**Mock test - possible solutions**

**1\_ Count how many different jobtitles exist among employees.. [5 marks]**

Select count (distinct jobtitle)

from HumanResources.Employee

**2\_ Retrieve all information from Product table regarding all products not containing Metal in their names.**

Select \* from Production.Product

Where name not like '%metal%'

**3\_ Show JobTitle, HireDate, FirstName, LastName of those employees hired before first of January 2002. Order records by HireDate.**

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

**4\_ Display all product names and corresponding culture names.**

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

**5\_ Show JobTitle, BirthDate, Age of Employee, FirstName, LastName of those Employees who are older than 60. Order records by years.**

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

**6\_ Show Rates of Employees’ salary and corresponding JobTitle when Rate is lower than average rate.**

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)

**7\_ Show the number of products per subcategory sorted in descending order only for those subcategories, which have more than 20 products.**

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

**8\_ 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

**9\_ 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.**

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

**10\_ Retrieve the BusinessEntityID, LoginID, JobTitle for those employees who are job candidates as well. You should use INTERSECT operator to combine the two queries and CTE.**

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;

**1. For each employee display job title and date when employee was hired. (use table**

**HumanResources.Employee) [8 marks]**

SELECT JobTitle, HireDate "date hired"

from HumanResources.Employee

order by HireDate desc

**2. Show all the information about products which have standard cost between 90 and 100. Sort**

**results by standard cost in descending order. (Use table Production.Product) [10 marks]**

select \*

from Production.Product

where StandardCost between 90 and 100

order by StandardCost desc

**3. Show products that have name containing word “Mountain”, and price between 500 and 800. In**

**the output display product name, sell start date, subcategory name, price, and color. Provide**

**meaningful names for the columns in the results. Sort results by color in descending order and**

**then by product name in ascending order. (Use tables production.Product and**

**production.ProductSubcategory) [14 marks]**

select p.Name "Product name", p.SellStartDate "Date on sail", c.Name "Suncategory name",

p.ListPrice, p.Color "Product color"

from production.Product p

join production.ProductSubcategory c on p.ProductSubcategoryID =

c.ProductSubcategoryID

where c.Name like '%Mountain%' and p.ListPrice between 600 and 900

order by p.Color desc , c.Name asc

**4. For each combination of marital status and gender show average sick leave hours of employees**

**and number of employees. Filter out employees who was born before '1980-10-10'. In the results**

**show only those records where average sick leave hours is more than 43. Provide meaningful**

**names for the columns in the results. Sort results by average sick leave hours in ascending order.**

**(Use table HumanResources.Employee) [15 marks]**

select MaritalStatus, Gender, count(\*) "Num of employees", avg(SickLeaveHours) "Avg seak leave hours"

from HumanResources.Employee

where BirthDate > '1980-10-10'

group by MaritalStatus, Gender

having avg(SickLeaveHours) > 43

order by "Avg seak leave hours" asc

**5. Show all sales persons who has bonus more than 100 and last year sales more than the average**

**last year sales among all sales persons. In the results display sales persons first and last names,**

**, months since ModifiedDate of the sales person till now, and average sales last year among all**

**sales persons. Provide meaningful names for the columns in the results. Sort results by the last**

**name in descending order. (Use tables Sales.SalesPerson and Person.Person) [18 marks]**

select p.FirstName, p.LastName, DATEDIFF(month, sp.ModifiedDate, getdate()) "Months till now", (select avg(SalesLastYear) from Sales.SalesPerson) "Avg sales last year"

from Sales.SalesPerson sp join [Person].[Person] p on sp.BusinessEntityID = p.BusinessEntityID

where sp.Bonus > 100 and sp.SalesLastYear > (select avg(SalesLastYear) from Sales.SalesPerson)

order by p.lastname desc

6. Create one view for any SQL query. State if the view is updatable or not, explain your decision.

[10 marks]

**7. Show business entities where store name contains word “Toy”. For each business entity display**

**entity ModifiedDate, business contact type name, store name, and full business entity address in**

**the format: “AddressLine1, PostalCode” (notice, separated by comma), e.g. “252851 Rowan**

**Place, V6B 3P7”. Provide meaningful names for the columns in the results. Sort results by contact**

**type name in ascending order, then by store name in descending order, and then by entity**

**ModifiedDate in ascending order. (Use tables Person.BusinessEntity,**

**person.BusinessEntityContact, person.ContactType, person.BusinessEntityAddress,**

**person.Address, Sales.Store). [25 marks]**

select e.ModifiedDate "Business entity date modified",ct.Name "Business contact type name" ,concat (a.AddressLine1, ', ', a.PostalCode) "Full address" ,s.Name "Store name"

from Person.BusinessEntity e

join person.BusinessEntityContact ec on ec.BusinessEntityID = e.BusinessEntityID

join person.ContactType ct on ct.ContactTypeID = ec.ContactTypeID

join person.BusinessEntityAddress ea on ea.BusinessEntityID = e.BusinessEntityID

join person.Address a on a.AddressID = ea.AddressID

join Sales.Store s on s.BusinessEntityID = e.BusinessEntityID

where s.Name like '%Toy%'

order by ct.Name asc, s.Name desc, e.ModifiedDate asc