## Basic SQL Assignments (AdventureWorks)

1. List all employees’ first and last names from Person.Person table.

SELECT FirstName, LastName FROM Person.Person;

1. Retrieve the first 10 products' names and list prices.

SELECT TOP 10 Name, ListPrice FROM Production.Product;

1. Show all unique job titles from HumanResources.Employee.

SELECT DISTINCT JobTitle FROM HumanResources.Employee;

1. Display addresses in 'Seattle' city.

SELECT \* FROM Person.Address WHERE City = 'Seattle';

1. Count the number of rows in the Production.Product table.

SELECT COUNT(\*) FROM Production.Product;

1. Find employees hired after 2014-01-01.

SELECT BusinessEntityID, HireDate FROM HumanResources.Employee WHERE HireDate > '2014-01-01';

1. Show departments in alphabetical order.

SELECT Name FROM HumanResources.Department ORDER BY Name;

1. List sales order IDs and their total due from 2013 onwards.

SELECT SalesOrderID, TotalDue FROM Sales.SalesOrderHeader WHERE OrderDate >= '2013-01-01';

1. Display all products where the color is Red.

SELECT ProductID, Name FROM Production.Product WHERE Color = 'Red';

1. List all employees with missing middle names.

SELECT FirstName, LastName FROM Person.Person WHERE MiddleName IS NULL;

1. Show the top 5 customers with the highest total due (from Sales.SalesOrderHeader).

SELECT TOP 5 CustomerID, SUM(TotalDue) AS TotalSpent

FROM Sales.SalesOrderHeader

GROUP BY CustomerID

ORDER BY TotalSpent DESC;

1. List all subcategories for 'Bikes'.

SELECT ps.Name

FROM Production.ProductSubcategory ps

JOIN Production.ProductCategory pc ON ps.ProductCategoryID = pc.ProductCategoryID

WHERE pc.Name = 'Bikes';

1. How many products are of Size 'L'?

SELECT COUNT(\*) FROM Production.Product WHERE Size = 'L';

1. Retrieve emails for contacts with a '.com' address.

SELECT EmailAddress FROM Person.EmailAddress WHERE EmailAddress LIKE '%.com';

1. Get the list of employees and their departments .

SELECT p.FirstName, p.LastName, d.Name AS Department

FROM HumanResources.EmployeeDepartmentHistory edh

JOIN HumanResources.Department d ON edh.DepartmentID = d.DepartmentID

JOIN Person.Person p ON edh.BusinessEntityID = p.BusinessEntityID

WHERE edh.EndDate IS NULL;

1. Input: DepartmentID = 3; Output: All employees in dept 3 with their hire dates. How do you do it?

SELECT e.BusinessEntityID, p.FirstName, p.LastName, e.HireDate

FROM HumanResources.EmployeeDepartmentHistory edh

JOIN HumanResources.Employee e ON edh.BusinessEntityID = e.BusinessEntityID

JOIN Person.Person p ON e.BusinessEntityID = p.BusinessEntityID

WHERE edh.DepartmentID = 3 AND edh.EndDate IS NULL;

1. List all phone numbers for person with LastName 'Smith'.

SELECT pa.PhoneNumber

FROM Person.PersonPhone pa

JOIN Person.Person p ON pa.BusinessEntityID = p.BusinessEntityID

WHERE p.LastName = 'Smith';

1. Find the most recently hired employee.

SELECT TOP 1 p.FirstName, p.LastName, e.HireDate

FROM HumanResources.Employee e

JOIN Person.Person p ON e.BusinessEntityID = p.BusinessEntityID

ORDER BY HireDate DESC;

1. Show all vendors and their credit rating.

SELECT Name, CreditRating FROM Purchasing.Vendor;

1. List all product models created since 2013.

SELECT Name, CatalogDescription FROM Production.ProductModel WHERE ModifiedDate >= '2013-01-01';

1. Count how many employees per department (current only).

SELECT d.Name AS Department, COUNT(\*) AS NumEmployees

FROM HumanResources.Department d

JOIN HumanResources.EmployeeDepartmentHistory edh ON d.DepartmentID = edh.DepartmentID

