**Mock test - possible solutions**

1. Count how many different JobTitles exist among employees. (Consider using: HumanResources.Employee) [5 marks]

Select count (distinct jobtitle) from HumanResources.Employee

1. Retrieve all information from Product table regarding all products not containing Metal in their names. (Consider using: Production.Product).

Select \* from Production.Product

Where name not like '%metal%'

1. Show JobTitle, HireDate, FirstName, LastName of those employees who were hired before first of January 2002. Order records by HireDate. (Consider using: HumanResources.Employee, person.Person)

Select jobtitle, HireDate, pp.FirstName, pp.LastName from HumanResources.Employee he

join person.Person pp on he.BusinessEntityID=pp.BusinessEntityID

where hiredate < '2002-01-01'

order by HireDate

1. Display all product names and corresponding culture names. (Consider using: Production.Product, production.ProductModelProductDescriptionCulture, Production.Culture)

Select pp.Name as product, pc.Name as model

from Production.Product pp

join production.ProductModelProductDescriptionCulture pm on pp.ProductModelID=pm.ProductModelID

join Production.Culture pc on pm.CultureID=pc.CultureID

1. Show JobTitle, BirthDate, Age of Employee, FirstName, LastName of those Employees who are older than 60. Order records by years. (Consider using: HumanResources.Employee, person.Person, DATEDIFF() function)

Select jobtitle, BirthDate, DATEDIFF(YEAR, BirthDate, GETDATE()) as Age, pp.FirstName, pp.LastName from HumanResources.Employee he

join person.Person pp on he.BusinessEntityID=pp.BusinessEntityID

where DATEDIFF(YEAR, BirthDate, GETDATE())>60

order by Age

1. Show Rates of Employees’ salary and corresponding JobTitles when Rate is lower than average rate. (Consider using: HumanResources.EmployeePayHistory, HumanResources.Employee)

select e.JobTitle, rate

from HumanResources.EmployeePayHistory eph join HumanResources.Employee e

on eph.BusinessEntityID=e.BusinessEntityID

where rate<(select avg(eph.Rate) from HumanResources.EmployeePayHistory eph)

1. Show the number of products per subcategory sorted in descending order only for those subcategories, which have more than 20 products. (Consider using: Production.Product, Production.ProductSubcategory)

Select count(pp.productID), ps.Name

from Production.Product pp join Production.ProductSubcategory ps

on pp.ProductSubcategoryID=ps.ProductSubcategoryID

group by ps.Name

having count(pp.productID)>20

order by count(pp.productID) desc

1. Create views on any two queries and comment whether they are updatable or not.

**create view products\_per\_subcat\_view as**

Select pp.FirstName, pp.LastName, rate, min(rate) over (partition by hd.name) as [minimum in its department], rate-min(rate) over (partition by hd.name) as difference from Person.Person pp

join HumanResources.EmployeePayHistory hep

on pp.BusinessEntityID=hep.BusinessEntityID

join HumanResources.EmployeeDepartmentHistory hed

on hed.BusinessEntityID=pp.BusinessEntityID

join HumanResources.Department hd

on hd.DepartmentID=hed.DepartmentID

**create view product\_names\_view as**

Select pp.Name as product, pc.Name as model

from Production.Product pp

join production.ProductModelProductDescriptionCulture pm

on pp.ProductModelID=pm.ProductModelID

join Production.Culture pc on pm.CultureID=pc.CultureID

1. Show list of employees ordered by department, employee rate and minimum rate for department this employee works at. For each employee show the difference between lowest rate in the department where employee works and his own salary. (Consider using: Person.Person, HumanResources.EmployeePayHistory, HumanResources.EmployeeDepartmentHistory, HumanResources.Department)

Select pp.FirstName, pp.LastName, rate, min(rate) over (partition by hd.name) as [minimum in its department], rate - min(rate) over (partition by hd.name) as difference

from Person.Person pp

join HumanResources.EmployeePayHistory hep

on pp.BusinessEntityID=hep.BusinessEntityID

join HumanResources.EmployeeDepartmentHistory hed

on hed.BusinessEntityID=pp.BusinessEntityID

join HumanResources.Department hd

on hd.DepartmentID=hed.DepartmentID

order by hd.Name, rate

1. Retrieve the BusinessEntityID, LoginID, JobTitle for those employees who are job candidates as well. (Use tables: HumanResources.Employee, HumanResources.JobCandidate)

WITH

cteCandidates (BusinessEntityID)

AS

(

SELECT BusinessEntityID

FROM HumanResources.Employee

INTERSECT

SELECT BusinessEntityID

FROM HumanResources.JobCandidate

)

SELECT

c.BusinessEntityID,

e.LoginID,

e.JobTitle

FROM

HumanResources.Employee AS e

INNER JOIN cteCandidates AS c

ON e.BusinessEntityID = c.BusinessEntityID

ORDER BY

c.BusinessEntityID;