INFO90002 Week 7 Lab Answers

Section 1 SQL

Connect to your MySQL database on the engineering server

1.1 Type the query to name the departments on the second floor

	Name
	Clothes
	Recreation

```
SELECT Name
FROM Department
WHERE Floor = 2;
```

1.2 List the names of items delivered by each supplier in the month of August. Arrange the report by supplier name, and within supplier name, list the items in alphabetical order

```
SELECT DISTINCT Supplier.Name, Item.Name
FROM Supplier INNER JOIN Delivery INNER JOIN DeliveryItem
INNER JOIN Item
ON Supplier.SupplierID = Delivery.SupplierID
AND Delivery.DeliveryID = DeliveryItem.DeliveryID
AND DeliveryItem.ItemID = Item.ItemID
WHERE monthname(delivery.deliverydate='August')
ORDER BY Supplier.Name, Item.Name;
```

Name	Name
All Points Inc.	Boots - Mens Hiking
All Points Inc.	Boots - Womens Goretex
All Points Inc.	Boots - Womens Hiking
All Points Inc.	Boots Riding
All Points Inc.	Compass - Silva
All Points Inc.	Cowboy Hat
All Points Inc.	Gortex Rain Coat
All Points Inc.	Horse saddle
All Points Inc.	Pocket knife - Essential
All Points Inc.	Polar Fleece Beanie
All Points Inc.	Sun Hat
All Points Inc.	Torch
Global Books & Maps	Compass - Silva
Global Books & Maps	Exploring in 10 Easy Lessons
Global Books & Maps	Geo positioning system
Global Books & Maps	How to Win Foreign Friends
Global Books & Maps	Map case
Global Books & Maps	Map measure
Global Books & Maps	Pocket knife - Steadfast
Global Books & Maps	Torch

1.3 Type a query to count the number of employees in each department. Be sure to list the department name

Name	EmployeeNumber
Accounting	1
Books	1
Clothes	2
Equipment	2
Furniture	1
Management	1
Marketing	3
Navigation	1
Personnel	1
Purchasing	3
Recreation	1

```
SELECT Department.Name, COUNT(Department.Name) AS
EmployeeNumber
FROM Employee NATURAL JOIN Department
GROUP BY Department.Name
ORDER BY Department.Name, EmployeeNumber;
```

1.4 Whom does Todd manage?

```
SELECT FirstName, LastName
FROM Employee
WHERE BossID IN
    (SELECT EmployeeID
    FROM Employee
    WHERE FirstName = "Todd");
```

	FirstName	LastName
	Nancy	Cartwright

1.5 Find the full name of Sophie's boss

```
SELECT concat(FirstName,' ', LastName) as FullName
FROM Employee
WHERE EmployeeID IN
     (SELECT BossID
     FROM Employee
     WHERE FirstName = 'Sophie');
```

	FullName
	Alice Munro

Self Joins

The query below is a self join to the employee table. You will notice that we have created an alias for the Employee table as emp for employees and boss for their manager. The bossid in the employee table becomes the employeeid in the boss table.

1.6 List the names of each manager and their employees arranged by manager's name and employee's name within manager.

```
SELECT boss.FirstName AS Manager, emp.FirstName AS Employee
  FROM Employee AS emp, Employee AS boss
  WHERE emp.BossID = boss.EmployeeID
  ORDER BY boss.FirstName, emp.FirstName;
```

	Manager	Employee
	Alice	Brier
	Alice	Ned
	Alice	Sophie
	Alice	Todd
	Andrew	James
	Andrew	Mark
	Andrew	Pat
	Andrew	Paul
	Andrew	Saniav
	Brier	Sarah
	Clare	Gigi
	Clare	Maggie
	Clare	Rita
	Ned	Andrew
	Ned	Clare
	Todd	Nancy

1.7 Find the departments that have never sold a geo positioning system

```
SELECT distinct(Department.Name), department.departmentid
FROM Department inner join Sale inner Join SaleItem
ON Department.DepartmentID = Sale.DepartmentID
AND Sale.SaleID = SaleItem.SaleID
WHERE SaleItem.ItemID NOT IN
     (SELECT itemID
      FROM Item
      WHERE name = 'Geo positioning system');
```