WHERE edh.EndDate IS NULL

GROUP BY d.Name;

1. Display all purchase orders for vendor 'Litware, Inc.'.

SELECT poh.PurchaseOrderID, poh.OrderDate

FROM Purchasing.PurchaseOrderHeader poh

JOIN Purchasing.Vendor v ON poh.VendorID = v.BusinessEntityID

WHERE v.Name = 'Litware, Inc.';

1. Which product has the highest list price?

SELECT TOP 1 Name, ListPrice

FROM Production.Product

ORDER BY ListPrice DESC;

1. Input: Show sales orders placed on 2014-06-15, output as OrderID, CustomerID, TotalDue.

SELECT SalesOrderID, CustomerID, TotalDue

FROM Sales.SalesOrderHeader

WHERE OrderDate = '2014-06-15';

1. Get all employees who have a title containing 'Manager'.

SELECT p.FirstName, p.LastName, e.JobTitle

FROM HumanResources.Employee e

JOIN Person.Person p ON e.BusinessEntityID = p.BusinessEntityID

WHERE e.JobTitle LIKE '%Manager%';

1. Find customers who have placed more than 10 orders.

SELECT CustomerID, COUNT(\*) AS Orders

FROM Sales.SalesOrderHeader

GROUP BY CustomerID

HAVING COUNT(\*) > 10;

1. List all persons with NULL email addresses.

SELECT p.BusinessEntityID, p.FirstName, p.LastName

FROM Person.Person p

LEFT JOIN ea ON p.BusinessEntityID = ea.BusinessEntityID

WHERE ea.EmailAddress IS NULL;

1. Which five products have the lowest standard cost?

SELECT TOP 5 Name, StandardCost FROM x ORDER BY StandardCost ASC;

1. List all orders along with the total line item quantity for each.

SELECT SalesOrderID, SUM(OrderQty) AS TotalQty

FROM Sales.SalesOrderDetail

GROUP BY SalesOrderID;

1. Show all products that have never been sold (i.e., not in SalesOrderDetail).

SELECT p.ProductID, p.Name

FROM Production.Product p

LEFT JOIN Sales.SalesOrderDetail sd ON p.ProductID = sd.ProductID

WHERE sd.ProductID IS NULL;

1. Input: CustomerID = 30025, Output: Last 3 orders with TotalDue.

SELECT TOP 3 SalesOrderID, OrderDate, TotalDue

FROM Sales.SalesOrderHeader

WHERE CustomerID = 30025

ORDER BY OrderDate DESC;

1. List all addresses in 'Washington' state.

SELECT \* FROM Person.Address WHERE StateProvinceID IN (SELECT StateProvinceID FROM Person.StateProvince WHERE Name = 'Washington');

1. Get the distinct sizes of products available.

SELECT DISTINCT Size FROM Production.Product WHERE Size IS NOT NULL;

1. Find the number of products for each subcategory.

SELECT ProductSubcategoryID, COUNT(\*) AS ProductCount

FROM Production.Product

GROUP BY ProductSubcategoryID;

1. “This is the input data: List of ‘Accessories’ subcategory products. This is the output: ProductName, ListPrice. How do you achieve this?”

SELECT p.Name, p.ListPrice

FROM Production.Product p

JOIN Production.ProductSubcategory s ON p.ProductSubcategoryID = s.ProductSubcategoryID

WHERE s.Name = 'Accessories';

## Advanced SQL Assignments (AdventureWorks)

1. List employees who have worked in more than one department.

SELECT BusinessEntityID

FROM HumanResources.EmployeeDepartmentHistory

GROUP BY BusinessEntityID

HAVING COUNT(DISTINCT DepartmentID) > 1;

1. Get the average hire year for each job title.

SELECT JobTitle, AVG(YEAR(HireDate)) AS AvgHireYear

FROM HumanResources.Employee

GROUP BY JobTitle;

1. Show the highest paid employee(s), including job title, department, and pay frequency.

SELECT TOP 1 WITH TIES e.BusinessEntityID, e.JobTitle, p.Rate, p.PayFrequency, d.Name AS Department

