Homework 3: Complex SQL
GROUP 9

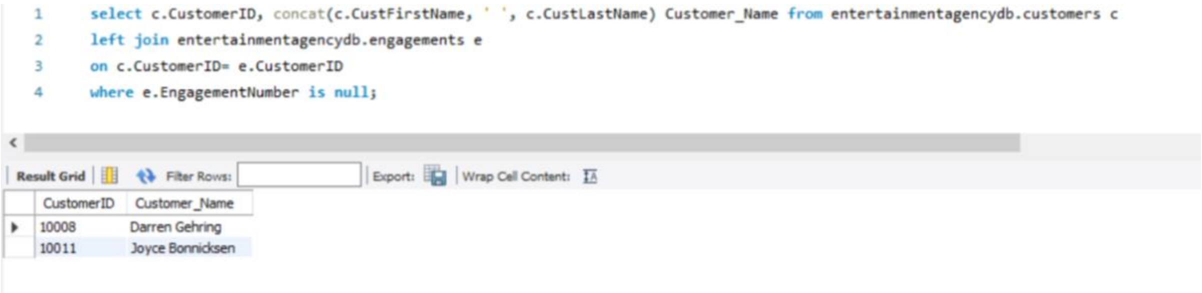
1. **List Customers with No Bookings**

TRANSLATION:

Select Customer Id from the Entertainment Agency Database and join it to the Customer ID where Engagement number is stated as null.

CLEAN UP:

Select Customer Id from the Entertainment Agency Database and join ~~it to~~ the Customer ID where Engagement number is ~~stated as~~ null.



```sql
select c.CustomerID, concat(c.CustFirstName, ' ', c.CustLastName) Customer_Name from entertainmentagencydb.customers c
left join entertainmentagencydb.engagements e
on c.CustomerID= e.CustomerID
where e.EngagementNumber is null;
```

CustomerID	Customer_Name
10008	Darren Gehring
10011	Joyce Bonnicksen

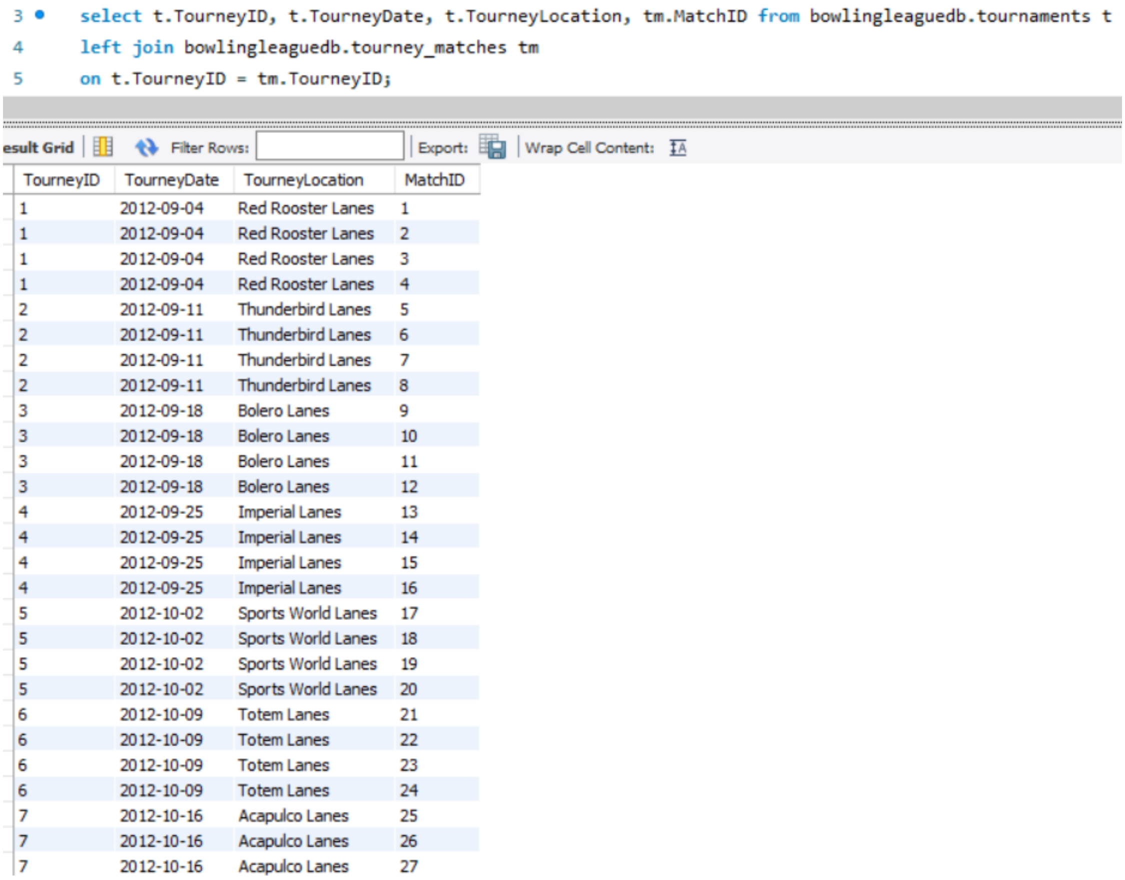
2. **Display all tournaments and any matches that have been played**

