**Queries**  
  
View

1. CREATE VIEW MenuItem\_v AS

SELECT a.dishname, Group\_Concat(DISTINCT a.dishprice, " ", a.size) AS 'Child Menu',

GROUP\_CONCAT(DISTINCT b.dishprice, " ", b.size) AS 'Lunch Menu',

GROUP\_CONCAT(DISTINCT c.dishprice, " ", c.size) AS 'Evening Menu',

'N/A' AS 'Buffet Menu',

GROUP\_CONCAT(DISTINCT spicyness)

FROM MenuDish a LEFT OUTER JOIN MenuDish b ON a.dishname = b.dishname

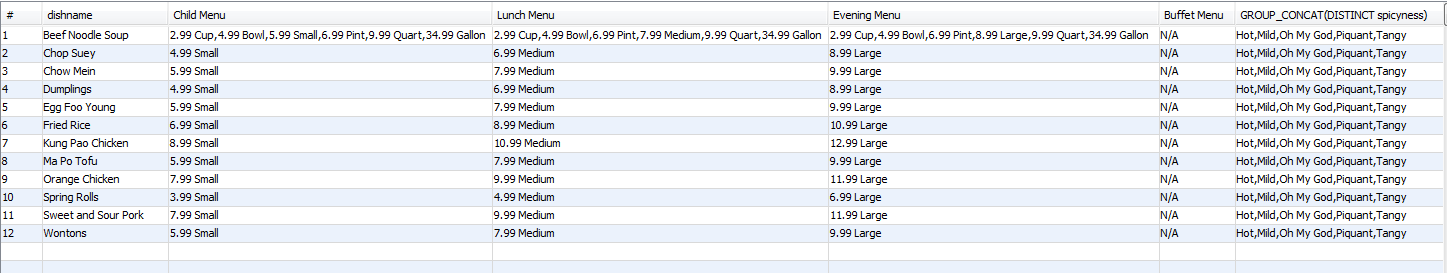
LEFT OUTER JOIN MenuDish c ON a.dishname = c.dishname

CROSS JOIN Spicyness

WHERE a.menutype = 'Children' AND b.menutype = 'Lunch' AND c.menutype = "Evening"

GROUP BY a.dishname;

GRANT SELECT ON MenuItem\_v TO public;



2.  
CREATE VIEW Customer\_addresses\_v AS

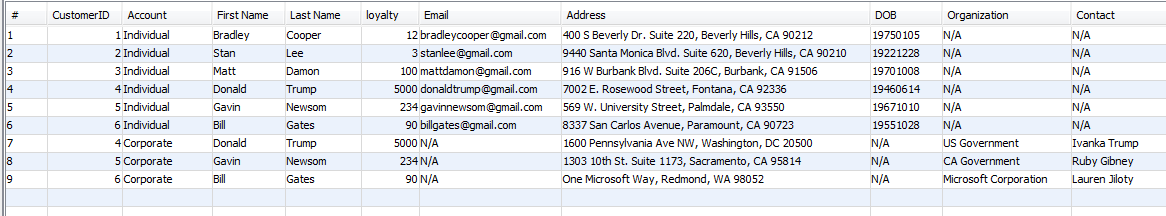
SELECT c.CustomerID, 'Individual' AS "Account", FName AS "First Name", LName AS "Last Name", loyalty, Email, SnailMail AS "Address", DOB, 'N/A' AS "Organization", 'N/A' AS Contact

FROM Customer c INNER JOIN Private p ON c.CustomerID = p.CustomerID

UNION

SELECT c1.CustomerID, 'Corporate' AS "Account", FName, LName, loyalty, 'N/A' AS "Email", Address, 'N/A' AS "DOB", Organization, Contact

FROM Customer c1 INNER JOIN Corporation c2 ON c1.CustomerID = c2.CustomerID;

GRANT SELECT ON Customer\_addresses\_v TO public;  