FROM HumanResources.EmployeePayHistory p

JOIN HumanResources.Employee e ON p.BusinessEntityID = e.BusinessEntityID

JOIN HumanResources.EmployeeDepartmentHistory edh ON e.BusinessEntityID = edh.BusinessEntityID

JOIN HumanResources.Department d ON edh.DepartmentID = d.DepartmentID

WHERE edh.EndDate IS NULL

ORDER BY p.Rate DESC;

1. Which sales territory has the highest total sales?

SELECT st.Name, SUM(soh.TotalDue) AS TotalSales

FROM Sales.SalesOrderHeader soh

JOIN Sales.SalesTerritory st ON soh.TerritoryID = st.TerritoryID

GROUP BY st.Name

ORDER BY TotalSales DESC;

1. List top 3 selling products for each year.

WITH ProductSales AS (

SELECT YEAR(soh.OrderDate) AS Year, p.Name, SUM(sod.OrderQty) AS TotalSold

FROM Sales.SalesOrderHeader soh

JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID

JOIN Production.Product p ON sod.ProductID = p.ProductID

GROUP BY YEAR(soh.OrderDate), p.Name

),

Ranked AS (

SELECT \*, ROW\_NUMBER() OVER (PARTITION BY Year ORDER BY TotalSold DESC) AS rn

FROM ProductSales

)

SELECT Year, Name, TotalSold FROM Ranked WHERE rn <= 3;

1. Show all customers who have never placed an order.

SELECT c.CustomerID

FROM Sales.Customer c

LEFT JOIN Sales.SalesOrderHeader soh ON c.CustomerID = soh.CustomerID

WHERE soh.SalesOrderID IS NULL;

1. List employees who are also vendors.

SELECT DISTINCT p.FirstName, p.LastName

FROM Person.Person p

JOIN Purchasing.Vendor v ON p.BusinessEntityID = v.BusinessEntityID

JOIN HumanResources.Employee e ON p.BusinessEntityID = e.BusinessEntityID;

1. Find the total and average sales per product category.

SELECT c.Name AS Category, SUM(sod.LineTotal) AS TotalSales, AVG(sod.LineTotal) AS AverageSales

FROM Sales.SalesOrderDetail sod

JOIN Production.Product p ON sod.ProductID = p.ProductID

JOIN Production.ProductSubcategory s ON p.ProductSubcategoryID = s.ProductSubcategoryID

JOIN Production.ProductCategory c ON s.ProductCategoryID = c.ProductCategoryID

GROUP BY c.Name;

1. Show the top 5 customers who buy the most expensive products.

SELECT TOP 5 soh.CustomerID, SUM(sod.UnitPrice) AS ExpensivePurchase

FROM Sales.SalesOrderHeader soh

JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID

GROUP BY soh.CustomerID

ORDER BY ExpensivePurchase DESC;

1. “Input: ProductID 750; Output: Quantity ordered in last 3 years, by year.”

SELECT YEAR(soh.OrderDate) AS Year, SUM(sod.OrderQty) AS OrderedQty

FROM Sales.SalesOrderHeader soh

JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID

WHERE sod.ProductID = 750 AND soh.OrderDate >= DATEADD(YEAR, -3, GETDATE())

GROUP BY YEAR(soh.OrderDate);

1. Rank employees by how long they've worked in the company.

SELECT p.FirstName, p.LastName, e.HireDate, RANK() OVER (ORDER BY e.HireDate) AS SeniorityRank

FROM HumanResources.Employee e

JOIN Person.Person p ON e.BusinessEntityID = p.BusinessEntityID;

1. Find the average order value for each customer.

SELECT CustomerID, AVG(TotalDue) AS AvgOrderValue

FROM Sales.SalesOrderHeader

GROUP BY CustomerID;

1. Show the month-over-month sales growth for 2014.

WITH MonthlySales AS (

SELECT MONTH(OrderDate) AS Month, SUM(TotalDue) AS Sales

FROM Sales.SalesOrderHeader

WHERE YEAR(OrderDate) = 2014

GROUP BY MONTH(OrderDate)

