LEVEL A (Answers)

-- 1. List of all customers

SELECT \* FROM Customers;

-- 2. List of all customers where company name ends in N

SELECT \* FROM Customers

WHERE CompanyName LIKE '%N';

-- 3. List of all customers who live in Berlin or London

SELECT \* FROM Customers

WHERE City IN ('Berlin', 'London');

-- 4. List of all customers who live in UK or USA

SELECT \* FROM Customers

WHERE Country IN ('UK', 'USA');

-- 5. List of all products sorted by product name

SELECT \* FROM Products

ORDER BY ProductName;

-- 6. List of all products where product name starts with an A

SELECT \* FROM Products

WHERE ProductName LIKE 'A%';

-- 7. List of customers who ever placed an order

SELECT DISTINCT Customers.\*

FROM Customers

JOIN Orders ON Customers.CustomerID = Orders.CustomerID;

-- 8. List of customers who live in London and have bought chai

SELECT DISTINCT Customers.\*

FROM Customers

JOIN Orders ON Customers.CustomerID = Orders.CustomerID

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN Products ON OrderDetails.ProductID = Products.ProductID

WHERE Customers.City = 'London' AND Products.ProductName = 'Chai';

-- 9. List of customers who never placed an order

SELECT \*

FROM Customers

WHERE CustomerID NOT IN (SELECT DISTINCT CustomerID FROM Orders);

-- 10. List of customers who ordered Tofu

SELECT DISTINCT Customers.\*

FROM Customers

JOIN Orders ON Customers.CustomerID = Orders.CustomerID

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN Products ON OrderDetails.ProductID = Products.ProductID

WHERE Products.ProductName = 'Tofu';

-- 11. Details of first order of the system

SELECT TOP 1 \*

FROM Orders

ORDER BY OrderDate ASC;

-- 12. Find the details of most expensive order date

SELECT TOP 1 Orders.\*, SUM(OrderDetails.UnitPrice \* OrderDetails.Quantity) AS TotalPrice

FROM Orders

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

GROUP BY Orders.OrderID, Orders.OrderDate

ORDER BY TotalPrice DESC;

-- 13. For each order get the OrderID and Average quantity of items in that order

SELECT OrderID, AVG(Quantity) AS AverageQuantity

FROM OrderDetails

GROUP BY OrderID;

-- 14. For each order get the OrderID, minimum quantity and maximum quantity for that order

SELECT OrderID, MIN(Quantity) AS MinQuantity, MAX(Quantity) AS MaxQuantity

FROM OrderDetails

GROUP BY OrderID;

-- 15. Get a list of all managers and total number of employees who report to them

SELECT ManagerID, COUNT(\*) AS NumberOfReports

FROM Employees

WHERE ManagerID IS NOT NULL

GROUP BY ManagerID;

-- 16. Get the OrderID and the total quantity for each order that has a total quantity of greater than 300

SELECT OrderID, SUM(Quantity) AS TotalQuantity

FROM OrderDetails

GROUP BY OrderID

HAVING SUM(Quantity) > 300;

-- 17. List of all orders placed on or after 1996/12/31

SELECT \*

FROM Orders

WHERE OrderDate >= '1996-12-31';

-- 18. List of all orders shipped to Canada

SELECT \*

FROM Orders

WHERE ShipCountry = 'Canada';

-- 19. List of all orders with order total > 200

SELECT Orders.OrderID, SUM(OrderDetails.UnitPrice \* OrderDetails.Quantity) AS TotalPrice

FROM Orders

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

GROUP BY Orders.OrderID

HAVING SUM(OrderDetails.UnitPrice \* OrderDetails.Quantity) > 200;

-- 20. List of countries and sales made in each country

SELECT Country, SUM(OrderDetails.UnitPrice \* OrderDetails.Quantity) AS TotalSales

FROM Orders

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

GROUP BY Country;

-- 21. List of Customer ContactName and number of orders they placed

SELECT Customers.ContactName, COUNT(Orders.OrderID) AS NumberOfOrders

FROM Customers

JOIN Orders ON Customers.CustomerID = Orders.CustomerID

GROUP BY Customers.ContactName;

-- 22. List of customer contactnames who have placed more than 3 orders

SELECT Customers.ContactName

FROM Customers

JOIN Orders ON Customers.CustomerID = Orders.CustomerID

GROUP BY Customers.ContactName

HAVING COUNT(Orders.OrderID) > 3;

-- 23. List of discontinued products which were ordered between 1/1/1997 and 1/1/1998

SELECT DISTINCT Products.\*

FROM Products

JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID

JOIN Orders ON OrderDetails.OrderID = Orders.OrderID

WHERE Products.Discontinued = 1

  AND Orders.OrderDate BETWEEN '1997-01-01' AND '1998-01-01';