TRANSLATION:

Select all the tournaments played in the bowling league from the tournaments table and all the matches played in the bowling league from the tourney_matches table and join the two on the TourneyID column.

CLEAN UP:

Select ~~all the~~ tournaments ~~played in the bowling league~~ from ~~the~~ tournaments table ~~and all the~~ matches ~~played in the bowling league~~ from the tourney_matches table ~~and~~ join ~~the two~~ on the TourneyID column.



```
select t.TourneyID, t.TourneyDate, t.TourneyLocation, tm.MatchID from bowlingleaguedb.tournaments t
left join bowlingleaguedb.tourney_matches tm
on t.TourneyID = tm.TourneyID;
```

TourneyID	TourneyDate	TourneyLocation	MatchID
1	2012-09-04	Red Rooster Lanes	1
1	2012-09-04	Red Rooster Lanes	2
1	2012-09-04	Red Rooster Lanes	3
1	2012-09-04	Red Rooster Lanes	4
2	2012-09-11	Thunderbird Lanes	5
2	2012-09-11	Thunderbird Lanes	6
2	2012-09-11	Thunderbird Lanes	7
2	2012-09-11	Thunderbird Lanes	8
3	2012-09-18	Bolero Lanes	9
3	2012-09-18	Bolero Lanes	10
3	2012-09-18	Bolero Lanes	11
3	2012-09-18	Bolero Lanes	12
4	2012-09-25	Imperial Lanes	13
4	2012-09-25	Imperial Lanes	14
4	2012-09-25	Imperial Lanes	15
4	2012-09-25	Imperial Lanes	16
5	2012-10-02	Sports World Lanes	17
5	2012-10-02	Sports World Lanes	18
5	2012-10-02	Sports World Lanes	19
5	2012-10-02	Sports World Lanes	20
6	2012-10-09	Totem Lanes	21
6	2012-10-09	Totem Lanes	22
6	2012-10-09	Totem Lanes	23
6	2012-10-09	Totem Lanes	24
7	2012-10-16	Acapulco Lanes	25
7	2012-10-16	Acapulco Lanes	26
7	2012-10-16	Acapulco Lanes	27