	Name	departmentid
	Books	2
	Clothes	3
	Equipment	4
	Furniture	5
	Navigation	6
	Recreation	7

1.8 Type the query to find who earns the lowest salary?

	FirstName	LastName
	Paul	Innit

```
SELECT FirstName, LastName
FROM Employee
WHERE Salary =
     (SELECT MIN(Salary)
      FROM Employee);
```

1.9 Find the department/s that sell at least 4 different items

```
SELECT Department.Name
FROM Department inner join Sale inner join SaleItem
ON Department.DepartmentID = Sale.DepartmentID and
Sale.SaleID = SaleItem.SaleID
Group by Department.Name
Having count(distinct(saleitem.itemid)) >= 4;
```

	Name
	Books
	Clothes
	Equipment
	Navigation
	Recreation

1.10 Find the departments that sell at least 4 items and list how many items each department sells (Alternative wording: Find the departments that have made 4 or more transactions and list how many transactions each department has made.)

```
SELECT Department.Name, count(saleitem.itemid) as
Sale_Quantity
FROM Department inner join Sale inner join SaleItem
ON Department.DepartmentID = Sale.DepartmentID and
Sale.SaleID = SaleItem.SaleID
Group by Department.Name
Having count(distinct(saleitem.itemid)) >= 4;
```

1.11 Type the query to find the Items (ItemID) sold on floors other than the second floor

The result set should look similar to this:

	ItemID
	1
	3
	5
	6
	9
	10
	11
	15
	16

```
SELECT DISTINCT SaleItem.ItemID
FROM Sale natural join SaleItem natural join Department
WHERE Floor <> 2
AND SaleItem.ItemID NOT IN
   (SELECT SaleItem.ItemID
     From Sale natural join SaleItem natural join Department
     Where Department.Floor=2)
ORDER BY ItemID;
```

1.12 Type the query to count the number of direct employees of each manager, List the EmployeeID, Manager Name and number of employees.

Your result set should look similar to this:

	EmployeeID	ENAME	Emp_count
	3	Andrew Jackson	5
	1	Alice Munro	4
	4	Clare Underwood	3
	2	Ned Kelly	2
	5	Todd Beamer	1
	7	Brier Patch	1

```
SELECT boss.EmployeeID, Concat(boss.FirstName,' ',
boss.LastName) as ENAME, COUNT(*) Emp_count
FROM Employee wrk INNER JOIN Employee boss
ON wrk.BossID = boss.EmployeeID
GROUP BY boss.EmployeeID, ENAME
ORDER BY EMP_COUNT DESC
```

1.13 List the department id and average salary where the average salary of the employees of each manager is more than $55000

```
SELECT wrk.DepartmentID, FORMAT(AVG(wrk.Salary),2) AS
AvgSalary
FROM Employee wrk
WHERE wrk.EmployeeID NOT IN
      (SELECT Department.ManagerID
       FROM Department
       WHERE wrk.EmployeeID = Department.ManagerID
       AND wrk.DepartmentID = Department.DepartmentID)
GROUP BY wrk.DepartmentID
HAVING AVG(wrk.Salary) > 55000;
```

	DepartmentID	AvgSalary
	9	86.000.00

1.14 Find the items (itemID) sold by ALL departments located on the second floor

```
SELECT SaleItem.ItemID
FROM SaleItem NATURAL JOIN Sale NATURAL JOIN Department
WHERE Department.Floor = 2
GROUP BY SaleItem.ItemID
HAVING count(DISTINCT Department.DepartmentID) =
             (SELECT count(DISTINCT DepartmentID)
```

```
                FROM Department
                WHERE Department.Floor = 2
                )
ORDER BY SaleItem.ItemID;
```

And using a different method

```
SELECT DISTINCT ItemID
FROM Item
WHERE NOT EXISTS
     (SELECT *
      FROM Department
      WHERE Department.Floor = 2
      AND NOT EXISTS
          (SELECT *
           FROM SaleItem natural join Sale
           WHERE SaleItem.ItemID = Item.ItemID
           AND Sale.DepartmentID = Department.DepartmentID
           )
       )
ORDER BY ItemID;
```

