAccounts.CustomerID IS NULL

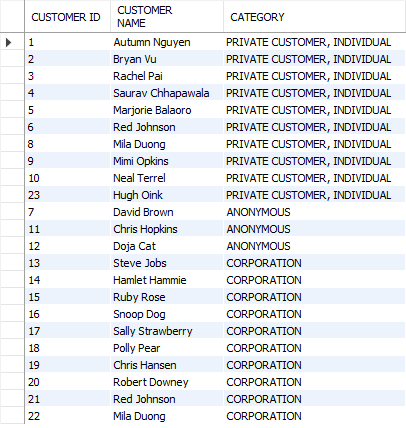
UNION

SELECT Customers.CustomerID AS 'CUSTOMER ID', Customers.CustomerName AS 'CUSTOMER NAME', 'CORPORATION' AS CATEGORY

FROM Customers Customers

RIGHT JOIN CustomerAccounts USING (customerID)

WHERE CustomerAccounts.CorporationName IS NOT NULL;



**-- 2. List the top three customers in terms of their net spending for the past two years (last**

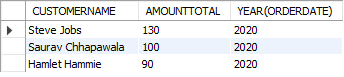
**-- 730 days), and the total that they have spent in that period.**

SELECT CUSTOMERNAME, AMOUNTTOTAL, YEAR(ORDERDATE)

FROM CUSTOMERORDER

ORDER BY AMOUNTTOTAL DESC

LIMIT 3;



**-- 3. Find all of the sous chefs who have three or more menu items that they can prepare. For**

**-- each sous chef, list their name, the number of menu items that they can prepare, and**

**-- each of the menu items. You can use group\_concat to get all of a given sous chef’s data**

**-- on one row, or print out one row per sous chef per menu item.**

SELECT SOUSCHEFMENTEEID, FIRSTNAME, LASTNAME, COUNT(MENUITEMID) AS 'Items Mastered', MENUITEMNAME

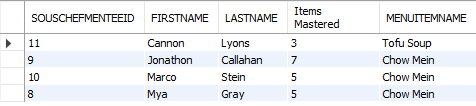
FROM EXPERTISE

INNER JOIN SOUSCHEF ON (EXPERTISE.SOUSCHEFMENTEEID = SOUSCHEF.STAFFID)

INNER JOIN STAFF ON (SOUSCHEF.STAFFID = STAFF.STAFFID)

GROUP BY FIRSTNAME, LASTNAME

HAVING COUNT(SOUSCHEFMENTEEID) >= 3;



**-- 4. Find all of the sous chefs who have three or more menu items in common.**

**-- i. Please give the name of each of the two sous chefs sharing three or more menu items.**

**-- ii. Please make sure that any given pair of sous chefs only shows up once.**

**-- iii. Please list the items that the two Sous Chefs have in common. Again, you can use group\_concat to get all of those items into one value in the output.**

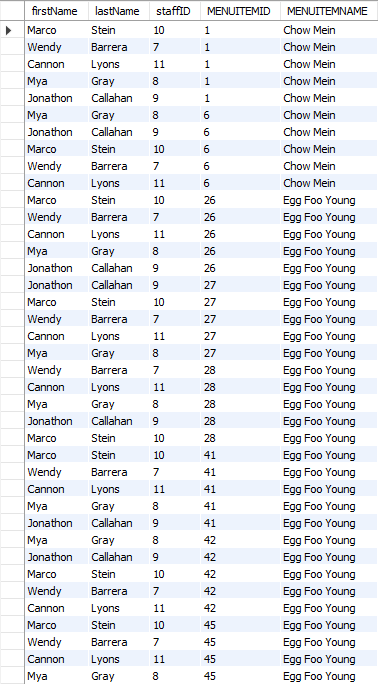
SELECT DISTINCT STAFF.firstName, STAFF.lastName, SOUSCHEF.staffID, EXPERTISE.MENUITEMID, MENUITEMNAME

