Tutorial - Week 7 Solutions

Objectives

- Learn SQL By Example
- Discover MySQL functions (HOMEWORK)

Connect to your MySQL database on the engineering server

1) TASK Type the query to list the id and name of all green items of type C

ItemID	Name
12	Gortex Rain Coat

```
SELECT ItemID, Name
FROM ITEM
WHERE Type = 'C'
And Colour = 'Green';
```

2) TASK Type the query to find the items delivered by at least two suppliers

Name
Compass - Silva
Exploring in 10 Easy Lessons
Geo positioning system
Gortex Rain Coat
How to Win Foreign Friends
Map case
Map measure
Pocket knife - Essential
Pocket knife - Steadfast
Torch

```
SELECT item.Name
FROM item
NATURAL JOIN deliveryitem
NATURAL JOIN delivery
GROUP BY item.Name
HAVING COUNT(DISTINCT(SupplierID)) >= 2;
```

3) TASK Find the name of the highest-paid employee in the Marketing department

Firstname	Lastname	Salary
Ned	Kelly	85000.00

```
SELECT employee.Firstname, employee.Lastname, employee.Salary
FROM employee
WHERE employee.salary IN
       (SELECT max(salary)
        FROM employee
        NATURAL JOIN department
    WHERE department.Name = 'Marketing')
```

4) TASK 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

5) TASK 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

6) TASK 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

7) TASK Type 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
```

8) TASK Find the items that are not sold by departments on the second floor but are sold on other floors within the store

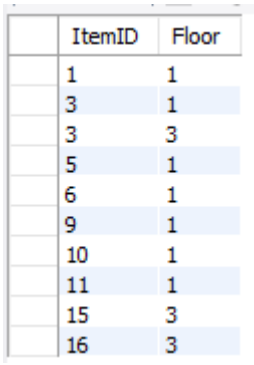
When solving problems like this, work in steps

1. Identify the items sold on the second floor,

2. Then find the items that are not in the result from part 1

```
SELECT DISTINCT saleitem.ItemID, department.floor
FROM sale
INNER JOIN saleitem
INNER JOIN department
ON sale.SaleID = saleitem.SaleID
AND sale.DepartmentID = department.DepartmentID
WHERE ItemID NOT IN
        (SELECT DISTINCT ItemID
        FROM sale
        INNER JOIN saleitem
        INNER JOIN department
        ON sale.SaleID = saleitem.SaleID AND sale.DepartmentID =
        department.DepartmentID
        WHERE department.Floor =2)
ORDER BY ItemID, department.floor;
```

The inner query identifies the itemid's that ARE sold on the second floor. The outer query then finds all itemids which have been sold but are not in the inner query but only for departments not located on the second floor.

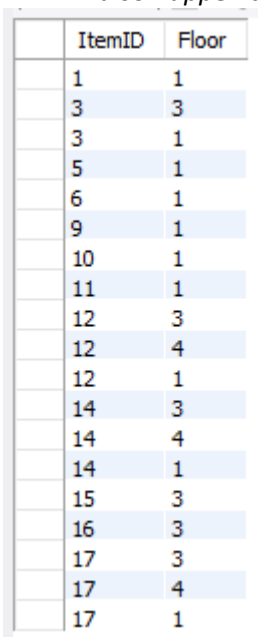


	ItemID	Floor
	1	1
	3	1
	3	3
	5	1
	6	1
	9	1
	10	1
	11	1
	15	3
	16	3

Compare that to this query:

```
SELECT distinct ItemID, department.Floor
FROM sale
INNER JOIN saleitem
INNER JOIN department
ON sale.SaleID = saleitem.SaleID
AND sale.DepartmentID = department.DepartmentID
WHERE department.Floor!=2
ORDER BY ITEMID;
```

This query only finds items sold on floors other than the second floor – but this includes items which also happen to have been sold on the second floor.



	ItemID	Floor
	1	1
	3	3
	3	1
	5	1
	6	1
	9	1
	10	1
	11	1
	12	3
	12	4
	12	1
	14	3
	14	4
	14	1
	15	3
	16	3
	17	3
	17	4
	17	1

The additional itemid's are 12 14 and 17. A final query will confirm which floors items 12, 14 and 17 are sold on

```
SELECT distinct(item.ItemID), department.Floor
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 saleitem.ItemID in (12,14,17)
ORDER BY item.itemid, department.floor;
```

ItemID	Floor
12	1
12	2
12	3
12	4
14	1
14	2
14	3
14	4
17	1
17	2
17	3
17	4

Be sure you understand the question being asked.

9) TASK Find the numbers and names of the employees who earn more than their manager.