3.CREATE VIEW Sous\_mentor\_v AS

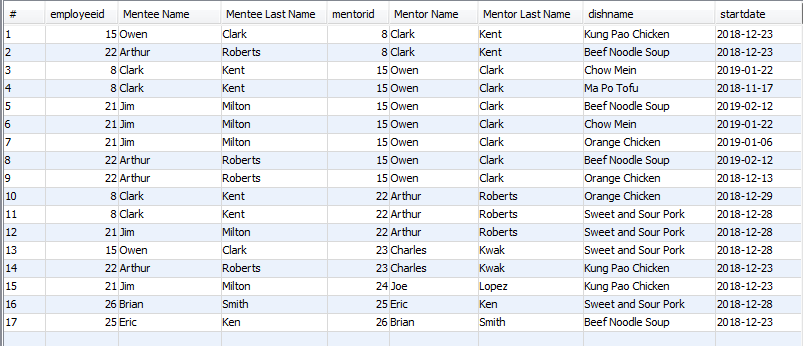
SELECT m.employeeid, e.fname AS "Mentee Name", e.lname AS "Mentee Last Name",

m.mentorid, s.fname AS "Mentor Name", s.lname AS "Mentor Last Name", dishname, startdate

FROM Expertises m INNER JOIN Employees e ON m.employeeid = e.employeeid

INNER JOIN Employees s ON m.mentorid = s.employeeid;

GRANT SELECT ON Sous\_mentor\_v TO public;



4.create view Customer\_Sales\_v as

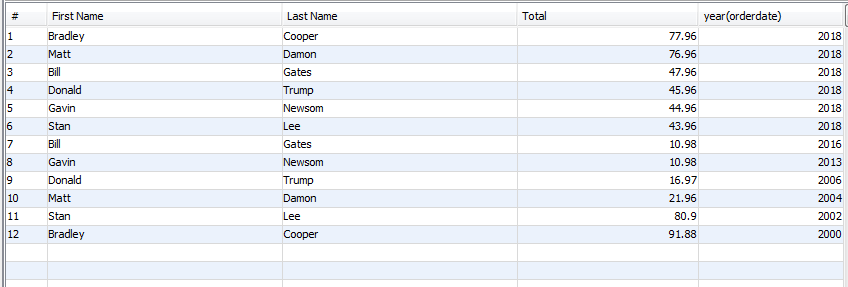
select fname as "First Name", lname as "Last Name", round(sum(menuprice),2) as Total, year(orderdate) from

(select fname, lname, menuprice, orderdate from Customer natural join Orders natural join OrderDish natural join MenuDish natural join Menu where menuprice <> 0 group by customerid, orderdate

union

select fname, lname, sum(dishprice\*quantity), orderdate from Customer natural join Orders natural join OrderDish natural join MenuDish group by fname,lname, orderdate) as ut

group by year(orderdate),fname, lname order by year(orderdate) desc, total desc;



5. create view Customer\_Value\_v as

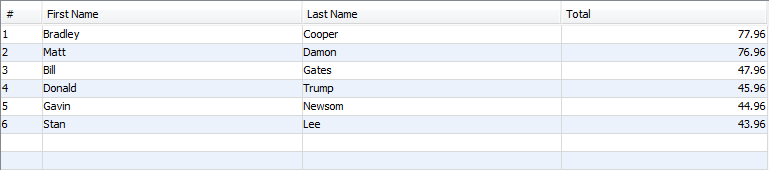
select fname as "First Name", lname as "Last Name", round(sum(menuprice),2) as Total from

(select fname, lname, menuprice from Customer natural join Orders natural join OrderDish natural join MenuDish natural join Menu where menuprice <> 0 group by customerid, orderdate

union

select dt.fname, dt.lname, sum(dt.dishprice\*dt.quantity) from (select fname,lname, dishprice, quantity from Customer natural join Orders natural join OrderDish natural join MenuDish where year(orderdate) = year(now())-1) as dt group by dt.fname,dt.lname) as ut

group by fname,lname order by total desc;



Queries

1. SELECT Customer.CustomerID AS "ID", LName, FName, "Private" AS "Kind"

FROM Customer

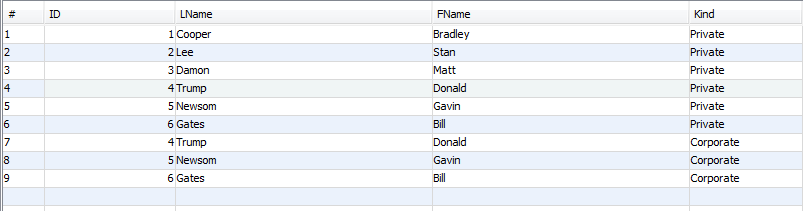
INNER JOIN Private ON Customer.CustomerID=Private.CustomerID