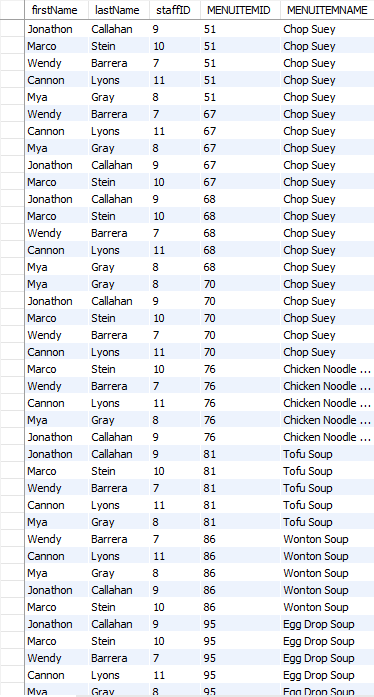
FROM SOUSCHEF

INNER JOIN STAFF USING (STAFFID)

INNER JOIN EXPERTISE ON SOUSCHEFMENTEEID

ORDER BY MENUITEMID, MENUITEMNAME;





**-- 5. Find the three menu items most often ordered from the Children's menu and order them from most frequently ordered to least frequently ordered.**

SELECT DISTINCT ORDERDETAIL.MENUITEMNAME, MENUITEMID, ITEMSPECIFICATION.AGEGROUP, COUNT(ORDERDETAIL.MENUITEMNAME) AS 'Number of Purchases'

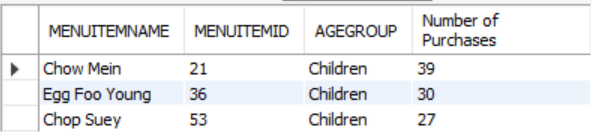
FROM ORDERDETAIL INNER JOIN ITEMSPECIFICATION USING (MENUITEMID)

WHERE AGEGROUP = 'CHILDREN'

GROUP BY MENUITEMNAME

ORDER BY COUNT(ORDERDETAIL.MENUITEMNAME) DESC

LIMIT 3;

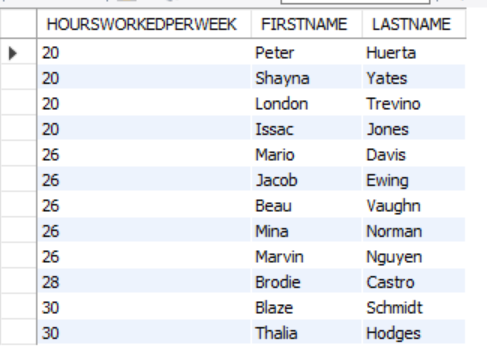
****

**-- 6. Show by week, how many hours each employee works.**

SELECT STAFFID, FIRSTNAME, LASTNAME, HOURSWORKEDPERWEEK

FROM PARTTIMESTAFF INNER JOIN STAFF USING (STAFFID)

ORDER BY HOURSWORKEDPERWEEK ASC;

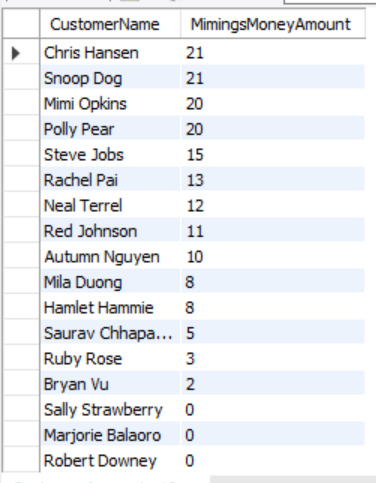


**-- 7. List the customers, sorted by the amount of Miming's Money that they have, from largest to smallest.**

SELECT CustomerName, MimingsMoneyAmount

FROM CustomerAccounts

ORDER BY MIMINGSMONEYAMOUNT DESC;

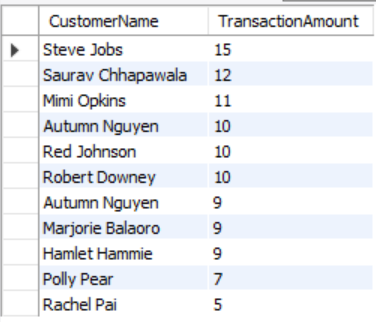


**-- 8. List the customers and the total that they have spent at Miming's ever, in descending order by the amount that they have spent.**

SELECT CustomerName, TransactionAmount

FROM CustomerAccounts INNER JOIN MimingsMoneyTransactionLogs USING (CustomerID)

ORDER BY TransactionAmount DESC;



**-- 9. Report on the customers at Miming's by the number of TIMES that they came in by month, order by most to least frequent**