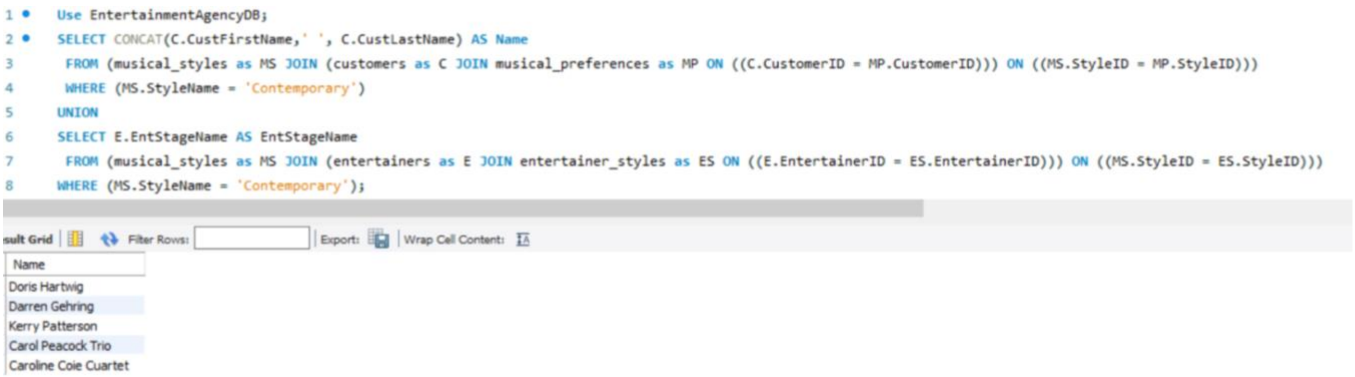
3. **Produce a list of customers who like contemporary music together with a list of entertainers who play contemporary music**

TRANSLATION:

Select distinct names of customers, names of entertainers and style from the Entertainment Agency Database, join the Engagements table and Customers table with Customer ID. Join the Entertainers table and Engagements table with Entertainment ID. Join the Entertainer_Styles table and the Entertainer table with Entertainment ID where style ID is 6 or 8. Order the customer names in ascending order.

CLEAN UP:

Select distinct names ~~of~~ customers, names of entertainers ~~and~~ style from the Entertainment Agency Database, join the Engagements ~~table and~~ Customers table with Customer ID. Join the Entertainers ~~table and~~ Engagements table with Entertainment ID. Join the Entertainer_Styles ~~table and the~~ Entertainer table with Entertainment ID where style ID is 6 or 8. Order the customer names in ascending order.



```sql
Use EntertainmentAgencyDB;
SELECT CONCAT(C.CustFirstName,' ', C.CustLastName) AS Name
 FROM (musical_styles as MS JOIN (customers as C JOIN musical_preferences as MP ON ((C.CustomerID = MP.CustomerID))) ON ((MS.StyleID = MP.StyleID)))
 WHERE (MS.StyleName = 'Contemporary')
UNION
SELECT E.EntStageName AS EntStageName
 FROM (musical_styles as MS JOIN (entertainers as E JOIN entertainer_styles as ES ON ((E.EntertainerID = ES.EntertainerID))) ON ((MS.StyleID = ES.StyleID)))
WHERE (MS.StyleName = 'Contemporary');
```

Name
Doris Hartwig
Darren Gehring
Kerry Patterson
Carol Peacock Trio
Caroline Coie Cuartet

