CS262- Problem Set 1

CS262- Database Systems
2021-CS-190 — Abdul Mateen

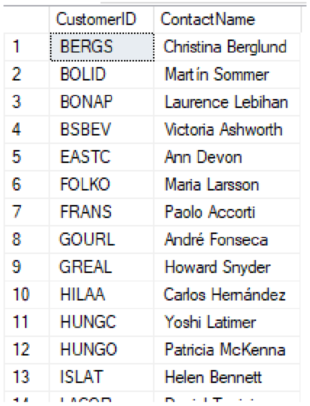
March 1, 2023

1 NorthWind

Problem 1. Give the names of customers whose orders were delayed

Solution.

```
SELECT CustomerID,ContactName
FROM Customers
WHERE CustomerID IN (
  SELECT CustomerID
  FROM Orders
  WHERE ShippedDate > RequiredDate
)
```



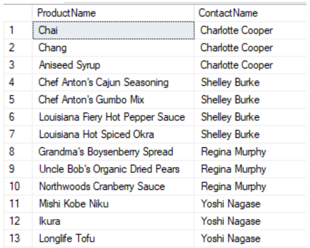
	CustomerID	ContactName
1	BERGS	Christina Berglund
2	BOLID	Martín Sommer
3	BONAP	Laurence Lebihan
4	BSBEV	Victoria Ashworth
5	EASTC	Ann Devon
6	FOLKO	Maria Larsson
7	FRANS	Paolo Accorti
8	GOURL	André Fonseca
9	GREAL	Howard Snyder
10	HILAA	Carlos Hernández
11	HUNGC	Yoshi Latimer
12	HUNGO	Patricia McKenna
13	ISLAT	Helen Bennett

□

Problem 2. Give the products details with its supplier company.

Solution.

```
SELECT Products.ProductName,Suppliers.ContactName
FROM Products
JOIN Suppliers
ON Products.SupplierID = Suppliers.SupplierID
WHERE Products.SupplierID IN (
  SELECT DISTINCT SupplierID
  FROM Products
)
```



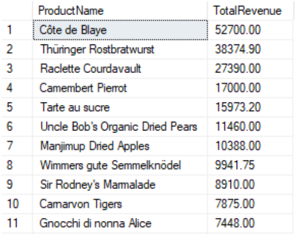
	ProductName	ContactName
1	Chai	Charlotte Cooper
2	Chang	Charlotte Cooper
3	Aniseed Syrup	Charlotte Cooper
4	Chef Anton's Cajun Seasoning	Shelley Burke
5	Chef Anton's Gumbo Mix	Shelley Burke
6	Louisiana Fiery Hot Pepper Sauce	Shelley Burke
7	Louisiana Hot Spiced Okra	Shelley Burke
8	Grandma's Boysenberry Spread	Regina Murphy
9	Uncle Bob's Organic Dried Pears	Regina Murphy
10	Northwoods Cranberry Sauce	Regina Murphy
11	Mishi Kobe Niku	Yoshi Nagase
12	Ikura	Yoshi Nagase
13	Longlife Tofu	Yoshi Nagase

□

Problem 3. Give the name of top products which have highest sale in the year 1998.

Solution.

```
SELECT P.ProductName, TotalRevenue
FROM Products P
INNER JOIN (
  SELECT ProductID, SUM(Quantity*UnitPrice) AS TotalRevenue
  FROM [Order Details] OD
  INNER JOIN Orders O ON OD.OrderID = O.OrderID
  WHERE YEAR(OrderDate) = '1998'
  GROUP BY ProductID
) AS Subquery ON P.ProductID = Subquery.ProductID
ORDER BY TotalRevenue DESC;
```



	ProductName	TotalRevenue
1	Côte de Blaye	52700.00
2	Thüringer Rostbratwurst	38374.90
3	Raclette Courdavault	27390.00
4	Camembert Pierrot	17000.00
5	Tarte au sucre	15973.20
6	Uncle Bob's Organic Dried Pears	11460.00
7	Manjimup Dried Apples	10388.00
8	Wimmers gute Semmelknödel	9941.75
9	Sir Rodney's Marmalade	8910.00
10	Carnarvon Tigers	7875.00
11	Gnocchi di nonna Alice	7448.00

□

Problem 4. Give the name of employees with its manager name.

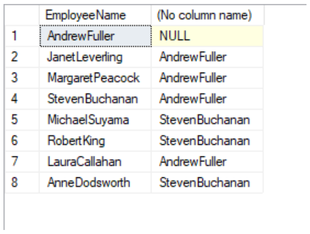
Solution.

```
SELECT CONCAT(E.FIRSTNAME,E.LASTNAME) EmployeeName,
(SELECT CONCAT(E1.FirstName, E1.LastName) ManagerName
FROM Employees E1
WHERE E.ReportsTo=E1.EmployeeID)
FROM Employees E
```

□

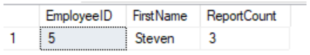
Problem 5. Give the full names of managers who have less than two employees.