UNION

SELECT Customer.CustomerID, LName, FName, "Corporate" AS "Kind"

FROM Customer

INNER JOIN Corporation ON Customer.CustomerID=Corporation.CustomerID;

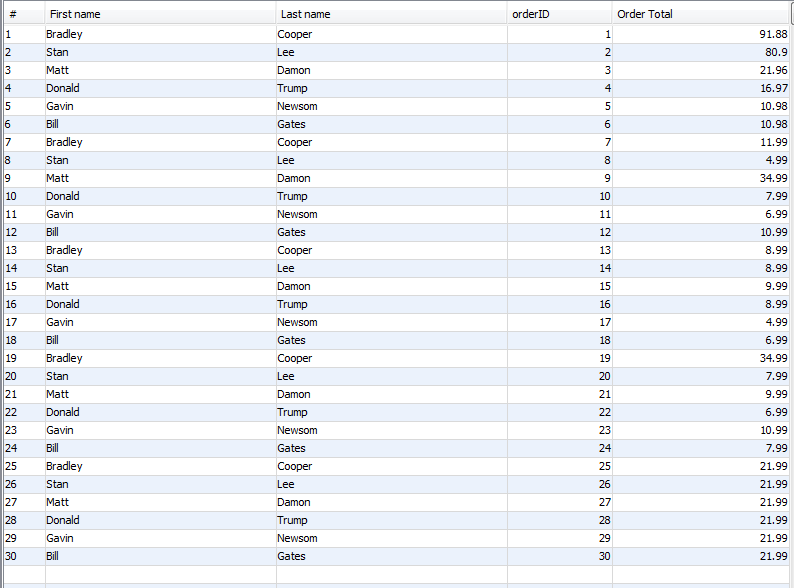


2.

select fname as "First name", lname as "Last name", orderID, round(sum(quantity\*dishprice) + menuprice,2) as "Order Total"

from Orders natural join OrderDish natural join MenuDish natural join Customer natural join Menu

group by orderid;



3. SELECT Customer.CustomerID, LName, FName, DishName, SUM(DishPrice) AS “Total Spent”, OrderDate

FROM Customer

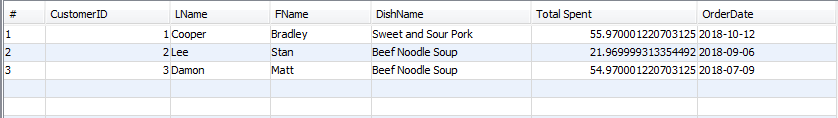
INNER JOIN Orders ON Orders.CustomerID=Customer.CustomerID

INNER JOIN OrderDish ON Orders.OrderID=OrderDish.OrderID

INNER JOIN MenuDish ON OrderDish.md\_id = MenuDish.md\_id

WHERE OrderDate <= CURDATE() AND OrderDate >= DATE\_SUB(CURDATE(), INTERVAL 2 YEAR)

GROUP BY Customer.CustomerID LIMIT 0, 3;



4. SELECT employeeid, fname, lname, GROUP\_CONCAT(DISTINCT dishname)

FROM Expertises NATURAL JOIN Employees

GROUP BY employeeid

HAVING Count(DISTINCT dishname) >= 3

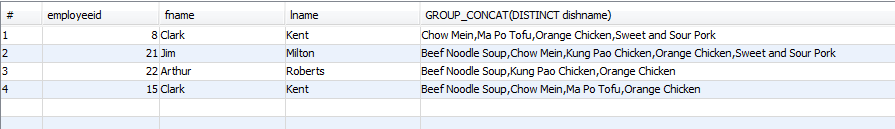
UNION

SELECT mentorid, fname, lname, GROUP\_CONCAT(DISTINCT dishname)

FROM Expertises NATURAL JOIN Employees

GROUP BY mentorid

HAVING COUNT(DISTINCT dishname) >= 3;



5. SELECT A.employeeid as "Sous Chef 1 ID", A.fname, A.lname, B.Employeeid as "Sous Chef 2 ID", B.fname, B.lname, GROUP\_CONCAT(A.dishname) as "Dish Name"

FROM

(SELECT Expertises.employeeid, fname, lname, dishname FROM Expertises INNER JOIN Employees ON Expertises.employeeid = Employees.employeeid) A

