

ΦΡΟΝΤΙΣΤΗΡΙΟ SQL

Ο ιστοχώρος <http://w3schools.com/sql/> είναι αφιερωμένος στην SQL. Περιέχει μία εκτενή εισαγωγή με παραδείγματα για «όλες» τις εντολές SQL προς μία βάση δεδομένων αποτελούμενη από 8 σχέσεις με συγκεκριμένα στιγμιότυπα (τα στιγμιότυπα μπορούν να βρεθούν στους συνδέσμους στα αριστερά της σελίδας http://w3schools.com/sql/trysql.asp?filename=trysql_select_all). Ο χρήστης μπορεί να εκτελέσει άμεσα τις εντολές του και να δει το αποτέλεσμα με τους προκύπτοντες πίνακες. Οι σχέσεις έχουν το εξής σχήμα:

- Customers(CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)
 - Categories(CategoryID, CategoryName, Description)
 - Employees(EmployeeID, LastName, FirstName, BirthDate, Photo, Notes)
 - OrderDetails(OrderDetailID, OrderID, ProductID, Quantity)
 - Orders(OrderID, CustomerID, EmployeeID, OrderDate, ShipperID)
 - Products(ProductID, ProductName, SupplierID, CategoryID, Unit, Price)
 - Shippers(ShipperID, ShipperName, Phone)
 - Suppliers(SupplierID, SupplierName, ContactName, Address, City, PostalCode, Country, Phone)
1. Ποιά είναι τα κλειδιά και τα ξένα κλειδιά στις σχέσεις αυτές. Να βρεθεί το Διάγραμμα Οντοτήτων-Συσχετίσεων από το οποίο προήλθαν αυτές οι σχέσεις.
 2. Ο επόμενος πίνακας περιέχει ερωτήματα SQL. Αρχικά, στην αριστερή στήλη είναι το ερώτημα στα αγγλικά και στη δεξιά στήλη η απάντηση. Τα ερωτήματα και οι απαντήσεις αυτές έχουν ληφθεί από το ανωτέρω URL. Συμπληρωματικά, στο τέλος του πίνακα υπάρχουν ερωτήματα στα ελληνικά και οι αντίστοιχες απαντήσεις τους.

select all customers with a City starting with the letter "s"	SELECT * FROM Customers WHERE City LIKE '%s';
select all customers with a Country containing the pattern "land"	SELECT * FROM Customers WHERE Country LIKE '%land%';
select all customers with a Country NOT containing the pattern "land"	SELECT * FROM Customers WHERE Country NOT LIKE '%land%';
select all customers with a City of "Paris" or "London"	SELECT * FROM Customers WHERE City IN ('Paris', 'London');
select all products with a price BETWEEN 10 and 20, but products with a CategoryID of 1,2, or 3 should not be displayed	SELECT * FROM Products WHERE (Price BETWEEN 10 AND 20) AND NOT CategoryID IN (1,2,3);
select all customers with orders	SELECT Customers.CustomerName, Orders.OrderID FROM Customers JOIN Orders ON Customers.CustomerID=Orders.CustomerID ORDER BY Customers.CustomerName;
select all different cities from the "Customers" and the "Suppliers" tables	SELECT City FROM Customers UNION SELECT City FROM Suppliers ORDER BY City;
selects the "ProductName" and "Price" records that have an above average price	SELECT ProductName, Price FROM Products WHERE Price>(SELECT AVG(Price) FROM Products);
count the number of unique customers in the "Orders" table	SELECT COUNT(DISTINCT CustomerID) AS NumberOfCustomers FROM Orders;

get the largest value of the "Price" column from the "Products" table	SELECT MAX(Price) AS HighestPrice FROM Products;
get the smallest value of the "Price" column from the "Products" table	SELECT MIN(Price) AS SmallestOrderPrice FROM Products;
find the sum of all the "Quantity" fields for the "OrderDetails" table	SELECT SUM(Quantity) AS TotalItemsOrdered FROM OrderDetails;
count as orders grouped by shippers	SELECT Shippers.ShipperName, COUNT(Orders.OrderID) AS NumberOfOrders FROM Orders LEFT JOIN Shippers ON Orders.ShipperID=Shippers.ShipperID GROUP BY ShipperName;
	SELECT Shippers.ShipperName, Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders FROM Orders JOIN Shippers ON Orders.ShipperID=Shippers.ShipperID JOIN Employees ON Orders.EmployeeID=Employees.EmployeeID GROUP BY ShipperName,LastName;
find if any of the employees has registered more than 10 orders	SELECT Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders FROM Orders JOIN Employees ON Orders.EmployeeID=Employees.EmployeeID GROUP BY LastName HAVING COUNT(Orders.OrderID) > 10;
find if the employees "Davolio" or "Fuller" have more than 25 orders	SELECT Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders FROM Orders JOIN Employees ON Orders.EmployeeID=Employees.EmployeeID WHERE LastName='Davolio' OR LastName='Fuller' GROUP BY LastName HAVING COUNT(Orders.OrderID) > 25;
Βρες τις ποσότητες ανά κωδικό προϊόντος	SELECT Products.productID, COUNT(quantity) FROM Products, OrderDetails WHERE Products.productID=OrderDetails.productID GROUP BY Products.productid;
Βρες τους κωδικούς πελατών, τους κωδικούς προμηθευτών και την πόλη, οι οποίοι βρίσκονται στην ίδια πόλη	SELECT CustomerID, SupplierID, Customers.city FROM Customers, Suppliers WHERE Customers.city=Suppliers.city;
Βρες την πόλη και το πλήθος των ζευγών πελάτη-προμηθευτή που βρίσκονται στην ίδια πόλη	SELECT Suppliers.city, Count(Suppliers.city) FROM Customers, Suppliers WHERE Customers.city=Suppliers.city GROUP BY Suppliers.city;
Βρες τα ζεύγη πελατών και τις πόλεις τους ώστε να μην είναι στην ίδια πόλη	SELECT B.CustomerName AS pelatis1, A.CustomerName AS pelatis2, A.city AS city1, B.city AS city2 FROM Customers AS A, Customers AS B WHERE city1<>city2 ORDER BY city1;