Advanced SQL

1. Write a SQL query to find the names and salaries of the employees that take the minimal salary in the company. Use a nested SELECT statement.

SELECT FirstName, Salary

FROM Employees

WHERE Salary =

(SELECT MIN(Salary) FROM Employees)

1. Write a SQL query to find the names and salaries of the employees that have a salary that is up to 10% higher than the minimal salary for the company.

SELECT FirstName, Salary

FROM Employees

WHERE Salary <

(SELECT 1.1 \* MIN(Salary) FROM Employees)

1. Write a SQL query to find the full name, salary and department of the employees that take the minimal salary in their department. Use a nested SELECT statement.

SELECT e.FirstName + ' ' + e.LastName AS [Full Name], e.Salary,

d.Name as [Department Name]

FROM Employees e

JOIN Departments d

ON e.DepartmentID = d.DepartmentID

WHERE e.Salary =

(SELECT MIN(Salary) FROM Employees)

1. Write a SQL query to find the average salary in the department #1.

SELECT AVG(Salary)

FROM Employees

WHERE DepartmentID = 1

1. Write a SQL query to find the average salary in the "Sales" department.

SELECT AVG(Salary)

FROM Employees e

JOIN Departments d

ON e.DepartmentID = d.DepartmentID

WHERE d.Name = 'Sales'

1. Write a SQL query to find the number of employees in the "Sales" department.

SELECT COUNT(\*)

FROM Employees e

JOIN Departments d

ON e.DepartmentID = d.DepartmentID

WHERE d.Name = 'Sales'

1. Write a SQL query to find the number of all employees that have manager.

SELECT COUNT(ManagerID)

FROM Employees

1. Write a SQL query to find the number of all employees that have no manager.

SELECT COUNT(\*) - COUNT(ManagerID) FROM Employees

1. Write a SQL query to find all departments and the average salary for each of them.

SELECT d.Name, AVG(e.Salary)

FROM Employees e

JOIN Departments d

ON e.DepartmentID = d.DepartmentID

GROUP BY d.Name

1. Write a SQL query to find the count of all employees in each department and for each town.

SELECT t.Name AS Town,

d.Name AS Department,

COUNT(\*) AS Employees

FROM Employees e

JOIN Departments d

ON e.DepartmentID = d.DepartmentID

JOIN Addresses a

ON e.AddressID = a.AddressID

JOIN Towns t

ON t.TownID = a.TownID

GROUP BY d.Name, t.Name

1. Write a SQL query to find all managers that have exactly 5 employees. Display their first name and last name.

SELECT m.FirstName, m.LastName

FROM Employees e

JOIN Employees m

ON e.ManagerID = m.EmployeeID

GROUP BY m.FirstName, m.LastName

HAVING COUNT(\*) = 5

1. Write a SQL query to find all employees along with their managers. For employees that do not have manager display the value "(no manager)".

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

ISNULL(m.FirstName, 'no') + ' ' + ISNULL(m.LastName, 'manager') AS Manager

FROM Employees e

FULL JOIN Employees m

ON e.ManagerID = m.EmployeeID

1. Write a SQL query to find the names of all employees whose last name is exactly 5 characters long. Use the built-in LEN(str) function.

SELECT LastName

FROM Employees

WHERE LEN(LastName) = 5

1. Write a SQL query to display the current date and time in the following format "day.month.year hour:minutes:seconds:milliseconds". Search in Google to find how to format dates in SQL Server.

SELECT FORMAT(GETDATE(), 'dd.MM.yyyy HH:mm:ss:fff');

1. Write a SQL statement to create a table Users. Users should have username, password, full name and last login time. Choose appropriate data types for the table fields. Define a primary key column with a primary key constraint. Define the primary key column as identity to facilitate inserting records. Define unique constraint to avoid repeating usernames. Define a check constraint to ensure the password is at least 5 characters long.
2. Write a SQL statement to create a view that displays the users from the Users table that have been in the system today. Test if the view works correctly.
3. Write a SQL statement to create a table Groups. Groups should have unique name (use unique constraint). Define primary key and identity column.
4. Write a SQL statement to add a column GroupID to the table Users. Fill some data in this new column and as well in the Groups table. Write a SQL statement to add a foreign key constraint between tables Users and Groups tables.
5. Write SQL statements to insert several records in the Users and Groups tables.
6. Write SQL statements to update some of the records in the Users and Groups tables.
7. Write SQL statements to delete some of the records from the Users and Groups tables.
8. Write SQL statements to insert in the Users table the names of all employees from the Employees table. Combine the first and last names as a full name. For username use the first letter of the first name + the last name (in lowercase). Use the same for the password, and NULL for last login time.
9. Write a SQL statement that changes the password to NULL for all users that have not been in the system since 10.03.2010.
10. Write a SQL statement that deletes all users without passwords (NULL password).
11. Write a SQL query to display the average employee salary by department and job title.
12. Write a SQL query to display the minimal employee salary by department and job title along with the name of some of the employees that take it.
13. Write a SQL query to display the town where maximal number of employees work.
14. Write a SQL query to display the number of managers from each town.
15. Write a SQL to create table WorkHours to store work reports for each employee (employee id, date, task, hours, comments). Don't forget to define identity, primary key and appropriate foreign key.

Issue few SQL statements to insert, update and delete of some data in the table.

Define a table WorkHoursLogs to track all changes in the WorkHours table with triggers. For each change keep the old record data, the new record data and the command (insert / update / delete).

1. Start a database transaction, delete all employees from the 'Sales' department along with all dependent records from the pother tables. At the end rollback the transaction.
2. Start a database transaction and drop the table EmployeesProjects. Now how you could restore back the lost table data?
3. Find how to use temporary tables in SQL Server. Using temporary tables backup all records from EmployeesProjects and restore them back after dropping and re-creating the table.