Extra SQL exercises

1. List all orders where delivery address is “59 Rue De l’Abbaye” and city “Rio de Janeiro”. Plus at the end of the table append only the name of the company which transported the order.

Solution:

**SELECT** o.\*, s.CompanyName

**FROM** orders o

**INNER** **JOIN** shippers s **ON** o.ShipVia = s.ShipperID

**WHERE** (o.ShipAddress = '59 Rue de l`abbaye' **or** o.ShipCity ='Rio de Janeiro') **AND** (o.ShippedDate< o.RequiredDate);

o.ShippedDate< o.RequiredDate е додадено ѕатоа што пишуваше секоја компанија која ја транспортирала нарачката, претпоставувам навремено.

1. Find the minimal, the average and the maximal price of the product per each category. Display the result into the following fomat: ‘[Category]’, ‘[Minimal price of existing product]’, ‘[Maximal price of existing product]’ and sort it by the maximal price of each existing product in ascending order.

Solution:

**SELECT** cat.CategoryName **AS** '[Category]', **MIN**(p.UnitPrice) **AS** '[Minimal price of existing product]', **AVG** (p.UnitPrice), **MAX**(p.UnitPrice) **AS** '[Maximal price of existing product]'

**FROM** products p

**JOIN** categories cat **ON** p.CategoryID=cat.CategoryID

**GROUP** **BY** cat.CategoryName

**ORDER** **BY** '[Maximal price of existing product]';

1. List all products in following format: '[Name of product]','[Price before discount]','[Price with discount]' where only on the products with price over 100 you'll add -10% discount. Sort the result by the price of the product in descending order.

Solution:

**SELECT** p.ProductName **AS** '[Name of the product]', p.UnitPrice **AS** '[Price before discount]', (**case** **when** p.UnitPrice>100 **then** (p.UnitPrice-(p.UnitPrice \*(10/100))) **ELSE** p.UnitPrice **END**) **AS** '[Price after discount]'

**FROM** products p

**ORDER** **BY** p.UnitPrice **DESC**;

1. List the first name, the last name and the total number for delivered orders by each employee. Sort the result so that at the top will be the employee who has delivered the majority of the orders. Display the result in following format: '[First name]', '[Last name]', '[Total number of delivered orders]'.

Solution:

**SELECT** e.FirstName, e.LastName, **COUNT**(o.OrderID) **AS** '[Total number of delivered orders]'

**FROM** employees e

**INNER** **JOIN** orders o **ON** e.EmployeeID=o.EmployeeID

**WHERE** o.ShippedDate< o.RequiredDate

**GROUP** **BY** e.EmployeeID

**ORDER** **BY** **COUNT**(o.OrderID) **DESC**;

1. Write a SQL statement to display employee ID, first name and last name of the employee who has brought the most income from orders, to the company.

Solution:

**SELECT** e.EmployeeID, e.FirstName, e.LastName, **SUM**(od.UnitPrice\*od.Quantity) **AS** TotalIncome

**FROM** employees e

**JOIN** orders o **ON** e.EmployeeID=o.EmployeeID

**JOIN** `order details` od **ON** o.OrderID=od.OrderId

**GROUP** **BY** e.EmployeeID

**ORDER** **BY** **SUM**(od.UnitPrice\*od.Quantity) **DESC**

**LIMIT** 1;

II nacin kade sto prikazuva SAMO koj e vraboteniot koj napravil najmnogu naracki

**Select** result.eid **as** Employeeid, Result.fname **as** firstname, result.lname **as** Lastname **from** (**SELECT** e.EmployeeID **as** eid, e.FirstName **as** fname, e.LastName **as** lname, **SUM**(od.UnitPrice\*od.Quantity) **AS** TotalIncome

**FROM** employees e

**JOIN** orders o **ON** e.EmployeeID=o.EmployeeID

**JOIN** `order details` od **ON** o.OrderID=od.OrderId

**GROUP** **BY** e.EmployeeID

**ORDER** **BY** **SUM**(od.UnitPrice\*od.Quantity) **DESC**

**LIMIT** 1) **as** result

1. Write a SQL statement to display all employees who were not involved in any order.

Solution:

**SELECT** \*

**FROM** employees e

**WHERE** **not** e.EmployeeID **IN** (**SELECT** **DISTINCT** o.EmployeeID

**FROM** orders o)

1. Write a SQL statement to display all employees and customers who were involved in a same order and are from the same city.

Solution:

**SELECT** e.FirstName, e.LastName

**FROM** employees e

**JOIN** orders o **ON** e.EmployeeID=o.EmployeeID

**JOIN** customers c **ON** o.CustomerID=c.CustomerID

**WHERE** c.City=e.City;

1. Write a SQL statement to display the IDs of all employees who have worked with more than 50 different customers.

Solution:

**SELECT** result.eid **AS** employeeid **FROM** (**SELECT** e.employeeid **AS** eid, **COUNT**(**DISTINCT** customerid)

**FROM** employees e

**INNER** **JOIN** orders o **ON** e.Employeeid=o.Employeeid

**GROUP** **BY** eid

**HAVING** **COUNT**(customerid) > 50) result