```
SELECT emp.EmployeeID, emp.FirstName, emp.LastName , emp.Salary as empSal,
boss.Salary as BossSal
FROM employee emp
INNER JOIN employee boss
ON emp.BossID = boss.EmployeeID
WHERE boss.Salary < emp.Salary;
```

Note: This is a unary join – we have created aliases for the Employee table. The first alias is 'emp' the second 'boss'

EmployeeID	FirstName	LastName	empSal	BossSal
8	Sarah	Fergusson	86000.00	73000.00

10) TASK Find, for each department on the second floor, the average salary of the employees

```
SELECT department.Name, FORMAT(AVG(employee.Salary),2) as AverageSalary
FROM employee
NATURAL JOIN department
WHERE department.Floor = 2
GROUP BY department.Name;
```

Name	AverageSalary
Clothes	46.000.00
Recreation	45.000.00

11) TASK List suppliers that deliver a total quantity of items of types C and N that is greater than 40

In this SQL query we are building the query in stages

i) First let's find the items that are of type C and N

```
SELECT item.Name, item.Type
FROM item
WHERE item.Type in ('C','N')
order by item.Name;
```

ii) Then find out how many of those items have been delivered

```
SELECT item.Name, SUM(deliveryitem.Quantity)
FROM deliveryitem
INNER JOIN item
ON deliveryitem.ItemID = item.ItemID
WHERE item.Type in ('C','N')
GROUP BY item.Name;
```

iii) And if the quantity delivered is greater than 40

```
SELECT item.Name, SUM(deliveryitem.Quantity)
FROM deliveryitem
INNER JOIN item
ON deliveryitem.ItemID = item.ItemID
WHERE item.Type in ('C','N')
GROUP BY item.Name
HAVING Sum(deliveryitem.Quantity) > 40;
```

iv) Now let's find the Supplier Names and IDs:
Placed below are three different approaches to solving this task

```
SELECT delivery.SupplierID, supplier.Name , SUM(deliveryitem.Quantity)
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 item.Type IN ('C','N')
GROUP BY delivery.SupplierID, supplier.Name
HAVING SUM(deliveryitem.Quantity) > 40;
```

Notice the difference in the WHERE statement using an OR:

```
SELECT delivery.SupplierID, supplier.Name , SUM(deliveryitem.Quantity)
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 (item.Type = 'C' OR item.Type = 'N')
GROUP BY delivery.SupplierID, supplier.Name
HAVING SUM(deliveryitem.Quantity) > 40;
```

And the WHERE x OR y condition can be written without parenthesis

```
SELECT delivery.SupplierID, supplier.Name , SUM(deliveryitem.Quantity)
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 item.Type = 'C'
    OR item.Type = 'N'
    GROUP BY delivery.SupplierID, supplier.Name
    HAVING SUM(deliveryitem.Quantity) > 40;
```

The result is the same:

SupplierID	Name
101	Global Books & Maps
105	All Points Inc.

12) TASK What is the average delivery quantity of items of type N made by each company who delivers them. Be sure to list the Supplier ID and name, Item type and name and average delivery quantity in your answer.

```
SELECT delivery.SupplierID, supplier.Name AS supplier,
item.Type, item.Name as Item, FORMAT(AVG(deliveryitem.Quantity),2) AS AvgDelQty
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 item.Type = 'N'
GROUP BY delivery.SupplierID,supplier.Name,item.Name;
```

SupplierID	supplier	Type	Item	AvgDelQty
101	Global Books & Maps	N	Compass - Silva	4.67
102	Nepalese Corp.	N	Compass - Silva	3.00
103	All Sports Manufacturing	N	Compass - Silva	8.00
105	All Points_ Inc.	N	Compass - Silva	1.00
101	Global Books & Maps	N	Geo positioning sy…	3.00
102	Nepalese Corp.	N	Geo positioning sy…	4.00
103	All Sports Manufacturing	N	Geo positioning sy…	1.50
101	Global Books & Maps	N	Map measure	10.00
102	Nepalese Corp.	N	Map measure	10.00
103	All Sports Manufacturing	N	Map measure	10.00

13) TASK List the name and salary of the managers with more than 2 employees

```
SELECT employee.FirstName, employee.LastName, employee.Salary
FROM employee
WHERE employeeID IN
     (SELECT BossID
      FROM employee
      GROUP BY BossID
      HAVING COUNT(*) > 2);
```

FirstName	LastName	Salary
Alice	Munro	125000.00
Andrew	Jackson	55000.00
Clare	Underwood	52000.00

14) TASK List item names that are delivered by Nepalese Corp and sold in the Navigation department