Solution.



	EmployeeName	(No column name)
1	AndrewFuller	NULL
2	JanetLeverling	AndrewFuller
3	MargaretPeacock	AndrewFuller
4	StevenBuchanan	AndrewFuller
5	MichaelSuyama	StevenBuchanan
6	RobertKing	StevenBuchanan
7	LauraCallahan	AndrewFuller
8	AnneDodsworth	StevenBuchanan

```
SELECT EmployeeID, FirstName, ReportCount
FROM (
  SELECT E2.EmployeeID, E2.FirstName, COUNT(E1.EmployeeID) AS ReportCount
  FROM Employees E1
  JOIN Employees E2 ON E1.ReportsTo = E2.EmployeeID
  GROUP BY E2.EmployeeID, E2.FirstName
) AS Subquery
WHERE ReportCount < 4;
```



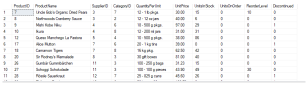
	EmployeeID	FirstName	ReportCount
1	5	Steven	3

□

Problem 6. List all the products whose price is more than average price.

Solution.

```
SELECT *
FROM Products P
WHERE P.UnitPrice > (
  SELECT AVG(UnitPrice)
  FROM Products
);
```



	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel	Discontinued
1	7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.	30.00	15	0	10	0
2	8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40.00	6	0	0	0
3	9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97.00	29	0	0	1
4	10	Ikura	4	8	12 - 200 ml jars	31.00	31	0	0	0
5	12	Queso Manchego La Pastora	5	4	10 - 500 g pkgs.	38.00	86	0	0	0
6	17	Alice Mutton	7	6	20 - 1 kg tins	39.00	0	0	0	1
7	18	Carnarvon Tigers	7	8	16 kg pkg.	62.50	42	0	0	0
8	20	Sir Rodney's Marmalade	8	3	30 gift boxes	81.00	40	0	0	0
9	26	Gumbär Gummibärchen	11	3	100 - 250 g bags	31.23	15	0	0	0
10	27	Schoggi Schokolade	11	3	100 - 100 g pieces	43.90	49	0	30	0
11	28	Rössle Sauerkraut	12	7	25 - 825 g cans	45.60	26	0	0	1

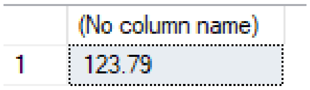
□

Problem 7. Find second highest priced product without using TOP statement

Solution.

```
SELECT MAX(UnitPrice)
FROM (
  SELECT *
  FROM Products
  WHERE UnitPrice NOT IN (
    SELECT MAX(UnitPrice)
    FROM Products
  )
) AS Subquery;
```

□

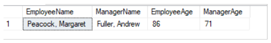


	(No column name)
1	123.79

Problem 8. Are there any employees who are elder than their managers? List that names of those employees.

Solution.

```
SELECT
    e1.LastName + ', ' + e1.FirstName AS EmployeeName,
    e2.LastName + ', ' + e2.FirstName AS ManagerName,
    YEAR(GETDATE()) - YEAR(e1.BirthDate) AS EmployeeAge,
    YEAR(GETDATE()) - YEAR(e2.BirthDate) AS ManagerAge
FROM
    Employees e1
    INNER JOIN Employees e2 ON e1.ReportsTo = e2.EmployeeID
WHERE
    YEAR(GETDATE()) - YEAR(e1.BirthDate) > YEAR(GETDATE()) - YEAR(e2.BirthDate)
```



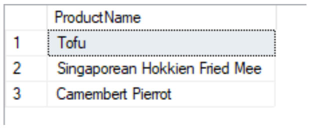
	EmployeeName	ManagerName	EmployeeAge	ManagerAge
1	Peacock, Margaret	Fuller, Andrew	86	71

□

Problem 9. List the names of products which were ordered on 8th August 1997.

Solution.

```
SELECT P.ProductName
FROM Products P
WHERE P.ProductID IN (SELECT ProductID FROM [Order Details] WHERE OrderID IN
(SELECT OrderID FROM Orders WHERE OrderDate='19970808'))
```



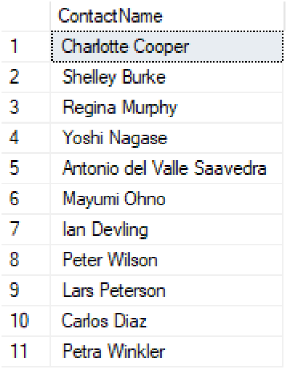
	ProductName
1	Tofu
2	Singaporean Hokkien Fried Mee
3	Camembert Pierrot

□

Problem 10. List the names of suppliers whose supplied products were ordered in 1997.