-- 24. List of employee firstname, lastname, supervisor firstname, lastname

SELECT e.FirstName AS EmployeeFirstName, e.LastName AS EmployeeLastName,

       s.FirstName AS SupervisorFirstName, s.LastName AS SupervisorLastName

FROM Employees e

LEFT JOIN Employees s ON e.ReportsTo = s.EmployeeID;

-- 25. List of Employees id and total sale conducted by employee

SELECT Orders.EmployeeID, SUM(OrderDetails.UnitPrice \* OrderDetails.Quantity) AS TotalSales

FROM Orders

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

GROUP BY Orders.EmployeeID;

-- 26. List of employees whose FirstName contains character a

SELECT \*

FROM Employees

WHERE FirstName LIKE '%a%';

-- 27. List of managers who have more than four people reporting to them

SELECT ManagerID

FROM Employees

WHERE ManagerID IS NOT NULL

GROUP BY ManagerID

HAVING COUNT(\*) > 4;

-- 28. List of Orders and ProductNames

SELECT Orders.OrderID, Products.ProductName

FROM Orders

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN Products ON OrderDetails.ProductID = Products.ProductID;

-- 29. List of orders placed by the best customer

SELECT Orders.\*

FROM Orders

WHERE Orders.CustomerID = (

    SELECT TOP 1 CustomerID

    FROM Orders

    GROUP BY CustomerID

    ORDER BY COUNT(OrderID) DESC

);

-- 30. List of orders placed by customers who do not have a Fax number

SELECT Orders.\*

FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

WHERE Customers.Fax IS NULL;

-- 31. List of Postal codes where the product Tofu was shipped

SELECT DISTINCT Customers.PostalCode

FROM Orders

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN Products ON OrderDetails.ProductID = Products.ProductID

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

WHERE Products.ProductName = 'Tofu';

-- 32. List of product Names that were shipped to France

SELECT DISTINCT Products.ProductName

FROM Orders

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN Products ON OrderDetails.ProductID = Products.ProductID

WHERE Orders.ShipCountry = 'France';

-- 33. List of ProductNames and Categories for the supplier 'Specialty Biscuits, Ltd.'

SELECT Products.ProductName, Categories.CategoryName

FROM Products

JOIN Suppliers ON Products.SupplierID = Suppliers.SupplierID

JOIN Categories ON Products.CategoryID = Categories.CategoryID

WHERE Suppliers.CompanyName = 'Specialty Biscuits, Ltd.';

-- 34. List of products that were never ordered

SELECT Products.\*

FROM Products

LEFT JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID

WHERE OrderDetails.ProductID IS NULL;

-- 35. List of products where units in stock is less than 10 and units on order are 0

SELECT \*

FROM Products

WHERE UnitsInStock < 10 AND UnitsOnOrder = 0;

-- 36. List of top 10 countries by sales

SELECT Country, SUM(OrderDetails.UnitPrice \* OrderDetails.Quantity) AS TotalSales

FROM Orders

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

GROUP BY Country

ORDER BY TotalSales DESC

LIMIT 10;

-- 37. Number of orders each employee has taken for customers with CustomerIDs between A and AO

SELECT Orders.EmployeeID, COUNT(Orders.OrderID) AS NumberOfOrders

FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

WHERE Customers.CustomerID BETWEEN 'A' AND 'AO'

GROUP BY Orders.EmployeeID;

-- 38. Order date of most expensive order

SELECT TOP 1 Orders.OrderDate

FROM Orders

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

GROUP BY Orders.OrderDate

ORDER BY SUM(OrderDetails.UnitPrice \* OrderDetails.Quantity) DESC;

-- 39. Product name and total revenue from that product

SELECT Products.ProductName, SUM(OrderDetails.UnitPrice \* OrderDetails.Quantity) AS TotalRevenue

FROM OrderDetails

JOIN Products ON OrderDetails.ProductID = Products.ProductID

GROUP BY Products.ProductName;

-- 40. Supplier ID and number of products offered

SELECT SupplierID, COUNT(\*) AS NumberOfProducts

FROM Products

GROUP BY SupplierID;

-- 41. Top ten customers based on their business

SELECT TOP 10 Customers.CustomerID, SUM(OrderDetails.UnitPrice \* OrderDetails.Quantity) AS TotalBusiness

FROM Orders

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

GROUP BY Customers.CustomerID

ORDER BY TotalBusiness DESC;

-- 42. What is the total revenue of the company

SELECT SUM(OrderDetails.UnitPrice \* OrderDetails.Quantity) AS TotalRevenue

FROM OrderDetails;

-- 43. List of employees who do not have a supervisor

SELECT \*

FROM Employees

WHERE ReportsTo IS NULL;

-- 44. List of products with their suppliers

SELECT Products.ProductName, Suppliers.CompanyName

FROM Products

JOIN Suppliers ON Products.SupplierID = Suppliers.SupplierID;

-- 45. List of products and their total sales

SELECT Products.ProductName, SUM(OrderDetails.UnitPrice \* OrderDetails.Quantity) AS TotalSales

FROM Products

JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID

GROUP BY Products.ProductName;