INNER JOIN

(SELECT Expertises.employeeid, fname, lname, dishname FROM Expertises INNER JOIN Employees ON Expertises.employeeid = Employees.employeeid) B

ON A.dishname = B.dishname AND A.employeeid < B.employeeid

WHERE A.employeeid IS NOT NULL AND B.employeeid IS NOT NULL

GROUP BY A.employeeid, A.fname, A.lname, B.employeeid, B.fname, B.lname HAVING COUNT(A.dishname) >= 3;

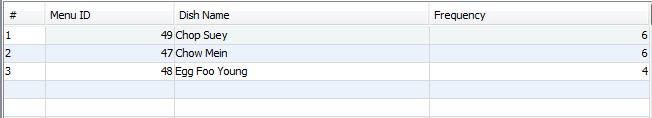


6. select md\_id as "Menu ID", DishName as "Dish Name", count(md\_id) as Frequency

from OrderDish natural join MenuDish group by md\_id

having md\_id in (select md\_id from MenuDish where MenuType = 'Children')

order by Frequency desc limit 3;



7. SELECT DISTINCT dishname, ordertime, orderdate, orderid, employeeid, fname as "First Name", lname as "Last Name"

FROM OrderDish NATURAL JOIN Orders NATURAL JOIN MenuDish

LEFT OUTER JOIN Employees ON shift = 'Morning'

NATURAL JOIN SousChefs NATURAL JOIN Expertises

WHERE Expertises.enddate > orderdate AND ordertime <= '13:00:00'

UNION ALL

SELECT DISTINCT dishname, ordertime, orderdate, orderid, employeeid, fname, lname

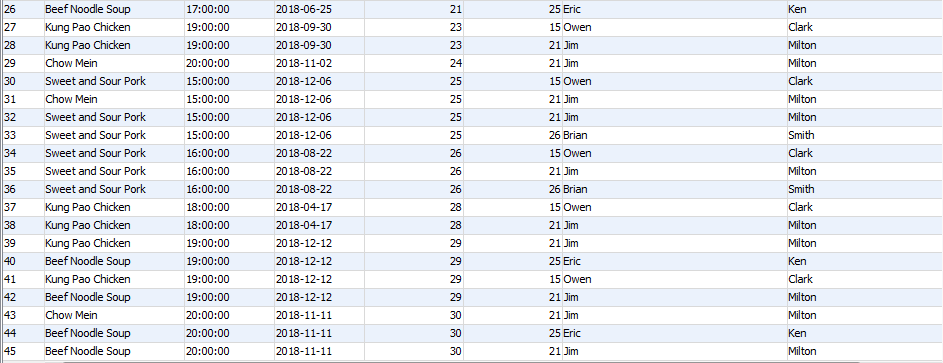
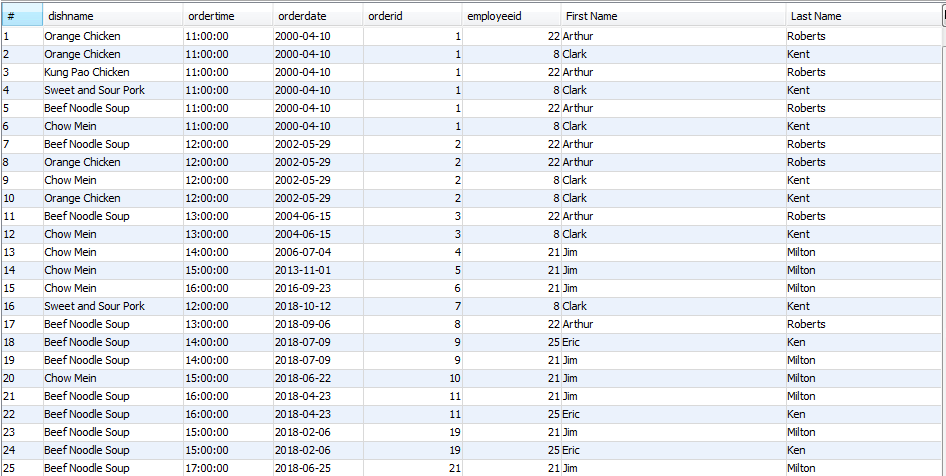
FROM OrderDish NATURAL JOIN Orders NATURAL JOIN MenuDish

