SQL Exercises – DML

1. Consider the database previously designed with the name ***Employees*** (Exercise 1 in the modelling databases exercises). Add a new column to the table ***Departments*** named ***AnnualBudget***. Modify the table ***Employees*** and add the column ***LastName***. After these modifications, insert some data into the tables. The following department names can be used: ***IT, Research, Human Resources, Accounting, Finance, QA, Intendance, Computing, Legal, School Control, Commerce, Law***. Use the data provided for the database ***Company*** to fill in the table ***Employees***. Once the database is completed, create and run the following ***SQL*** queries:
    a. Get the last names of the employees.
    b. Obtain the last names of the employees without repetitions (duplicates).
    c. Obtain all the data of the employees that are called with certain last name ***Peacock***.
    d. Obtain all the data of the employees that are called with last name ***Fuller*** and those that are called with the last name ***King***.
    e. Obtain all the data of the employees that work for department ***2***.
    f. Obtain all the data of the employees that work for department ***7*** and for department ***11***.
    g. Obtain all the data of the employees whose last name begins with ***P***.
    h. Get the total budget of all departments.
    i. Obtain the number of employees in each department.
    j. Obtain a complete list of employees, including the name and last name of the employee along with the name and budget of their department.
    k. Obtain the names and last names of the employees who work in departments whose budget is greater than 160000.
    l. Obtain data from departments whose budget is higher than the average budget of all departments.
    m. Get the names (only the names) of the departments that have more than two employees.
    n. Add a new department named ***Unimportant Matters***, with a budget of 140000.
    o. Add an employee linked to the newly created department. The name of the employee is: ***John Smith***.
    p. Apply a budget cut of 10% to all departments.
    q. Reassign the employees of the ***Research*** department to the ***IT*** department.

2. Consider the database designed to manage products, customers and vendors. Insert some data into the tables (10 records minimum). After inserting the records, create the following queries:
    a. A list of all products.
    b. All data about the customers.
    c. Average price of all products.
    d. Minimum and maximum price of all products.
    e. Vendor that supplies the product with ***ID = 5***.
    f. Products that provides the vendor with ***ID = 2***.
    g. Vendor that supplies the most expensive product.
    h. Increase the price of all products in ***5%***.

i. The first three customers where the birth date is greater than ***09/01/1980***.
j. The total number of products.
k. A list of all customer whose name starts with the letter ***a***.
l. All products with price between $10 and $20.
m. A list of all vendors with the products they provide.

3. Design a database to manage the articles sold in a computer products store, which are supplied by different manufacturers. It is known that a manufacturer can supply several articles, while an article can only be supplied by one manufacturer. When registering manufacturers information, it's desired to store the code and its name, while the articles' information will be the code, name and price. After inserting some data into the database, execute the following queries:
   a. Get the names of the products in the store.
   b. Get the names and prices of the store's articles.
   c. Get the name of the articles whose price is less than or equal to ***$20***.
   d. Obtain all the data of the articles whose price is between ***$16*** and ***$32*** (both included).
   e. Get the name and price in euros of articles.
   f. Select the average price of all articles.
   g. Obtain the average price of articles whose manufacturer code is ***2***.
   h. Get the number of articles whose price is greater than or equal to ***$18***.
   i. Get the name and price of the items whose price is greater than or equal to ***$18*** and order them down by price, and then ascending by name.
   j. Obtain a complete list of articles, including for each article the data of the article and its manufacturer.
   k. Obtain the average price of the products of each manufacturer, showing the name of the manufacturer.
   l. Obtain the names of the manufacturers that offer articles whose average price is greater than or equal to ***$30***.
   m. Get the name and price of the cheapest article.
   n. Obtain a list with the name and price of the most expensive articles from each provider (including the name of the provider).
   o. Change the name of the product ***6*** to ***Laser Printer***.
   p. Apply a ***10%*** discount to all articles.

4. Design a database that allows to keep updated the information related to the ***boxes*** that are stored in ***warehouses***. It is known that in a ***warehouse*** several boxes can be stored, while the same ***box*** can only be stored in a warehouse. From warehouses is desired to store ***code***, ***geographic location*** and ***capacity*** in cubic meters. From the boxes is desired to store ***code***, ***content*** and ***price***. After filling in the database with at least 10 items, create the following queries against the database:
   a. Obtain a list of all data about warehouses.
   b. Obtain a list of all the boxes whose content has a price greater than $350.
   c. Obtain a list of content types of the boxes (Do not consider repeated contents).

d. Get the average price of all boxes.
e. Get the average price of the boxes of each warehouse.
f. Obtain the codes of the stores in which the average price of the boxes is greater than \$1000.
g. Get the code of each box along with the name of the city where it is located.
h. Get the number of boxes in each warehouse.
i. Get the codes of the boxes that are in Naples.
j. Insert a new warehouse in Bonita Springs with a capacity of 200 cubic meters.
k. Insert a new box with ***Paper*** content, value \$200, and located in warehouse with ID = 2.
l. Lower the value of all boxes by 15%.
m. Decrease by 20% the price of all the boxes whose price is higher than the average price of all boxes.
n. Eliminate all boxes whose value is less than \$200.