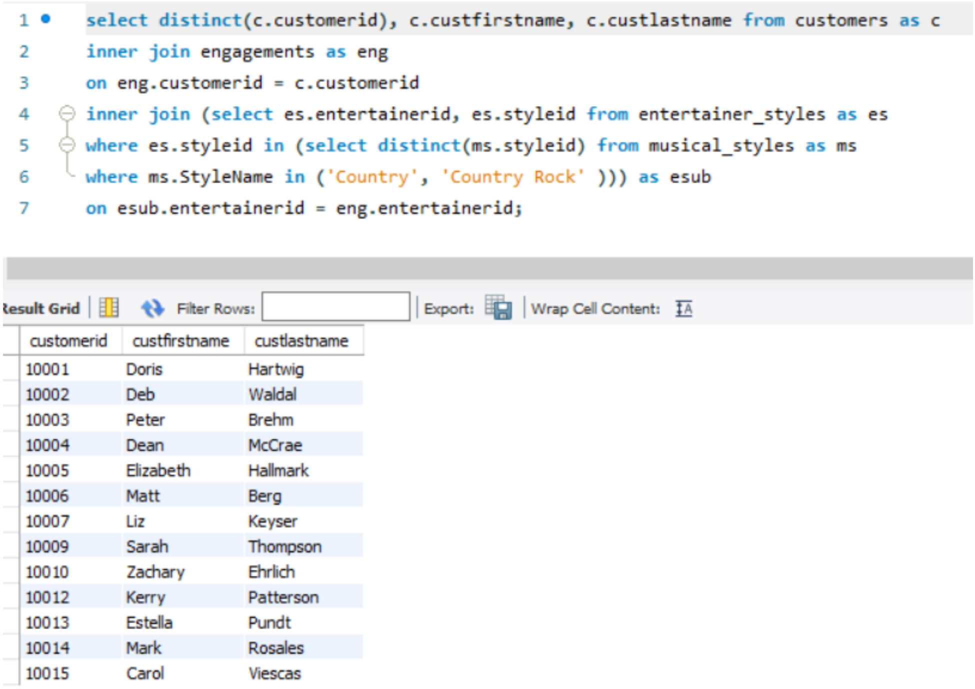
4. **List customers who have booked entertainers who play country and rock music**

TRANSLATION:

Select distinct features Customer ID, Customer Name, Customer Last Name from Customers table joined with engagements table in engagement customer ID table matching customer ID where style ID is treated as ms and style name is treated as esub table matching Entertainment ID.

CLEAN UP:

Select distinct ~~features~~ Customer ID, Customer Name, Customer Last Name from Customers table joined ~~with~~ engagements table in engagement customer ID ~~table matching~~ customer ID where style ID ~~is treated~~ as ms and style name ~~is treated~~ as esub table matching Entertainment ID.



```sql
select distinct(c.customerid), c.custfirstname, c.custlastname from customers as c
inner join engagements as eng
on eng.customerid = c.customerid
inner join (select es.entertainerid, es.styleid from entertainer_styles as es
where es.styleid in (select distinct(ms.styleid) from musical_styles as ms
where ms.StyleName in ('Country', 'Country Rock' ))) as esub
on esub.entertainerid = eng.entertainerid;
```

customerid	custfirstname	custlastname
10001	Doris	Hartwig
10002	Deb	Waldal
10003	Peter	Brehm
10004	Dean	McCrae
10005	Elizabeth	Hallmark
10006	Matt	Berg
10007	Liz	Keyser
10009	Sarah	Thompson
10010	Zachary	Ehrlich
10012	Kerry	Patterson
10013	Estella	Pundt
10014	Mark	Rosales
10015	Carol	Viescas

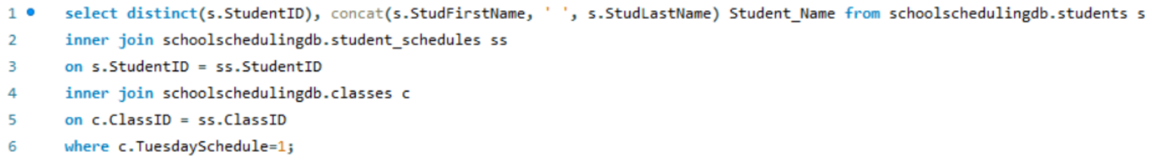
5. **Display students enrolled in a class on Tuesday**

TRANSLATION:

