Tutorial - Week 7

Objectives

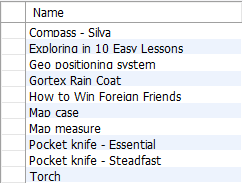
* Learn SQL By Example
* Discover MySQL functions (HOMEWORK)

Connect to your MySQL database on the engineering server

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



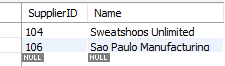
1. TASK Type the query to find the items delivered by at least two suppliers



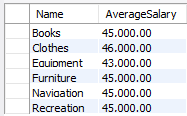
# TASK Find the name of the highest-paid employee in the Marketing department



1. TASK Find the supplier id and supplier names that do not deliver compasses



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



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

A screenshot of a cell phone  Description automatically generated

# 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:



# 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

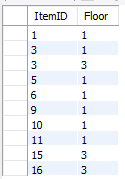
FROM sale

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.*



Compare that to this query:

**SELECT** distinct ItemID, department.Floor

**FROM** sale

**INNER JOIN** saleitem

**INNER JOIN** department

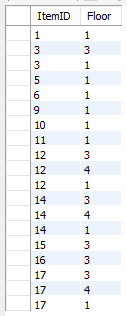
**ON** sale.SaleID = saleitem.SaleID

**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.*



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

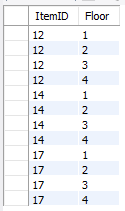
**ON** item.ItemID = saleitem.ItemID

**AND** saleitem.SaleID = sale.SaleID

**AND** sale.DepartmentID = department.DepartmentID

**WHERE** saleitem.ItemID in (12,14,17)

**ORDER BY** item.itemid, department.floor;

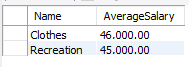


Be sure you understand the question being asked.

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



1. TASK Find, for each department on the second floor, the average salary of the employees



# 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

1. 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;

1. 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;

1. 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;

1. 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

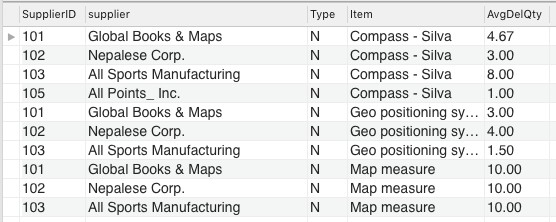
## WHERE item.Type = 'C' OR item.Type = 'N'

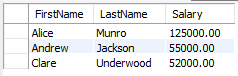
GROUP BY delivery.SupplierID, supplier.Name HAVING SUM(deliveryitem.Quantity) > 40;

The result is the same:

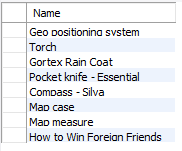


# TASK What is the average delivery quantity of items of type N made by each company who delivers them. Be sure to list the Supplier ID and name, Item type and name and average delivery quantity in your answer.



1. TASK List the name and salary of the managers with more than 2 employees

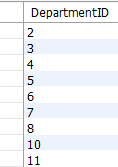
# TASK List item names that are delivered by Nepalese Corp and sold in the Navigation department



1. TASK Type the query that finds the name and salary of Clare Underwood's manager



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



1. 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

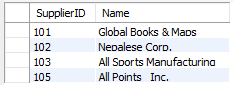
WHERE delivery.SupplierID IN (SELECT SupplierID

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;

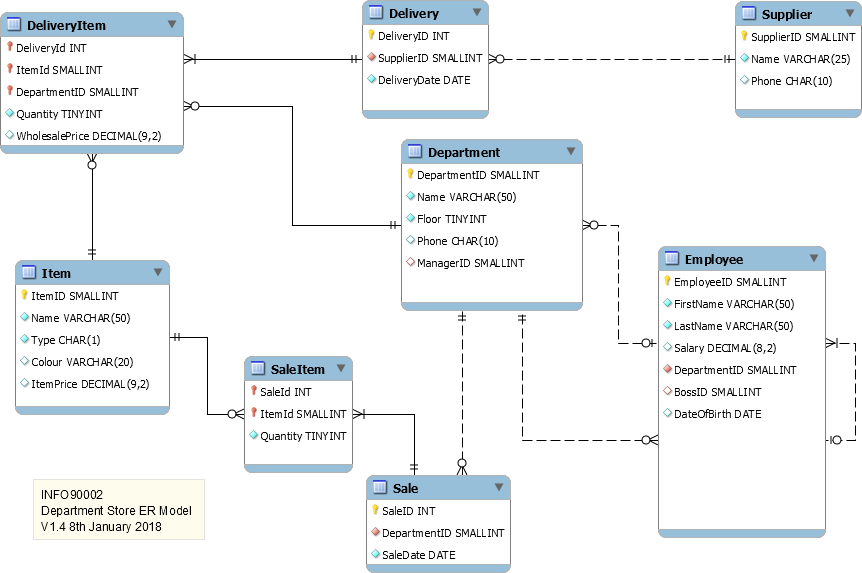


*Note: Attempt 1 uses the approach to find those suppliers that supply things other than compasses and also supply compasses (sub query).*

*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.)*

**End of Tutorial 7**

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>

# H1 How many deliveries have there been in the month of July?

*Hint: the only information you have been given is the month name*

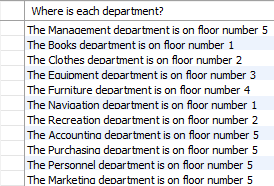
# H2 List the names of the tents available for sale

1. H3 What month has had the highest number of sales?

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

1. H5 How many sales have been on a Sunday?

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

1. H7 Produce the following output by writing a SQL statement

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