LEFT OUTER JOIN Employees ON shift = 'Evening'

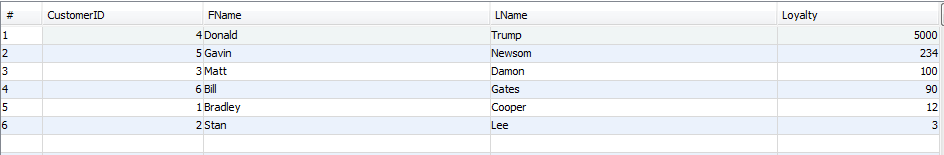
NATURAL JOIN SousChefs NATURAL JOIN Expertises

WHERE Expertises.enddate > orderdate AND ordertime > '13:00:00'

ORDER BY orderid;



8. SELECT \* FROM Customer ORDER BY loyalty DESC;



9.SELECT Customer.CustomerID AS "ID", LName as “Last Name”, FName as “First Name”, ROUND(SUM(DishPrice),2) AS "Total"

FROM Customer

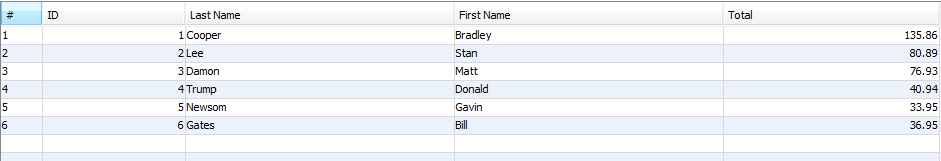
INNER JOIN Orders ON Customer.CustomerID=Orders.CustomerID

INNER JOIN OrderDish ON Orders.OrderID=OrderDish.OrderID

INNER JOIN MenuDish ON OrderDish.md\_id=MenuDish.md\_id

GROUP BY Customer.CustomerID

ORDER BY "Total";



10. SELECT COUNT(OrderID) AS Orders, LName as “Last Name”, FName as “First Name”, MONTH(OrderDate) AS "Month", YEAR(OrderDate) AS "Year"

FROM Orders

INNER JOIN Customer ON Customer.CustomerID=Orders.CustomerID

GROUP BY Orders.CustomerID, YEAR(OrderDate), MONTH(OrderDate)

ORDER BY Orders DESC;



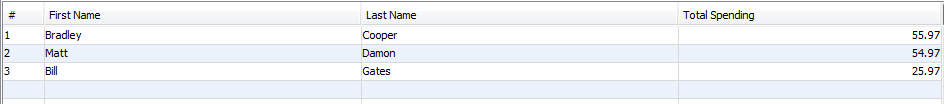
11. select fname as "First Name", lname as "Last Name", round(sum(quantity\*dishprice)+menuprice,2) as "Total Spending"

from Customer natural join Orders natural join OrderDish natural join MenuDish natural join Menu

where year(orderdate) = year(now())-1

group by customerid

order by sum(dishprice\*quantity)+menuprice desc limit 3;

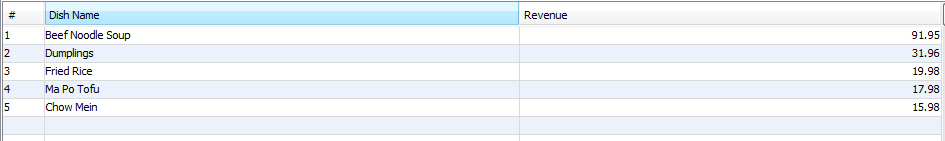


12.select dt.dishname as "Dish Name", round(sum(dt.quantity\*dt.dishprice),2) as Revenue

from (select \* from Orders natural join OrderDish natural join MenuDish where year(orderdate) = year(now())-1) as dt

group by dishname

order by Revenue desc limit 5;



13. SELECT mentorid, fname as “First Name”, lname as “Last Name”, GROUP\_CONCAT(DISTINCT dishname), count(\*) AS "mentor times"

FROM Expertises LEFT OUTER JOIN Employees e ON mentorid = e.employeeid GROUP BY mentorid

HAVING count(\*) =

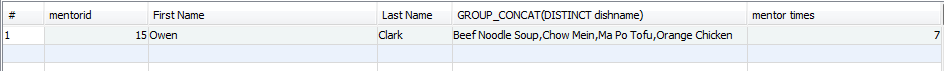
(SELECT MAX(mycount) as highest

FROM (SELECT mentorid, COUNT(\*) mycount

FROM Expertises

GROUP BY mentorid)

as t);