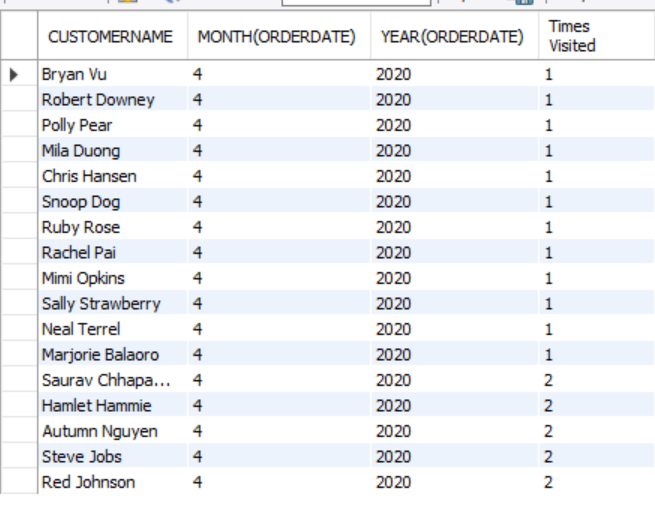
**-- Each row in the output should have the Customer name, month, year, and number of times they came in per month**

SELECT CUSTOMERNAME, MONTH(ORDERDATE), YEAR(ORDERDATE), COUNT(DISTINCT ORDERID) AS 'Times Visited'

FROM CUSTOMERORDER

GROUP BY CUSTOMERNAME

ORDER BY COUNT(DISTINCT ORDERID);



**-- 10. List the three customers who have spent the most at Miming's over the past year (365 days).**

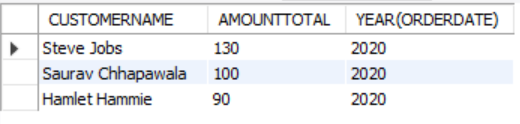
**-- Order by the amount that they have spent, from largest to smallest.**

SELECT CUSTOMERNAME, AMOUNTTOTAL, YEAR(ORDERDATE)

FROM CUSTOMERORDER

ORDER BY AMOUNTTOTAL DESC

LIMIT 3;



**-- 11. List the five menu items who have spent the most at Miming's over the past year**

**-- Order by the amount that they spent from largest to smallest**

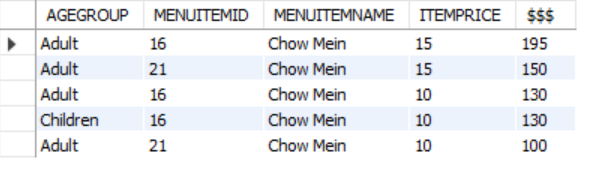
SELECT DISTINCT AGEGROUP, MENUITEMID, ITEMSPECIFICATION.MENUITEMNAME, ITEMSPECIFICATION.ITEMPRICE, (ITEMPRICE \* ORDERDETAIL.QUANTITY) AS '$$$'

FROM ITEMSPECIFICATION INNER JOIN ORDERDETAIL USING (MENUITEMID)

ORDER BY (ITEMPRICE \* ORDERDETAIL.QUANTITY) DESC

LIMIT 5;

NOTE: All of them are the same because of a design error that we had.



**-- 12**

**Find the sous chef who is mentoring the most other sous chef. List the menu items that the sous chef is passing along to the other sous chefs.**

SELECT STAFFID, FIRSTNAME, LASTNAME,

count(STAFFID) AS 'Chefs mentored', MENUITEMNAME

FROM STAFF

INNER JOIN MENTORSHIP ON STAFFID = SOUSCHEFMENTORID

WHERE STAFFID IN (

SELECT SOUSCHEFMENTORID FROM MENTORSHIP

GROUP BY SOUSCHEFMENTORID

HAVING count(SOUSCHEFMENTEEID) = (

SELECT max(A.SOUSCHEFMENTORED) FROM (

SELECT SOUSCHEFMENTORID, count(SOUSCHEFMENTEEID)

AS SOUSCHEFMENTORED

FROM MENTORSHIP

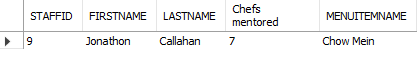
GROUP BY SOUSCHEFMENTORID

) AS A

)

)

GROUP BY STAFFID;