)

SELECT Month, Sales,

LAG(Sales) OVER (ORDER BY Month) AS PrevMonthSales,

(Sales - LAG(Sales) OVER (ORDER BY Month)) AS Growth

FROM MonthlySales;

1. Find customers who ordered the same product more than once in a year.

SELECT soh.CustomerID, sod.ProductID, YEAR(soh.OrderDate) AS Year, COUNT(\*) AS Orders

FROM Sales.SalesOrderHeader soh

JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID

GROUP BY soh.CustomerID, sod.ProductID, YEAR(soh.OrderDate)

HAVING COUNT(\*) > 1;

1. Find top 5 salespersons by their total sales in 2013.

SELECT TOP 5 soh.SalesPersonID, SUM(soh.TotalDue) AS TotalSales

FROM Sales.SalesOrderHeader soh

WHERE YEAR(soh.OrderDate) = 2013 AND soh.SalesPersonID IS NOT NULL

GROUP BY soh.SalesPersonID

ORDER BY TotalSales DESC;

1. Given: List of product names and list prices. Output: Show only those with list price > average. How to achieve?

SELECT Name, ListPrice

FROM Production.Product

WHERE ListPrice > (SELECT AVG(ListPrice) FROM Production.Product);

1. For each year, show number of new products introduced.

SELECT YEAR(SellStartDate) AS Year, COUNT(\*) AS NewProducts

FROM Production.Product

GROUP BY YEAR(SellStartDate);

1. Show products with a price change history (list those with more than one price).

SELECT ProductID

FROM Production.ProductListPriceHistory

GROUP BY ProductID

HAVING COUNT(DISTINCT ListPrice) > 1;

1. Which vendors supplied a product that was ordered more than 10 times?

SELECT DISTINCT v.Name

FROM Purchasing.ProductVendor pv

JOIN Purchasing.Vendor v ON pv.BusinessEntityID = v.BusinessEntityID

JOIN Production.Product p ON pv.ProductID = p.ProductID

JOIN Sales.SalesOrderDetail sod ON p.ProductID = sod.ProductID

GROUP BY v.Name, sod.ProductID

HAVING SUM(sod.OrderQty) > 10;

1. Find products which were never discontinued.

SELECT Name FROM Production.Product WHERE SellEndDate IS NULL;

1. List all retired employees and their department at retirement.

SELECT p.FirstName, p.LastName, d.Name AS Department, edh.EndDate

FROM HumanResources.EmployeeDepartmentHistory edh

JOIN HumanResources.Department d ON edh.DepartmentID = d.DepartmentID

JOIN Person.Person p ON edh.BusinessEntityID = p.BusinessEntityID

WHERE edh.EndDate IS NOT NULL;

1. Input: A table of ProductID and average order quantity per year. Output: List ProductIDs ordered on average more than 100 units/year. How to do this?

WITH ProductYearAvg AS (

SELECT sod.ProductID, AVG(sod.OrderQty) AS AvgQty

FROM Sales.SalesOrderDetail sod

JOIN Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID

GROUP BY sod.ProductID, YEAR(soh.OrderDate)

)

SELECT ProductID

FROM ProductYearAvg

GROUP BY ProductID

HAVING AVG(AvgQty) > 100;

1. Which department has the largest average tenure (years)?

SELECT d.Name, AVG(DATEDIFF(YEAR, edh.StartDate, ISNULL(edh.EndDate, GETDATE()))) AS AvgTenure

FROM HumanResources.EmployeeDepartmentHistory edh

JOIN HumanResources.Department d ON edh.DepartmentID = d.DepartmentID

GROUP BY d.Name

ORDER BY AvgTenure DESC;

1. For each customer, return the most expensive product they've ever ordered.

SELECT soh.CustomerID, MAX(sod.UnitPrice) AS MaxPrice

FROM Sales.SalesOrderHeader soh

JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID

GROUP BY soh.CustomerID;