5. Consider the database developed to manage the information regarding a car shop (customers, cars and revisions). Insert enough data into the database to create the following queries against the database using MySQL Workbench or phpMyAdmin.
   a. Show only the fields plate, make and model of cars.
   b. Modify the previous query to add the price, and only show those cars make Ford.
   c. Modify the previous query to get the Ford that have a price higher than 260000.
   d. Show the last names and city of those customers who have purchased Ford or Citroën, customers must appear alphabetically within each city.
   e. Show how many cars have been sold, the profits for these sales, and the average amount sold, should not count the cars of the Citroën brand.
   f. Show how many revisions have been made regarding oil change and how many regarding filter changes.
   g. Obtain a record of all revisions done to cars make Ford.
   h. Increase the price to Ford cars by 5%.
   i. Remove the records of revisions made to Ford cars.

6. Create a database to manage the information related to a sales store. The store is visited by several customers whose can be assisted by salesmen who work at the store. The customers make orders, and it is desired to store the following information about them: ID, purchase amount and order date. Regarding the customers is desired to store: ID, full name, city and grade (like customer classification). The needed information about salesmen is: ID, full name, city and commission.

   Insert the following information into the tables:

Customers:

Full Name	**City**	**Grade**
Nick Rimando	New York	100
Graham Zusi	California	200
Brad Guzan	London	
Fabian Johns	Paris	300
Brad Davis	New York	200
Geoff Camero	Berlin	100
Julian Green	London	300
Jozy Altidor	Moscow	200

Salesmen:

Full Name	**City**	**Commission**
James Hoog	New York	0.15
Nail Knite	Paris	0.13
Pit Alex	London	0.11
Mc Lyon	Paris	0.14
Lauson Hen	Naples	0.12
Paul Adam	Rome	0.13

Orders:

Purchase Amount	**Date**
150.5	2017-10-05
270.65	2017-09-10
65.26	2017-10-05
110.5	2017-08-17
948.5	2017-09-10
2400.6	2017-07-27
5760	2017-09-10
1983.43	2017-10-10
2480.4	2017-10-10
250.45	2017-06-27
75.29	2017-08-17
3045.6	2017-04-25

After creation of the database and filling in the tables, create the following queries against the database:

a. Display name and commission for all the salesmen.
b. Show the columns in a specific order like order date, salesman id, order number and purchase amount from for all the orders.

c. Retrieve the value of salesman id of all salesmen, getting orders from the customers in orders table without any repeats.
d. Get the names and city of salesman, who belongs to the city of Paris.
e. Gather all the information for those customers with a grade of 200.
f. Display the order number followed by order date and the purchase amount for each order which will be delivered by the salesman who is holding the ID = 1.
g. Display all customers with a grade above 100.
h. Show all customers in New York who have a grade value above 100.
i. Get all customers, who are either belongs to the city New York or had a grade above 100.
j. Display all the customers, who are either belongs to the city New York or not had a grade above 100.
k. Show those customers who are neither belongs to the city New York nor grade value is more than 100.
l. Display salesman ID, name, city and commission who gets the commission within the range more than 0.10% and less than 0.12%.
m. Gather the total purchase amount of all orders.
n. Find the average purchase amount of all orders.
o. Find the number of salesmen currently listing for all of their customers.
p. Find the number of customers who gets at least a gradation for his/her performance.
q. Get the maximum purchase amount of all the orders.
r. Find those customers with their name and those salesmen with their name and city who lives in the same city.
s. Find the names of all customers along with the salesmen who works for them.
t. Display all those orders by the customers not located in the same cities where their salesmen live.
u. Find out each order number followed by the name of the customers who made the order.
v. Show a list with salesman name, customer name and their cities for the salesmen and customer who belongs to the same city.
w. Make a list with order no, purchase amount, customer name and their cities for those orders which order amount between 500 and 2000.
x. List which salesman are working for which customer.
y. Display all the orders from the orders table issued by the salesman 'Paul Adam'.
z. Display all the orders for the salesman who belongs to the city London.