Tasks Description:  
  
https://softuni.bg/trainings/resources/officedocument/10272/data-aggregation-exercise-problem-descriptions-databases-basics-sqlserver-september-2016  
  
01.  
SELECT COUNT(\*) AS 'Count' FROM WizzardDeposits  
  
02.  
SELECT MAX(w.MagicWandSize) AS 'LongestMagicWand' FROM WizzardDeposits AS w  
  
03.  
SELECT w.DepositGroup, MAX(w.MagicWandSize) AS LongestMagicWand FROM WizzardDeposits AS w  
GROUP BY w.DepositGroup  
  
04.  
SELECT w.DepositGroup AS AverageMagicWand FROM WizzardDeposits AS w  
GROUP BY w.DepositGroup  
HAVING AVG(w.MagicWandSize) IN (  
SELECT TOP 1 AVG(w.MagicWandSize) AS AverageMagicWand FROM WizzardDeposits AS w  
GROUP BY w.DepositGroup  
ORDER BY AverageMagicWand)  
  
05.  
SELECT w.DepositGroup, SUM(w.DepositAmount) AS TotalSum FROM WizzardDeposits AS w  
GROUP BY w.DepositGroup  
  
06.  
SELECT w.DepositGroup, SUM(w.DepositAmount) AS TotalSum FROM WizzardDeposits AS w  
WHERE w.MagicWandCreator = 'Ollivander family'  
GROUP BY w.DepositGroup  
  
07.  
SELECT w.DepositGroup, SUM(w.DepositAmount) AS TotalSum FROM WizzardDeposits AS w  
WHERE w.MagicWandCreator = 'Ollivander family'  
GROUP BY w.DepositGroup  
HAVING SUM(w.DepositAmount) < 150000  
ORDER BY TotalSum DESC  
  
08.  
SELECT w.DepositGroup, w.MagicWandCreator, MIN(w.DepositCharge) AS MinDepositCharge FROM WizzardDeposits AS w  
GROUP BY w.DepositGroup, w.MagicWandCreator  
ORDER BY w.MagicWandCreator, w.DepositGroup  
  
09.  
SELECT CASE

WHEN w.Age BETWEEN 0 AND 10 THEN '[0-10]'

WHEN w.Age BETWEEN 11 AND 20 THEN '[11-20]'

WHEN w.Age BETWEEN 21 AND 30 THEN '[21-30]'

WHEN w.Age BETWEEN 31 AND 40 THEN '[31-40]'

WHEN w.Age BETWEEN 41 AND 50 THEN '[41-50]'

WHEN w.Age BETWEEN 51 AND 60 THEN '[51-60]'

WHEN w.Age > 60 THEN '[61+]'

END AS AgeGroup,

COUNT(\*)

FROM WizzardDeposits AS w

GROUP BY CASE

WHEN w.Age BETWEEN 0 AND 10 THEN '[0-10]'

WHEN w.Age BETWEEN 11 AND 20 THEN '[11-20]'

WHEN w.Age BETWEEN 21 AND 30 THEN '[21-30]'

WHEN w.Age BETWEEN 31 AND 40 THEN '[31-40]'

WHEN w.Age BETWEEN 41 AND 50 THEN '[41-50]'

WHEN w.Age BETWEEN 51 AND 60 THEN '[51-60]'

WHEN w.Age > 60 THEN '[61+]'

END  
  
10.  
SELECT LEFT(w.FirstName, 1) AS 'first\_letter' FROM WizzardDeposits AS w  
WHERE w.DepositGroup = 'Troll Chest'  
GROUP BY LEFT(w.FirstName, 1)  
ORDER BY LEFT(w.FirstName, 1)  
  
11.  
SELECT w.DepositGroup, w.IsDepositExpired, AVG(w.DepositInterest) FROM WizzardDeposits AS w  
WHERE w.DepositStartDate > CAST('01/01/1985'AS DATE)  
GROUP BY w.DepositGroup, w.IsDepositExpired  
ORDER BY w.DepositGroup DESC, w.IsDepositExpired  
  
12.  
SELECT SUM(XX.DIFF)  
FROM (SELECT w.DepositAmount -   
(SELECT A.DepositAmount FROM WizzardDeposits AS A   
WHERE A.Id = w.Id + 1)   
AS DIFF FROM WizzardDeposits AS w) AS XX  
  
14.  
SELECT e.DepartmentID, MIN(e.Salary) AS MinimumSalary FROM Employees AS e  
WHERE e.DepartmentID IN (2,5,7) AND e.HireDate > CAST('01/01/2000' AS smalldatetime)  
GROUP BY e.DepartmentID  
  
15.  
SELECT \*INTO NewTable  
FROM Employees AS e   
WHERE e.Salary > 30000

DELETE  
FROM [dbo].[NewTable]  
Where ManagerID = 42  
UPDATE NewTable  
SET Salary = Salary + 5000  
WHERE DepartmentID=1

SELECT DepartmentID, AVG(Salary) AS 'DepartmentID' FROM NewTable AS n  
GROUP BY n.DepartmentID  
  
16.  
SELECT e.DepartmentID, MAX(e.Salary) AS MaxSalary FROM Employees AS e   
GROUP BY e.DepartmentID  
HAVING MAX(e.Salary) < 30000 OR MAX(e.Salary) > 70000  
  
17.  
SELECT COUNT(e.Salary) AS Count FROM Employees AS e   
WHERE e.ManagerID IS NULL  
  
18.  
SELECT DISTINCT sal.DepartmentID, sal.Salary FROM

(SELECT e.DepartmentID, e.Salary, DENSE\_RANK() OVER (PARTITION BY e.DepartmentID ORDER BY e.Salary DESC) AS SalaryRank FROM Employees AS e) AS sal

WHERE sal.SalaryRank = 3  
  
19.  
SELECT TOP 10 u.FirstName, u.LastName, u.DepartmentID FROM Employees as u

JOIN

(SELECT e.DepartmentID, AVG(e.Salary) as AverageSalary FROM Employees AS e GROUP BY e.DepartmentID) AS a  
  
on a.DepartmentID = u.DepartmentID  
WHERE u.Salary > a.AverageSalary

ORDER BY a.DepartmentID