```
SELECT DISTINCT item.Name
FROM item
WHERE ItemID IN
  (SELECT ItemID
   FROM deliveryitem
   NATURAL JOIN delivery
   NATURAL JOIN supplier
   WHERE supplier.Name = 'Nepalese corp.')
AND ItemID IN
  (SELECT ItemID
   FROM saleitem
   NATURAL JOIN sale
   NATURAL JOIN department
   WHERE department.Name = 'Navigation');
```

	Name
	Geo positioning system
	Torch
	Gortex Rain Coat
	Pocket knife - Essential
	Compass - Silva
	Map case
	Map measure
	How to Win Foreign Friends

15) TASK Type the query that finds the name and salary of Clare Underwood's manager

```
SELECT employee.FirstName, employee.LastName, employee.Salary
FROM employee
WHERE employeeID IN
  (SELECT BossID
   FROM employee
   WHERE employee.FirstName = 'Clare'
   AND employee.LastName = 'Underwood');
```

	FirstName	LastName	Salary
	Ned	Kelly	85000.00

16) TASK List the ids of the departments where all of the employees earn less than their manager

```
SELECT DISTINCT DepartmentID
FROM employee
WHERE DepartmentID NOT IN
  (SELECT wrk.DepartmentID
   FROM employee wrk
   INNER JOIN employee boss
   ON wrk.BossID = boss.employeeID
   WHERE wrk.Salary >= boss.Salary)
AND employee.bossid IS NOT NULL
ORDER BY DepartmentID;
```

Hint: Notice that the inner query uses a unary join to create a result set that lists all departmentids where an employee earns more than their boss. That is why the condition is NOT IN

DepartmentID
2
3
4
5
6
7
8
10
11

17) TASK Find the supplier id and supplier names that deliver both compasses and an item other than compasses

Attempt 1:

```
SELECT DISTINCT delivery.SupplierID, supplier.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 item.Name NOT LIKE 'Compass%'
AND delivery.SupplierID IN
       (SELECT SupplierID
        FROM delivery
        NATURAL JOIN item
        NATURAL JOIN deliveryitem
        WHERE item.Name LIKE 'Compass%')
ORDER BY delivery.SupplierID;
```

Attempt 2:

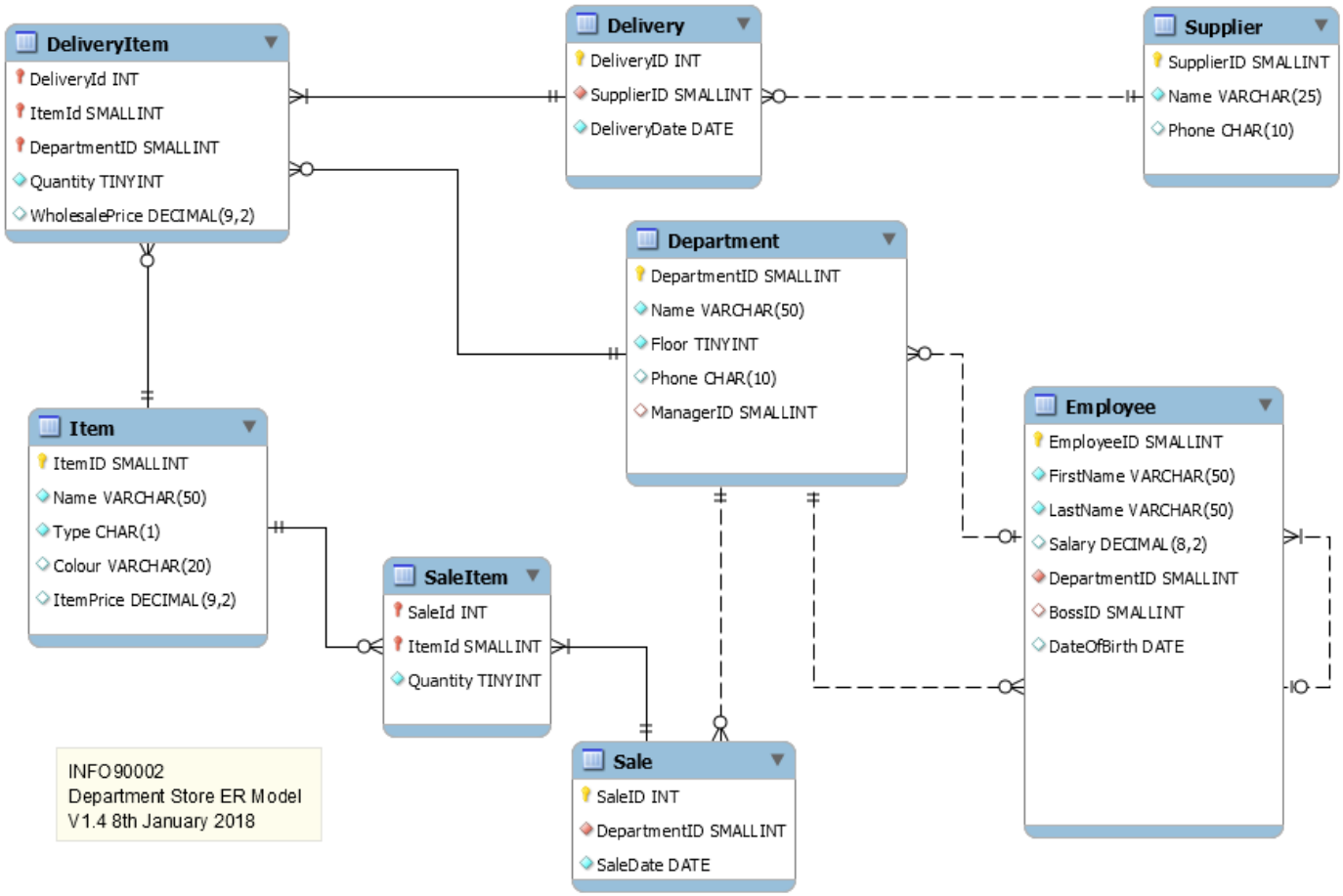
```
SELECT DISTINCT delivery.SupplierID, supplier.Name
FROM supplier
INNER JOIN delivery
INNER JOIN deliveryitem
INNER JOIN item
ON supplier.SupplierID = delivery.SupplierID
AND delivery.DeliveryID = deliveryitem.DeliveryID
AND deliveryitem.ItemID = item.ItemID
WHERE delivery.SupplierID IN
       (SELECT SupplierID
        FROM delivery
        NATURAL JOIN deliveryitem
        NATURAL JOIN item
        WHERE item.Name LIKE'Compass%')
GROUP BY delivery.SupplierID, supplier.Name
HAVING COUNT(DISTINCT item.Name) > 1
order by delivery.SupplierID;
```

	SupplierID	Name
	101	Global Books & Maps
	102	Nepalese Corp.
	103	All Sports Manufacturing
	105	All Points Inc.

Note: Attempt 1 uses the approach to find those suppliers that supply things other than compasses and also supply compasses (sub query).

Attempt 2 uses a more generalizable approach. The generalizable approach is better as it allows queries such as "Find suppliers that deliver two items other than compasses" – change the >1 to >2 in the HAVING clause in Attempt 2 to do this. (Attempt 2 uses DISTINCT to handle multiple deliveries of compasses for the same supplier.)

Appendix New Department Store Physical ER Model



SQL Homework - Functions

Most of this week's homework requires you to read the manual. That is the functions section of the MySQL reference manual https://dev.mysql.com/doc/refman/8.0/en/functions.html

1) H1 How many deliveries have there been in the month of July?
   Hint: the only information you have been given is the month name