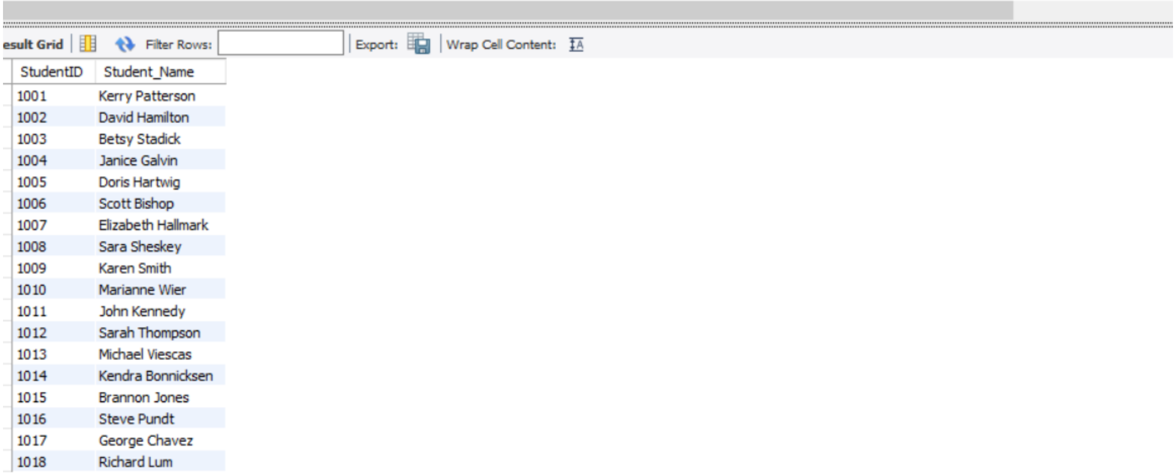
Select student ID, student name from students table joined with student schedules table on student id in student table matching student id in student schedules table joined with classes table on class id in classes table matching class id in student schedules table where Tuesday schedule in classes table = 1

CLEAN UP:

Select student ID, student name from students ~~table~~ joined ~~with~~ student schedules ~~table~~ on student id in student ~~table~~ matching student id in student schedules ~~table~~ joined with classes ~~table~~ on class id in classes ~~table~~ matching class id in student schedules ~~table~~ where Tuesday schedule in classes ~~table~~ = 1



```
select distinct(s.StudentID), concat(s.StudFirstName, ' ', s.StudLastName) Student_Name from schoolschedulingdb.students s
inner join schoolschedulingdb.student_schedules ss
on s.StudentID = ss.StudentID
inner join schoolschedulingdb.classes c
on c.ClassID = ss.ClassID
where c.TuesdaySchedule=1;
```



StudentID	Student_Name
1001	Kerry Patterson
1002	David Hamilton
1003	Betsy Stadick
1004	Janice Galvin
1005	Doris Hartwig
1006	Scott Bishop
1007	Elizabeth Hallmark
1008	Sara Sheskey
1009	Karen Smith
1010	Marianne Wier
1011	John Kennedy
1012	Sarah Thompson
1013	Michael Viescas
1014	Kendra Bonnicksen
1015	Brannon Jones
1016	Steve Pundt
1017	George Chavez
1018	Richard Lum

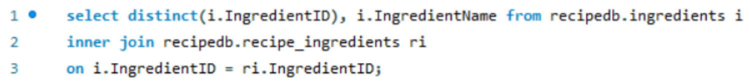
6. **List the ingredients that are used in some recipe where the measurement amount in the recipe is not the default measurement amount**

TRANSLATION:

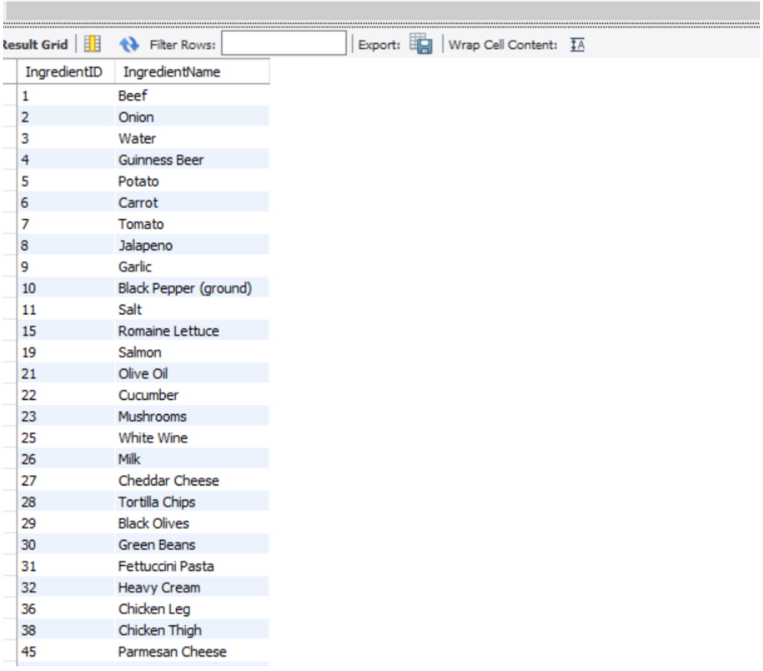
Select ingredient ID, ingredient name from ingredients table joined with recipe ingredients table on ingredient id in ingredients table matching ingredient id in recipe ingredients table