**-- 13**

**Find the three menu items that have the fewest sous chefs skilled in those menu items.**

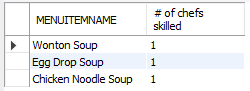
SELECT MENUITEMNAME, count(MENUITEMNAME) AS '# of chefs skilled'

FROM EXPERTISE

GROUP BY MENUITEMNAME

ORDER BY count(MENUITEMNAME) ASC

LIMIT 3;



**-- 14**

**List all of the customers who eat at Miming’s on their own as well as ordering for their corporation.**

SELECT CORP.CUSTOMERNAME, CORP.CORPORATIONNAME,

if(PRIVATE.EMAIL IS NULL, 'NO', 'YES') AS 'PRIVATE CUSTOMER' FROM

(SELECT CUSTOMERID, CUSTOMERNAME, CUSTOMERTYPE, CORPORATIONNAME, EMAIL

FROM CustomerAccounts

WHERE CUSTOMERTYPE = 'Corporation') AS CORP

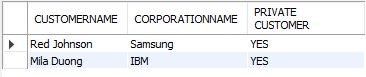
INNER JOIN

(SELECT CUSTOMERID, CUSTOMERNAME, CUSTOMERTYPE, CORPORATIONNAME, EMAIL

FROM CustomerAccounts

WHERE CUSTOMERTYPE = 'Private') AS PRIVATE

USING(CUSTOMERNAME);



**-- 15**

**List the contents and prices of each of the menus.**

SELECT AGEGROUP, MENUITEMNAME, PORTIONSIZE, ITEMPRICE,

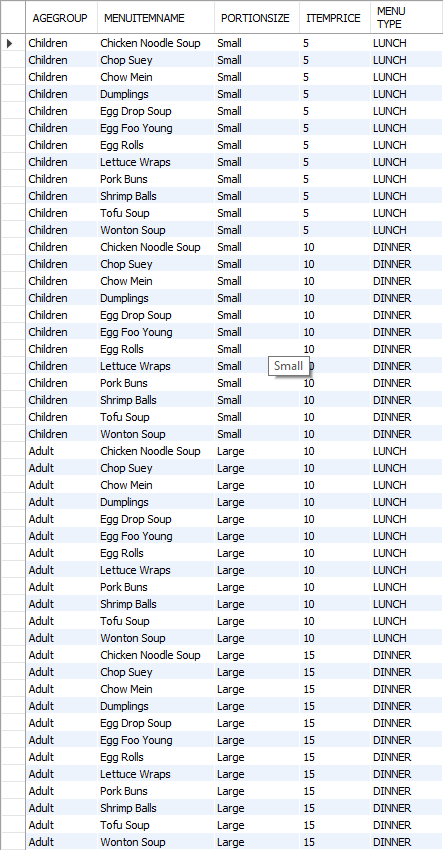
if(MENUID = 2 OR MENUID = 5, 'DINNER', 'LUNCH') AS 'MENU TYPE'

FROM ITEMSPECIFICATION

WHERE MENUID != 3 AND MENUID != 6

GROUP BY MENUITEMNAME, PORTIONSIZE, ITEMPRICE

ORDER BY MENUID, MENUITEMNAME;



-- 16

Three additional queries that demonstrate the five additional business rules. Feel free to create additional views to support these queries if you so desire.

**-- Business rule 1**

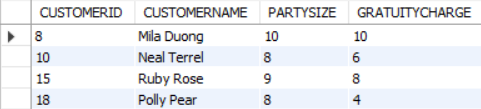
**Groups that dine in with more than 8 guests will be added an automatic 20% gratuity fee to their total check.--**

SELECT CUSTOMERID, CUSTOMERNAME, PARTYSIZE, GRATUITYCHARGE

FROM CUSTOMERORDER

INNER JOIN EATINORDER USING(ORDERID)

WHERE PARTYSIZE >= 8;



**-- Business rule 5**

**The start date of a sous chef learning a recipe must be unique (ie, they cannot start learning more than 1 recipe a day).**

SELECT STAFFID, FIRSTNAME, LASTNAME, STARTDATE

FROM STAFF

INNER JOIN SOUSCHEF USING(STAFFID)

INNER JOIN MENTORSHIP ON SOUSCHEF.STAFFID = MENTORSHIP.SOUSCHEFMENTEEID

ORDER BY STARTDATE;