```
SELECT COUNT(DeliveryID)
FROM Delivery
WHERE Monthname(deliverydate) = 'July';
```

2) H2 List the names of the tents available for sale

```
SELECT name
FROM Item
WHERE name like '%Tent%';
```

3) H3 What month has had the highest number of sales?

```
SELECT Count(SaleID), Monthname(SaleDate)
FROM sale
Group By Monthname(SaleDate)
Order by Count(SaleID) DESC
LIMIT 1;
```

4) H4 List the salary total and employee count for each departmentid. Order by the smallest salary total to largest.

```
SELECT DepartmentID, COUNT(employeeid), SUM(Salary)
FROM Employee
GROUP BY DepartmentID
ORDER BY Sum(Salary);
```

5) H5 How many sales have been on a Sunday?

```
SELECT count(saleID)
FROM sale
Where DayName(Saledate) = 'Sunday';
```

6) H6 How many days have elapsed between the first delivery date and most recent delivery date for each supplier?

```
SELECT SupplierID, Datediff(max(deliverydate),min(deliverydate)),
count(distinct(deliverydate))
FROM Delivery
Group by SupplierID;
```

7) H7 Produce the following output by writing a SQL statement

Where is each department?
The Management department is on floor number 5
The Books department is on floor number 1
The Clothes department is on floor number 2
The Equipment department is on floor number 3
The Furniture department is on floor number 4
The Navigation department is on floor number 1
The Recreation department is on floor number 2
The Accounting department is on floor number 5
The Purchasing department is on floor number 5
The Personnel department is on floor number 5
The Marketing department is on floor number 5

```
SELECT CONCAT('The ',name,' department is on floor number ', floor) AS
'Where is each department?'
FROM department;
```

8) H8 Find the minimum, maximum, average and standard deviation for salaries in each department

```
SELECT DepartmentID, MIN(Salary), Max(Salary), STDDEV(Salary)
FROM Employee
GROUP BY DepartmentID;
```