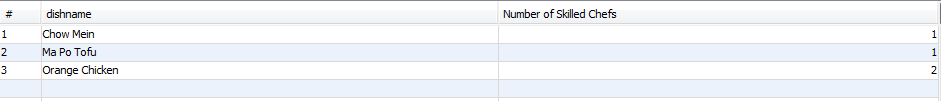


14. select dishname, count(mentorid) as "Number of Skilled Chefs"

from (select distinct dishname, mentorid from Expertises) as dt

group by dishname

order by count(mentorid) asc limit 3;



15. select dt1.customerid as "Customer ID", dt1.lname as "Last Name", dt2.fname as "First Name"

from (select customerid, lname, fname from Customer natural join Private) as dt1

inner join (select customerid, lname, fname from Customer natural join Corporation) as dt2

on dt1.customerid = dt2.customerid;



16. select menutype as "Menu", dishname as "Dish Name", dishprice as "Price"

from MenuDish order by menutype, dishname;



17. SELECT employeeid, fname, lname, "Manager" AS "Job", shift

FROM Managers NATURAL JOIN Employees

UNION

SELECT employeeid, fname, lname, "Head Chef", shift

FROM HeadChefs NATURAL JOIN Employees

UNION

SELECT employeeid, fname, lname, "Sous Chef", shift

FROM SousChefs NATURAL JOIN Employees

UNION

SELECT employeeid, fname, lname, "Line Cook", shift

FROM LineCooks NATURAL JOIN Employees

UNION

SELECT employeeid, fname, lname, "Maitre D", shift

FROM MaitreDs NATURAL JOIN Employees

UNION

SELECT employeeid, fname, lname, "Waiter", shift

FROM WaitStaff NATURAL JOIN Employees

UNION

SELECT employeeid, fname, lname, "Dishwasher", shift

FROM DishWashers NATURAL JOIN Employees

ORDER BY shift DESC, Job;



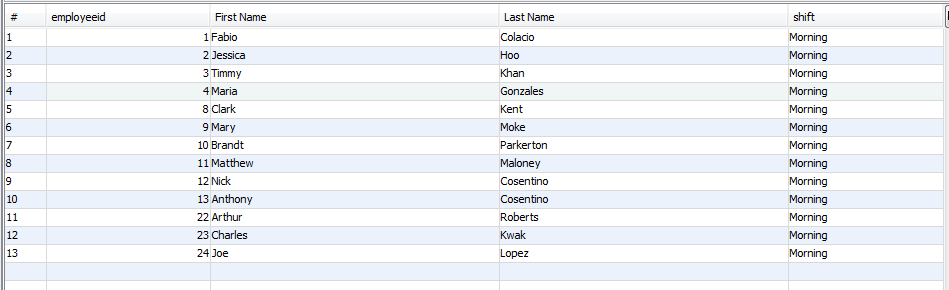
18.

Gather all employees whose crew rating is above a 4

SELECT employeeid, fname as “First Name”, lname as “Last Name”, shift

FROM Employees NATURAL JOIN Shift

WHERE crewrating >= 4;



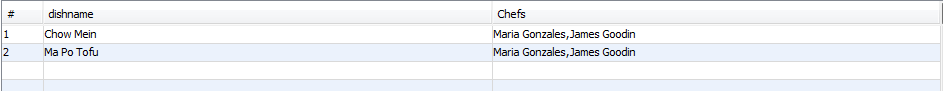
Gather all dishes that had multiple chefs creating them

SELECT dishname, GROUP\_CONCAT(fname, " ", lname) AS Chefs

FROM CreatorRoles NATURAL JOIN Employees

GROUP BY dishname

HAVING COUNT(employeeid) >= 2;



Cash Payment 5% discount (excluding Sunday brunch buffet)

select fname as "First Name", lname as "Last Name", orderid, round(sum(dishprice\*quantity),2) as "Total", round(sum(dishprice\*quantity)\*0.95,2) as "Discounted Total"

from Customer natural join Orders natural join OrderDish natural join MenuDish

where paymenttype = 'Cash' and menutype <> 'Sunday brunch buffet' group by orderid;