CLEAN UP:

Select ingredient ID, ingredient name from ingredients ~~table~~ joined ~~with~~ recipe ingredients ~~table on~~ ingredient id in ingredients ~~table~~ matching ingredient id in recipe ingredients ~~table~~



```
select distinct(i.IngredientID), i.IngredientName from recipedb.ingredients i
inner join recipedb.recipe_ingredients ri
on i.IngredientID = ri.IngredientID;
```



IngredientID	IngredientName
1	Beef
2	Onion
3	Water
4	Guinness Beer
5	Potato
6	Carrot
7	Tomato
8	Jalapeno
9	Garlic
10	Black Pepper (ground)
11	Salt
15	Romaine Lettuce
19	Salmon
21	Olive Oil
22	Cucumber
23	Mushrooms
25	White Wine
26	Milk
27	Cheddar Cheese
28	Tortilla Chips
29	Black Olives
30	Green Beans
31	Fettuccini Pasta
32	Heavy Cream
36	Chicken Leg
38	Chicken Thigh
45	Parmesan Cheese

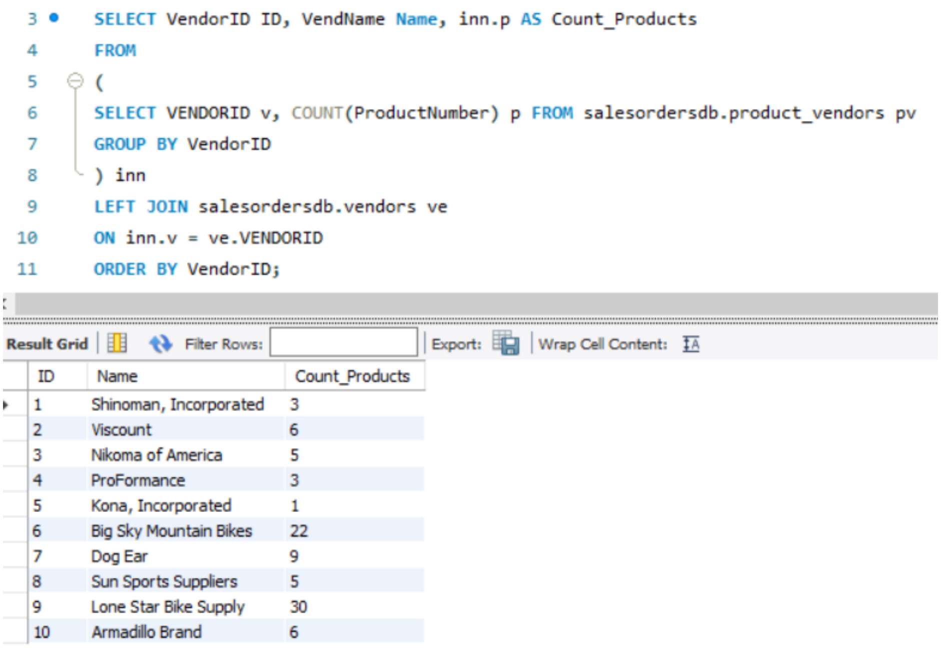
7. **List all vendors and count of products sold by each**

TRANSLATION:

Select vendor ID, Vendor name and count of products from table made with selecting vendor ID and count of product number from product vendors grouped by vendor ID joined with vendors table ordered by vendor ID.

