1. **What is SQL? What is DML? What is DDL? Recite the most important SQL commands.**

SQL is a declarative language for query and manipulation of relational data.

DML means Data Manipulation Language – SELECT, INSERT, UPDATE, and DELETE;

DDL means Data Definition Language – CREATE, DROP, ALTER, GRANT and REVOKE.

1. **What is Transact-SQL (T-SQL)?**

T-SQL is an extension to the standard SQL language, supports “if” statements, loops and exceptions and is used for writing stored procedures, functions, triggers, etc.

1. **Start SQL Management Studio and connect to the database TelerikAcademy. Examine the major tables in the "TelerikAcademy" database.**

Done.

1. **Write a SQL query to find all information about all departments (use "TelerikAcademy" database).**

SELECT \*

FROM [TelerikAcademy].dbo.Departments;

1. **Write a SQL query to find all department names.**

SELECT Name

FROM [TelerikAcademy].dbo.Departments;

1. **Write a SQL query to find the salary of each employee.**

SELECT Salary

FROM [TelerikAcademy].dbo.Employees;

1. **Write a SQL query to find the full name of each employee.**

SELECT FirstName + ' ' + LastName

FROM [TelerikAcademy].dbo.Employees;

1. **Write a SQL query to find the email addresses of each employee (by his first and last name). Consider that the mail domain is telerik.com. Emails should look like “John.Doe@telerik.com". The produced column should be named "Full Email Addresses".**

SELECT (FirstName + '.' + LastName + '@telerik.com')

AS [Full Email Addresses]

FROM [TelerikAcademy].dbo.Employees;

1. **Write a SQL query to find all different employee salaries.**

SELECT Distinct Salary

FROM [TelerikAcademy].dbo.Employees;

1. **Write a SQL query to find all information about the employees whose job title is “Sales Representative“.**

SELECT \*

FROM [TelerikAcademy].dbo.Employees

WHERE JobTitle = 'Sales Representative';

1. **Write a SQL query to find the names of all employees whose first name starts with "SA".**

SELECT FirstName + ' ' + LastName AS Name

FROM [TelerikAcademy].dbo.Employees

WHERE FirstName like 'SA%';

1. **Write a SQL query to find the names of all employees whose last name contains "ei".**

SELECT FirstName + ' ' + LastName AS Name

FROM [TelerikAcademy].dbo.Employees

WHERE LastName like '%ei%';

1. **Write a SQL query to find the salary of all employees whose salary is in the range [20000…30000].**

SELECT Salary

FROM [TelerikAcademy].dbo.Employees

WHERE Salary between 20000 and 30000;

1. **Write a SQL query to find the names of all employees whose salary is 25000, 14000, 12500 or 23600.**

SELECT FirstName + ' ' + LastName AS Name, Salary

FROM [TelerikAcademy].dbo.Employees

WHERE Salary IN (25000, 14000, 12500, 23600);

1. **Write a SQL query to find all employees that do not have manager.**

SELECT FirstName + ' ' + LastName AS Name

FROM [TelerikAcademy].dbo.Employees

WHERE ManagerID is null;

1. **Write a SQL query to find all employees that have salary more than 50000. Order them in decreasing order by salary.**

SELECT FirstName + ' ' + LastName AS Name, Salary

FROM [TelerikAcademy].dbo.Employees

WHERE Salary > 50000

ORDER BY Salary desc;

1. **Write a SQL query to find the top 5 best paid employees.**

SELECT TOP 5 FirstName + ' ' + LastName AS Name, Salary

FROM [TelerikAcademy].dbo.Employees

ORDER BY Salary desc;

1. **Write a SQL query to find all employees along with their address. Use inner join with ON clause.**

SELECT e.FirstName + ' ' + e.LastName AS Name,

a.AddressText AS Address

FROM [TelerikAcademy].dbo.Employees e

INNER JOIN Addresses a

ON e.AddressID = a.AddressID;

1. **Write a SQL query to find all employees and their address. Use equijoins (conditions in the WHERE clause).**

SELECT FirstName + ' ' + LastName AS Name, AddressText AS Address

FROM [TelerikAcademy].dbo.Employees e,

[TelerikAcademy].dbo.Addresses a

WHERE e.AddressID = a.AddressID;

1. **Write a SQL query to find all employees along with their manager.**

SELECT e.FirstName + ' ' + e.LastName AS Name,

m.FirstName + ' ' + m.LastName AS Manager

FROM [TelerikAcademy].dbo.Employees e

JOIN Employees m

ON m.ManagerID = e.EmployeeID;

1. **Write a SQL query to find all employees, along with their manager and their address. Join the 3 tables: Employees e, Employees m and Addresses a.**

SELECT e.FirstName + ' ' + e.LastName AS Name,

m.FirstName + ' ' + m.LastName AS Manager,

a.AddressText AS Address

FROM [TelerikAcademy].dbo.Employees e

JOIN Employees m

ON m.ManagerID = e.EmployeeID

JOIN Addresses a

ON e.AddressID = a.AddressID;

1. **Write a SQL query to find all departments and all town names as a single list. Use UNION.**

SELECT Name

FROM Departments

UNION

SELECT Name

FROM Towns;

1. **Write a SQL query to find all the employees and the manager for each of them along with the employees that do not have manager. Use right outer join. Rewrite the query to use left outer join.**

SELECT e.FirstName + ' ' + e.LastName AS Employee,

m.FirstName + ' ' + m.LastName AS Manager

FROM Employees e

RIGHT OUTER JOIN Employees m

ON m.ManagerID = e.EmployeeID;

SELECT e.FirstName + ' ' + e.LastName AS Employee,

m.FirstName + ' ' + m.LastName AS Manager

FROM Employees e

Left OUTER JOIN Employees m

ON e.ManagerID = m.EmployeeID;

1. **Write a SQL query to find the names of all employees from the departments "Sales" and "Finance" whose hire year is between 1995 and 2005.**

SELECT FirstName + ' ' + LastName AS Employee,

d.Name AS Department,

e.HireDate

FROM Employees e

INNER JOIN Departments d

ON e.DepartmentID = d.DepartmentID

and d.Name IN ('Sales', 'Finance')

and e.HireDate BETWEEN '1995' AND '2005';