Solution.

```
SELECT ContactName
FROM Suppliers
WHERE SupplierID IN (
  SELECT SupplierID
  FROM Products
  WHERE ProductID IN (
    SELECT ProductID
    FROM [Order Details]
    WHERE OrderID IN (
      SELECT OrderID
      FROM Orders
      WHERE YEAR(OrderDate) = ’1997’
    )
  )
)
```



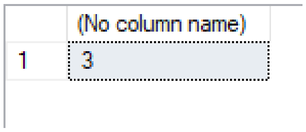
	ContactName
1	Charlotte Cooper
2	Shelley Burke
3	Regina Murphy
4	Yoshi Nagase
5	Antonio del Valle Saavedra
6	Mayumi Ohno
7	Ian Devling
8	Peter Wilson
9	Lars Peterson
10	Carlos Diaz
11	Petra Winkler

□

Problem 11. How many employees are assigned to Eastern region. Give count.

Solution.

```
SELECT COUNT(DISTINCT EmployeeID)
FROM EmployeeTerritories
WHERE TerritoryID IN(
        SELECT TerritoryID
        FROM Territories
        WHERE RegionID IN (
                SELECT RegionID
                FROM Region
                WHERE RegionDescription=’Eastern  ’
        )
)
```



	(No column name)
1	3

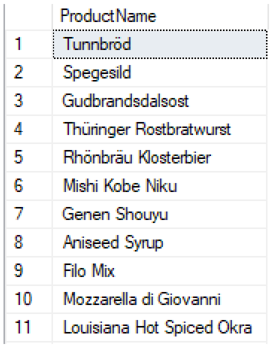
□

Problem 12. Give the name of products which were not ordered in 1996.

Solution.

```
SELECT P.ProductName
FROM Products P
```

```
WHERE P.ProductID IN (
            SELECT ProductID
            FROM [Order Details]
            WHERE OrderID IN
                        (SELECT OrderID
                        FROM Orders
                        WHERE YEAR(OrderDate)<>'1996'
            )
)
```



	ProductName
1	Tunnbröd
2	Spegesild
3	Gudbrandsdalsost
4	Thüringer Rostbratwurst
5	Rhönbräu Klosterbier
6	Mishi Kobe Niku
7	Genen Shouyu
8	Aniseed Syrup
9	Filo Mix
10	Mozzarella di Giovanni
11	Louisiana Hot Spiced Okra

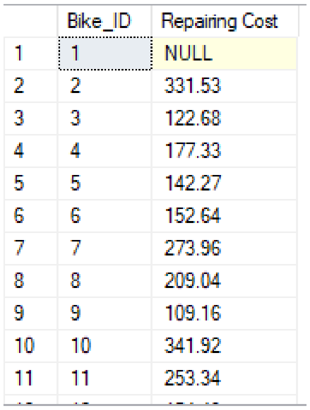
□

2 Dublinbikes

Problem 2. Give the total repairing cost of each bike. Schema should be like this

Solution.

```
SELECT B.Bike_ID,
          (SELECT SUM(Price)
           FROM Repairs R
            JOIN Bike_Status BS ON R.B_Status_ID = BS.B_Status_ID
           WHERE BS.Bike_ID = B.Bike_ID) AS 'Repairing Cost'
FROM Bikes B;
```



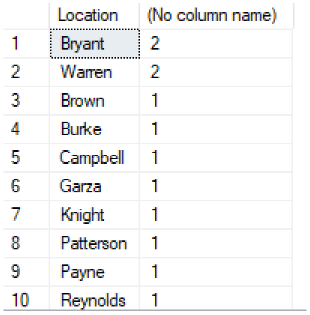
	Bike_ID	Repairing Cost
1	1	NULL
2	2	331.53
3	3	122.68
4	4	177.33
5	5	142.27
6	6	152.64
7	7	273.96
8	8	209.04
9	9	109.16
10	10	341.92
11	11	253.34

□

Problem 3. Bikes of which station needed most repairing

Solution.

```
SELECT S.Location, COUNT(B.Bike_ID)
FROM Stations S
JOIN Bikes B
ON S.Station_ID = B.Station_ID
WHERE B.Bike_ID IN (
    SELECT BS.Bike_ID
    FROM Bike_Status BS
    JOIN Repairs R
    ON BS.B_Status_ID = R.B_Status_ID
)
GROUP BY S.Location
ORDER BY COUNT(B.Bike_ID) DESC;
```



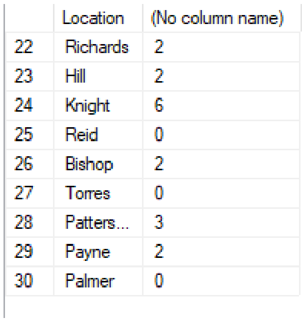
	Location	(No column name)
1	Bryant	2
2	Warren	2
3	Brown	1
4	Burke	1
5	Campbell	1
6	Garza	1
7	Knight	1
8	Patterson	1
9	Payne	1
10	Reynolds	1

□

Problem 4. How many bikes are owned by each station?