CLEAN UP:

Select vendor ID, Vendor name ~~and~~ count~~-of~~ products from ~~table made with~~ selecting vendor ID and count ~~of~~ product number from product vendors group~~ed~~ by vendor ID join~~ed with~~ vendors table order~~ed~~ by vendor ID.



```sql
SELECT VendorID ID, VendName Name, inn.p AS Count_Products
FROM
(
SELECT VENDORID v, COUNT(ProductNumber) p FROM salesordersdb.product_vendors pv
GROUP BY VendorID
) inn
LEFT JOIN salesordersdb.vendors ve
ON inn.v = ve.VENDORID
ORDER BY VendorID;
```

ID	Name	Count_Products
1	Shinoman, Incorporated	3
2	Viscount	6
3	Nikoma of America	5
4	ProFormance	3
5	Kona, Incorporated	1
6	Big Sky Mountain Bikes	22
7	Dog Ear	9
8	Sun Sports Suppliers	5
9	Lone Star Bike Supply	30
10	Armadillo Brand	6

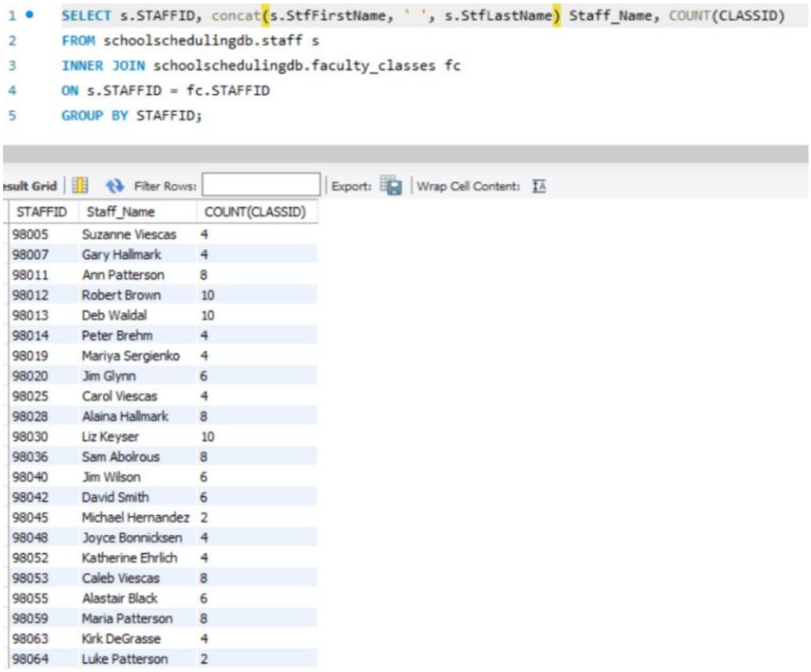
8. **List each staff member and the count of products sold by each**

TRANSLATION:

Select staff ID, first name and last name and count of class ID from Staff table joined with faculty classes table on staff id in staff table matching staff id in faculty classes. Group by staff ID.

CLEAN UP:

Select staff ID, first name ~~and~~ last name ~~and~~ count ~~of~~ class ID from Staff table joined ~~with~~ faculty classes ~~table~~ on staff id in staff ~~table~~ ~~matching~~ staff id in faculty classes. Group by staff ID.



```sql
SELECT s.STAFFID, concat(s.StfFirstName, ' ', s.StfLastName) Staff_Name, COUNT(CLASSID)
FROM schoolschedulingdb.staff s
INNER JOIN schoolschedulingdb.faculty_classes fc
ON s.STAFFID = fc.STAFFID
GROUP BY STAFFID;
```

