Intro To SQL Homework

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

SQL is a [special-purpose programming language](http://en.wikipedia.org/wiki/Special-purpose_programming_language), designed for managing data held in a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system). SQL consists of a [data definition language](http://en.wikipedia.org/wiki/Data_definition_language) and a [data manipulation language](http://en.wikipedia.org/wiki/Data_manipulation_language). The scope of SQL includes data insert, query, update and delete, [schema](http://en.wikipedia.org/wiki/Database_schema) creation and modification, and data access control.

A **data definition language** or **data description language** (**DDL**) is a syntax similar to a computer [programming language](http://en.wikipedia.org/wiki/Programming_language) for defining [data structures](http://en.wikipedia.org/wiki/Data_structure), especially [database schemas](http://en.wikipedia.org/wiki/Database_schema). SQL, however, uses a collection of imperative verbs whose effect is to modify the schema of the database by adding, changing, or deleting definitions of tables or other objects. These statements can be freely mixed with other SQL statements, so the DDL is not truly a separate language. Statements include: SELECT, DROP, ALTER, RENAME and referential integrity statements.

A **data manipulation language** (**DML**) is a family of syntax elements similar to a computer [programming language](http://en.wikipedia.org/wiki/Programming_language) used for selecting, inserting, deleting and updating data in a [database](http://en.wikipedia.org/wiki/Database). Performing read-only queries of data is sometimes also considered a component of DML. In SQL the DML uses SELECT, INSERT UPDATE and DELETE

**TASK 2. What is Transact-SQL (T-SQL)?**

**Transact-SQL** (**T-SQL**) is [Microsoft](http://en.wikipedia.org/wiki/Microsoft)'s and [Sybase](http://en.wikipedia.org/wiki/Sybase)'s proprietary extension to [SQL](http://en.wikipedia.org/wiki/SQL). SQL, the acronym for Structured Query Language, is a [standardized](http://en.wikipedia.org/wiki/SQL#Standardization) computer language that was originally developed by IBM for querying, altering and defining relational databases, using [declarative](http://en.wikipedia.org/wiki/Declarative_programming) statements. T-SQL expands on the SQL standard to include [procedural](http://en.wikipedia.org/wiki/Procedural_programming) programming, [local variables](http://en.wikipedia.org/wiki/Local_variable), various support functions for string processing, date processing, mathematics, etc. and changes to the [DELETE](http://en.wikipedia.org/wiki/Delete_(SQL)) and [UPDATE](http://en.wikipedia.org/wiki/Update_(SQL)) statements. These additional features make Transact-SQL [Turing complete](http://en.wikipedia.org/wiki/Turing_complete)

**TASK 3. Start SQL Management Studio and connect to the database TelerikAcademy. Examine the major tables in the "TelerikAcademy" database.**

The database consists of Addresses, Departments, Employees, Projects, Towns and EmployeesProjects tables, all in a relationship to one another.

**TASK 4. Write a SQL query to find all information about all departments (use "TelerikAcademy" database).**

select \*

from Departments

**TASK 5. Write a SQL query to find all department names.**

select Name as [Department Names]

from Departments

**TASK 6. Write a SQL query to find the salary of each employee.**

select FirstName, LastName, Salary

from Employees

**TASK 7. Write a SQL to find the full name of each employee.**

select EmployeeId, FirstName + ' ' + LastName as [Employees Full Name]

from Employees

**TASK 8. 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 Employees

**TASK 9. Write a SQL query to find all different employee salaries.**

select distinct Salary

from Employees

**TASK 10. Write a SQL query to find all information about the employees whose job title is “Sales Representative“.**

select \*

from Employees

where JobTitle = 'Sales Representative'

**TASK 11. Write a SQL query to find the names of all employees whose first name starts with "SA".**

select FirstName, LastName

from Employees

where FirstName like 'SA%'

**TASK 12. Write a SQL query to find the names of all employees whose last name contains "ei".**

select FirstName, LastName

from Employees

where LastName like '%ei%'

**TASK 13. Write a SQL query to find the salary of all employees whose salary is in the range [20000…30000].**

select Salary

from Employees

where Salary between 20000 and 30000

**TASK 14. Write a SQL query to find the names of all employees whose salary is 25000, 14000, 12500 or 23600.**

select FirstName, Salary

from Employees

where Salary in (25000, 14000, 12500, 23600)

**TASK 15. Write a SQL query to find all employees that do not have manager.**

select FirstName, LastName

from Employees

where ManagerID is null

**TASK 16. Write a SQL query to find all employees that have salary more than 50000. Order them in decreasing order by salary.**

select FirstName, LastName, Salary

from Employees

where Salary > 50000

order by Salary desc

**TASK 17. Write a SQL query to find the top 5 best paid employees.**

select top 5 FirstName, LastName, Salary

from Employees

order by Salary desc

**TASK 18. Write a SQL query to find all employees along with their address. Use inner join with ON clause.**

select e.FirstName, e.LastName, a.AddressText as [Employee Addresses]

from Employees e

inner join Addresses a

on e.AddressID = a.AddressID

**TASK 19. Write a SQL query to find all employees and their address. Use equijoins (conditions in the WHERE clause).**

select e.FirstName, e.LastName,

a.AddressText as [Employee Adresses]

from Employees e, Addresses a

where e.AddressID = a.AddressID

**TASK 20. Write a SQL query to find all employees along with their manager.**

select e.FirstName, e.LastName, m.LastName as [Manager]

from Employees e, Employees m

where e.ManagerID = m.EmployeeID

**TASK 21. 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, m.LastName as [Managers], a.AddressText as [Address]

from Employees e

inner join Employees m

ON e.ManagerID = m.EmployeeID

inner join Addresses a

ON a.AddressID = e.AddressID

**TASK 22. 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

**TASK 23. 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, m.LastName as [Manager]

from Employees m

right outer join Employees e

on e.ManagerID = m.EmployeeID

select e.FirstName, e.LastName, m.LastName as [Manager]

from Employees e

left outer join Employees m

on e.ManagerID = m.EmployeeID

**TASK 24. 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 e.FirstName, e.LastName, d.Name as [Department Names]

from Employees e

inner join Departments d

on e.DepartmentID = d.DepartmentID

where (d.Name IN ('Sales', 'Finance')) and (YEAR(e.HireDate) between 1995 and 2005)