	ItemID
	14
	NULL

1.15 Find the supplier id and supplier names that do not deliver compasses

```
SELECT SupplierID, Supplier.Name
FROM Supplier
WHERE SupplierID NOT IN
      (SELECT SupplierID
       FROM Delivery NATURAL JOIN DeliveryItem NATURAL JOIN
ITEM
       WHERE Item.Name Like 'Compass%');
```

	SupplierID	Name
	104	Sweatshops Unlimited
	106	Sao Paulo Manufacturing
	NULL	NULL

1.16 Find, for each department that has sold items of type E. List the department name and the average salary of the employees

```
SELECT Department.Name, FORMAT(AVG(Employee.Salary),2) AS
AverageSalary
FROM Employee INNER JOIN Department INNER JOIN Sale
INNER JOIN SaleItem INNER JOIN Item
ON Employee.DepartmentID = Department.DepartmentID
AND Department.DepartmentID = Sale.DepartmentID
AND Sale.SaleID = SaleItem.SaleID
AND SaleItem.ItemID = Item.ItemID
WHERE Item.Type = 'E'
GROUP BY Department.Name;
```

	Name	AverageSalary
	Books	45.000.00
	Clothes	46.000.00
	Equipment	43.000.00
	Furniture	45.000.00
	Navigation	45.000.00
	Recreation	45.000.00

1.17 Find the total number of items (list the item and sale quantity) of type E sold by the departments on the second floor

```
SELECT ITEM.Name, SUM(SaleItem.Quantity) AS QUANTITY
FROM Item INNER JOIN SaleItem INNER JOIN Sale INNER JOIN
Department
ON Item.ItemID = SaleItem.ItemID
AND Sale.SaleID = SaleItem.SaleID
AND Department.DepartmentID = Sale.DepartmentID
WHERE Item.Type = 'E'
AND Department.Floor = 2
GROUP BY ITEM.ITEMID;
```

	Name	QUANTITY
	Pocket knife - Essential	9
	Torch	8

1.18 Write the query to find the total quantity sold of each item by the departments on the second floor

The result set should look similar to this:

	Name	TOTAL_SALES
	Sun Hat	10
	Pocket knife - Essential	9
	Torch	8
	Polar Fleece Beanie	6
	Tent - 2 person	5
	Boots - Womens Goretex	4
	Tent - 8 person	2
	Gortex Rain Coat	2
	Boots - Mens Hiking	2
	Boots - Womens Hiking	1
	Tent - 4 person	1
	Cowboy Hat	1

```
SELECT Item.Name, SUM(SaleItem.Quantity) as TOTAL_SALES
FROM Item INNER JOIN SaleItem INNER JOIN Sale INNER JOIN
Department
on Item.ItemiD = SaleItem.ItemID
AND SaleItem.SaleID = Sale.SaleID
AND Sale.DepartmentID = Department.DepartmentID
WHERE Department.Floor = 2
GROUP BY Item.Name
ORDER BY TOTAL_SALES DESC
```

1.19 List each item (ItemName) delivered to at least two departments by each supplier that delivers it

```
SELECT DISTINCT ItemName
FROM Delivery NATURAL JOIN Item
WHERE ItemID NOT IN
(SELECT ItemID FROM Delivery
GROUP BY ItemID, SupplierID
HAVING COUNT(DISTINCT DepartmentID) < 2);
```

ItemName
Boots - snakeproof
How to Win Foreign Friends
Exploring in 10 Easy Lessons

1.20 What is the average delivery quantity of items of type N delivered by each supplier to each department (given that the supplier delivers items of type N to the department)?

```
SELECT Delivery.SupplierID, SupplierName, DepartmentID,
Item.ItemName,
FORMAT(AVG(DeliveryQTY),2) AS DelQTY
FROM Delivery INNER JOIN Supplier INNER JOIN Item
ON Delivery.SupplierID =  Supplier.SupplierID
AND Delivery.ItemID = Item.ItemID
WHERE  ItemType = 'N'
GROUP BY Delivery.SupplierID,  SupplierName,
DepartmentID,  Item.ItemName;
```