STAFFID	Staff_Name	COUNT(CLASSID)
98005	Suzanne Viescas	4
98007	Gary Hallmark	4
98011	Ann Patterson	8
98012	Robert Brown	10
98013	Deb Waldal	10
98014	Peter Brehm	4
98019	Mariya Sergienko	4
98020	Jim Glynn	6
98025	Carol Viescas	4
98028	Alaina Hallmark	8
98030	Liz Keyser	10
98036	Sam Abolrous	8
98040	Jim Wilson	6
98042	David Smith	6
98045	Michael Hernandez	2
98048	Joyce Bonnicksen	4
98052	Katherine Ehrlich	4
98053	Caleb Viescas	8
98055	Alastair Black	6
98059	Maria Patterson	8
98063	Kirk DeGrasse	4
98064	Luke Patterson	2

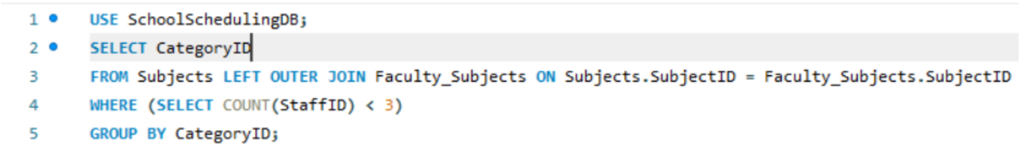
9. **Show me the subject categories that have fewer than three full professors teaching that subject**

TRANSLATION:

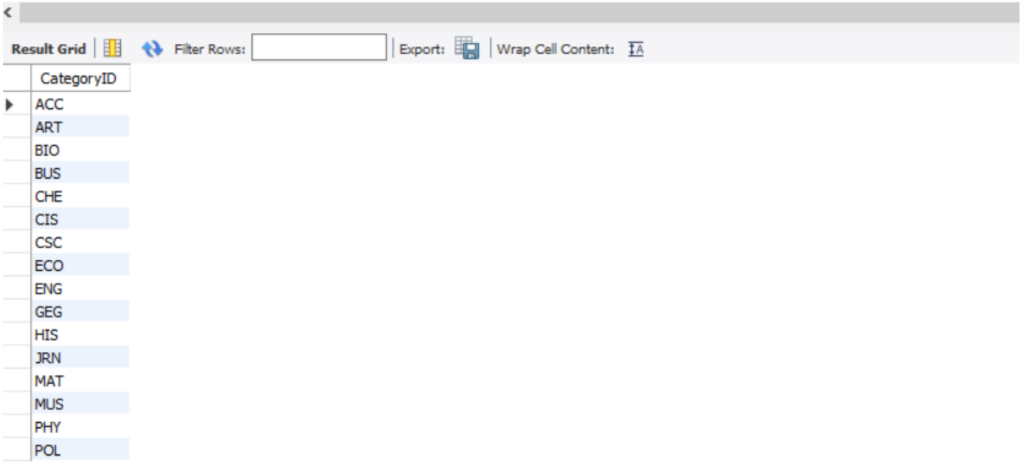
Select CategoryID from Subjects table joined with faculty subjects table on subject ID in subjects table matching subject ID in faculty table where count of staff ID is less than 3. Group by category ID

CLEAN UP:

Select CategoryID from Subjects ~~table~~ joined ~~with~~ faculty subjects ~~table~~ on subject ID in subjects ~~table~~ matching subject ID in faculty ~~table~~ where count ~~of~~ staff ID ~~is~~ less than 3. Group by category ID



```
USE SchoolSchedulingDB;
SELECT CategoryID
FROM Subjects LEFT OUTER JOIN Faculty_Subjects ON Subjects.SubjectID = Faculty_Subjects.SubjectID
WHERE (SELECT COUNT(StaffID) < 3)
GROUP BY CategoryID;
```



CategoryID
ACC
ART
BIO
BUS
CHE
CIS
CSC
ECO
ENG
GEG
HIS
JRN
MAT
MUS
PHY
POL