Solution.

```
SELECT S.Location, COUNT(B.Bike_ID)
FROM Stations S
LEFT JOIN (
    SELECT Bike_ID, Station_ID
    FROM Bikes
) B
ON S.Station_ID = B.Station_ID
GROUP BY S.Location;
```



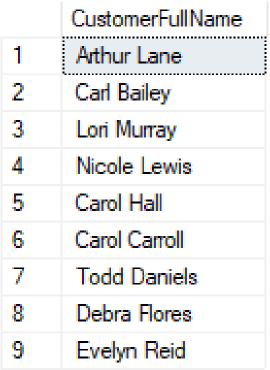
	Location	(No column name)
22	Richards	2
23	Hill	2
24	Knight	6
25	Reid	0
26	Bishop	2
27	Torres	0
28	Patters...	3
29	Payne	2
30	Palmer	0

□

Problem 5. Given the name customers who never rented a bike

Solution.

```
SELECT CONCAT(Fname, ’ ’, Lname) CustomerFullName
FROM Customer_Details
WHERE CONCAT(Fname, ’ ’, Lname) NOT IN (
    SELECT CONCAT(Fname, ’ ’, Lname)
    FROM Customer_Details
    WHERE Customer_ID IN (
        SELECT Customer_ID
        FROM Customers
        WHERE Customer_ID NOT IN (
            SELECT Customer_ID
            FROM Bike_Rentals
        )
    )
);
```



	CustomerFullName
1	Arthur Lane
2	Carl Bailey
3	Lori Murray
4	Nicole Lewis
5	Carol Hall
6	Carol Carroll
7	Todd Daniels
8	Debra Flores
9	Evelyn Reid

□

Problem 7. Identify the customers who always pay using mastercard.

Solution.

```
SELECT CONCAT(Fname, ’ ’, Lname) AS ’name of customers’
FROM (
    SELECT CD.Fname, CD.Lname, C.Customer_ID
    FROM Customer_Details CD
    LEFT JOIN Customers C
    ON CD.Customer_ID = C.Customer_ID
    JOIN Payments P
    ON P.Customer_ID = C.Customer_ID
    JOIN Payment_Method PM
    ON PM.Method_ID = P.Method_ID
    WHERE PM.Method <> ’MASTERCARD’
    GROUP BY CD.Fname, CD.Lname, C.Customer_ID
) AS A
WHERE CONCAT(Fname, ’ ’, Lname) NOT IN (
    SELECT CONCAT(Fname, ’ ’, Lname)
    FROM Customer_Details CD
    LEFT JOIN Customers C
    ON CD.Customer_ID = C.Customer_ID
    JOIN Payments P
    ON P.Customer_ID = C.Customer_ID
    JOIN Payment_Method PM
    ON PM.Method_ID = P.Method_ID
    WHERE PM.Method <> 'MASTERCARD'
    GROUP BY CONCAT(Fname, ' ', Lname)
);
```

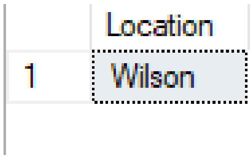
name of customers

□

Problem 8. For which station (Station Name) the most bikes are moved using vans in year 2015.

Solution.

```
SELECT TOP (1) A.Location
FROM (
    SELECT S.Location, SUM(V.Bikes) AS TotalBikes
    FROM Stations S
    JOIN Vans V
    ON S.Station_ID = V.Station_ID
    WHERE YEAR(V.Date_stamp) = '2015'
    GROUP BY S.Location
) AS A
ORDER BY A.TotalBikes DESC;
```



	Location
1	Wilson

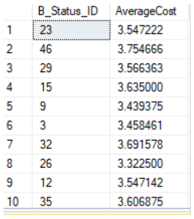
□

Problem 9. Give the average cost of repairing that was spent on each bike

Solution.

```
SELECT B.B_Status_ID, AVG(B.Price) AS AverageCost
FROM (
    SELECT R.B_Status_ID, AVG(R.Price) AS Price
    FROM Repairs R
    GROUP BY R.B_Status_ID
) AS B
JOIN Bike_Status BS
ON BS.B_Status_ID = B.B_Status_ID
GROUP BY B.B_Status_ID;
```

□



	B_Status_ID	AverageCost
1	23	3.547222
2	46	3.754666
3	29	3.566363
4	15	3.635000
5	9	3.439375
6	3	3.458461
7	32	3.691578
8	26	3.322500
9	12	3.547142
10	35	3.606875