SupplierID	SupplierName	DepartmentID	ItemName	DelQTY
101	Global Books & Maps	6	Compass	50.00
101	Global Books & Maps	6	Geo positioning system	10.00
101	Global Books & Maps	6	Map measure	10.00
101	Global Books & Maps	6	Sextant	2.00
102	Nepalese Corp.	2	Compass	5.00
102	Nepalese Corp.	6	Compass	1.00
102	Nepalese Corp.	6	Geo positioning system	1.00
102	Nepalese Corp.	6	Map measure	10.00
102	Nepalese Corp.	7	Sextant	5.00
103	All Sports Manufacturing	2	Geo positioning system	1.00
103	All Sports Manufacturing	2	Sextant	1.00
103	All Sports Manufacturing	4	Compass	20.00
103	All Sports Manufacturing	6	Map measure	15.00
104	Sweatshops Unlimited	6	Sextant	3.00
105	All Points_ Inc.	4	Compass	5.00
105	All Points_ Inc.	4	Sextant	1.00
106	Sao Paulo Manufacturing	4	Sextant	1.00

Relational Divides

1.21 List the department names that have not recorded a sale for *all* the items of type N

```
SELECT department.Name
FROM department
WHERE DepartmentID NOT IN
      (SELECT DepartmentID
       FROM department
       WHERE NOT EXISTS
            (SELECT *
             FROM item
```

```
            WHERE item.Type = 'N'
            AND NOT EXISTS
                  (SELECT *
                   FROM sale natural join saleitem
                   WHERE sale.DepartmentID =
department.DepartmentID
                   AND saleitem.ItemID = item.ItemID)
            )
      )
ORDER BY department.Name;
```

	Name
	Accounting
	Books
	Clothes
	Equipment
	Furniture
	Management
	Marketing
	Personnel
	Purchasing
	Recreation

1.22 Who are the supplier id and supplier names that deliver all the items of type N?

NB: A simpler Relational Divide problem but restricted to items of type N

```
SELECT SupplierID, supplier.Name
FROM supplier
WHERE NOT EXISTS
      (SELECT *
       FROM item
       WHERE item.Type = 'N'
       AND NOT EXISTS
            (SELECT * FROM delivery natural join deliveryitem
             WHERE delivery.SupplierID = supplier.SupplierID
             AND deliveryitem.ItemID = item.ItemID)
       );
```

	SupplierID	Name
	101	Global Books & Maps
	102	Nepalese Corp.
	103	All Sports Manufacturing
	NULL	NULL

1.23 List the departments that have at least one sale of all the items delivered to them

Attempt 1:

```
SELECT DISTINCT DepartmentID
FROM deliveryitem del1
WHERE NOT EXISTS
      (SELECT *
       FROM deliveryitem del2
       WHERE del2.DepartmentID = del1.DepartmentID
AND NOT EXISTS
      (SELECT *
        FROM saleitem natural join sale
        WHERE del2.ItemID = saleitem.ItemID
              AND del1.DepartmentID = sale.DepartmentID));
```

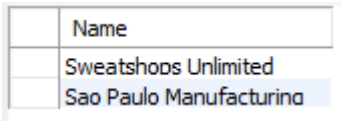
	DepartmentID
	2
	5

Attempt 2:

```
SELECT DISTINCT DepartmentID
FROM deliveryitem del1
WHERE NOT EXISTS
      (SELECT *
       FROM deliveryitem del2
       WHERE del2.DepartmentID = del1.DepartmentID
            AND ItemID NOT IN
            (SELECT ItemID
             FROM sale natural join saleitem
             WHERE sale.DepartmentID = del1.DepartmentID)
      );
```

	DepartmentID
	2
	5

1.24 Type a relational divide query that lists the suppliers that delivery only items sold by the Books department



	Name
	Sweatshops Unlimited
	Sao Paulo Manufacturing

```
SELECT supplier.Name
FROM supplier
WHERE SupplierID IN
    (SELECT SupplierID
    FROM delivery)
    AND NOT EXISTS
        (SELECT *
        FROM deliveryitem natural join delivery
        WHERE delivery.SupplierID = supplier.SupplierID
        AND ItemID NOT IN
            (SELECT ItemID
             FROM saleitem NATURAL JOIN sale NATURAL JOIN
department
             WHERE department.Name = 'Books'));
```

As you will see there are many different queries that can achieve the same result set.

End of Week 7 Lab

Appendix: The New Department Store ER Physical Model