1. Show all products that had no price change since their introduction.

SELECT p.ProductID, p.Name

FROM Production.Product p

JOIN Production.ProductListPriceHistory plph ON p.ProductID = plph.ProductID

GROUP BY p.ProductID, p.Name

HAVING COUNT(DISTINCT plph.ListPrice) = 1;

1. List vendors who supplied products more than 2 years ago but not in the last year.

SELECT v.Name

FROM Purchasing.PurchaseOrderHeader poh

JOIN Purchasing.Vendor v ON poh.VendorID = v.BusinessEntityID

WHERE poh.OrderDate < DATEADD(YEAR, -1, GETDATE())

AND poh.OrderDate >= DATEADD(YEAR, -2, GETDATE());

1. Show the quarter-over-quarter revenue growth for 2014.

WITH QuarterlySales AS (

SELECT DATEPART(QUARTER, OrderDate) AS Quarter, SUM(TotalDue) AS Revenue

FROM Sales.SalesOrderHeader

WHERE YEAR(OrderDate) = 2014

GROUP BY DATEPART(QUARTER, OrderDate)

)

SELECT Quarter, Revenue, LAG(Revenue) OVER (ORDER BY Quarter) AS PrevQuarter,

(Revenue - LAG(Revenue) OVER (ORDER BY Quarter)) AS Growth

FROM QuarterlySales;

1. Find the average number of days to ship an order per year.

SELECT YEAR(OrderDate) AS OrderYear, AVG(DATEDIFF(DAY, OrderDate, ShipDate)) AS AvgDaysToShip

FROM Sales.SalesOrderHeader

WHERE ShipDate IS NOT NULL

GROUP BY YEAR(OrderDate);

1. For each region, count the number of distinct customers who ordered in 2015.

SELECT st.Name AS Region, COUNT(DISTINCT soh.CustomerID) AS NumCustomers

FROM Sales.SalesOrderHeader soh

JOIN Sales.SalesTerritory st ON soh.TerritoryID = st.TerritoryID

WHERE YEAR(soh.OrderDate) = 2015

GROUP BY st.Name;

1. Show the longest streak of consecutive years a product was sold.  
   *(This requires a recursive CTE / advanced logic beyond standard SQL and may not be supported by all SQL engines. For brevity, suggest an aggregate with MIN/MAX instead):*

SELECT ProductID, MIN(YEAR(soh.OrderDate)) AS FirstYear, MAX(YEAR(soh.OrderDate)) AS LastYear, (MAX(YEAR(soh.OrderDate)) - MIN(YEAR(soh.OrderDate)) + 1) AS YearsActive

FROM Sales.SalesOrderDetail sod

JOIN Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID

GROUP BY ProductID;

1. Find which employee has worked in the most different job titles.

SELECT TOP 1 e.BusinessEntityID, COUNT(DISTINCT edh.JobTitle) AS NumTitles

FROM HumanResources.EmployeeDepartmentHistory edh

JOIN HumanResources.Employee e ON edh.BusinessEntityID = e.BusinessEntityID

GROUP BY e.BusinessEntityID

ORDER BY NumTitles DESC;

1. Provide a monthly count of products added in 2023.

SELECT MONTH(SellStartDate) AS Month, COUNT(\*) AS NewProducts

FROM Production.Product

WHERE YEAR(SellStartDate) = 2023

GROUP BY MONTH(SellStartDate);

1. List sales orders where the invoice and ship-to country are different.

SELECT SalesOrderID

FROM Sales.SalesOrderHeader

WHERE ShipToAddressID <> BillToAddressID;

1. Show all customers who ordered a product that was discontinued after their purchase.

SELECT DISTINCT soh.CustomerID

FROM Sales.SalesOrderHeader soh

JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID

JOIN Production.Product p ON sod.ProductID = p.ProductID

WHERE p.SellEndDate IS NOT NULL AND p.SellEndDate > soh.OrderDate;

1. “Input: Table with CustomerID, TotalDue. Output: Only those with total over $10,000. How do you achieve this?”

SELECT CustomerID, SUM(TotalDue) AS TotalAmount

FROM

GROUP BY CustomerID

1. HAVING SUM(TotalDue) > 10000;