

SELECT Dnumber, Dname, COUNT(E.Ssn) AS NumOfEmployee

FROM (

SELECT Dnumber, Dname

FROM DEPARTMENT AS D

JOIN EMPLOYEE AS E ON E.Dno = D.Dnumber

GROUP BY Dnumber, Dname

HAVING COUNT(E.Ssn) > 5

)D

JOIN (

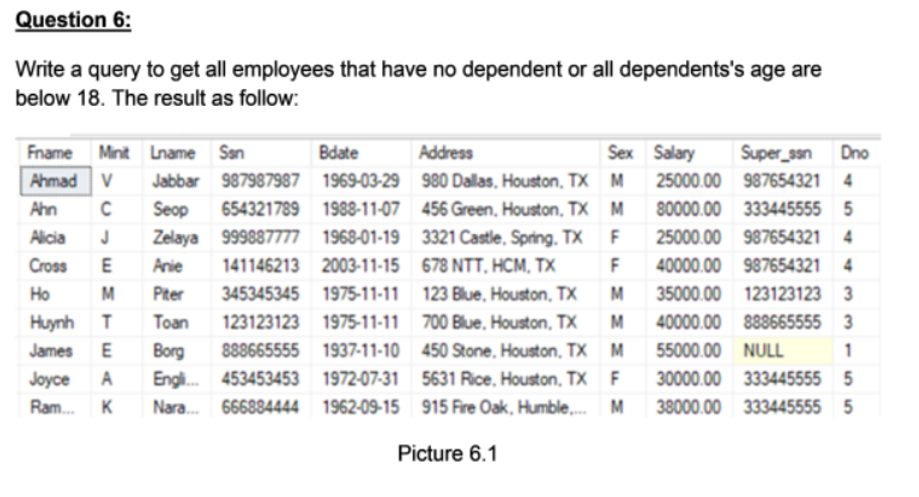
SELECT \* FROM EMPLOYEE AS E

WHERE Salary > 40000

) E ON E.Dno = D.Dnumber

GROUP BY D.Dnumber, D.Dname

ORDER BY NumOfEmployee DESC



SELECT \*

FROM EMPLOYEE

WHERE Ssn NOT IN (

SELECT Ssn FROM EMPLOYEE

WHERE Ssn IN (

SELECT Ssn FROM EMPLOYEE

JOIN DEPENDENT ON DEPENDENT.Essn = EMPLOYEE.Ssn

)

)

OR Ssn NOT IN (

SELECT Ssn FROM EMPLOYEE

JOIN DEPENDENT ON DEPENDENT.Essn = EMPLOYEE.Ssn

WHERE Ssn IN (

SELECT Ssn FROM EMPLOYEE

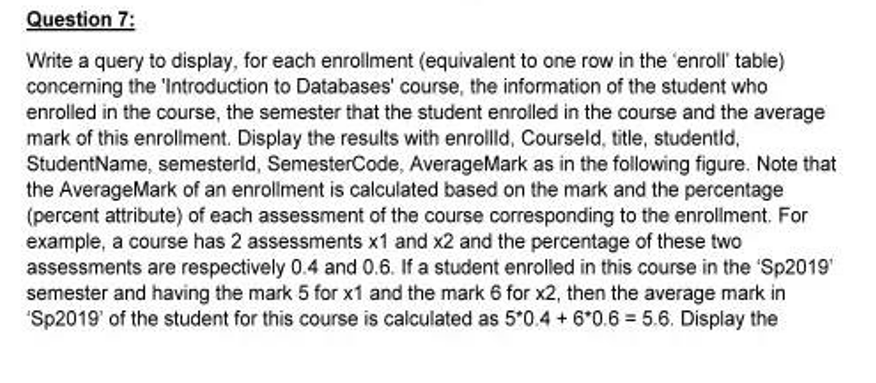
JOIN DEPENDENT ON DEPENDENT.Essn = EMPLOYEE.Ssn

WHERE YEAR(GETDATE()) - YEAR(DEPENDENT.Bdate) >= 18

)

)

ORDER BY Fname ASC



SELECT E.enrollId, C.id AS CourseId, C.title, E.studentId,

ST.name AS StudentName, E.semesterId, SM.code AS SemesterCode,

ROUND(SUM(DISTINCT M.mark \* A.[percent]) / SUM(A.[percent]), 3) AS AverageMark

FROM enroll AS E

JOIN marks AS M ON E.enrollId = M.enrollId

JOIN Students AS ST ON E.studentId = ST.id

JOIN semesters AS SM ON E.semesterId = SM.id

JOIN Courses AS C ON E.courseId = C.id

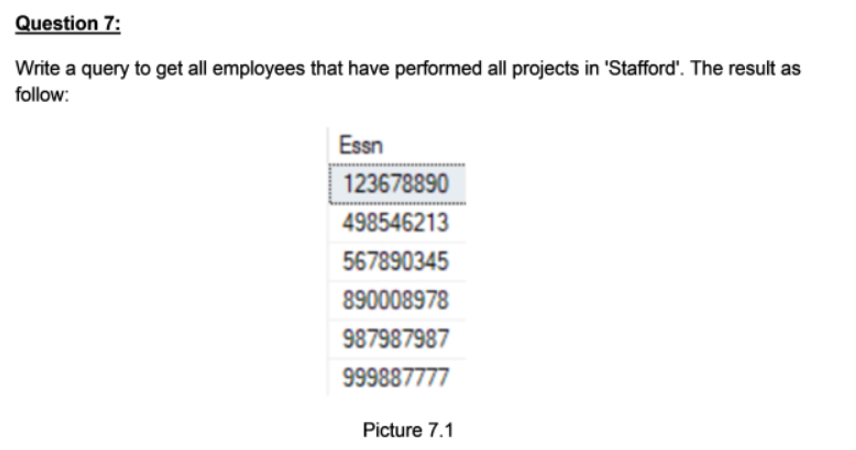
JOIN Assessments AS A ON M.assessmentId = A.id

WHERE C.title = 'Introduction to Databases'

GROUP BY E.enrollId, C.id, C.title, E.studentId,

ST.name, E.semesterId, SM.code

ORDER BY E.studentId ASC, E.semesterId DESC



SELECT \*

FROM EMPLOYEE E

WHERE NOT EXISTS ( --lọc những nhân viên chưa từng làm Stafford

SELECT P.Pnumber

FROM PROJECT P

WHERE P.Plocation = 'Stafford'

EXCEPT

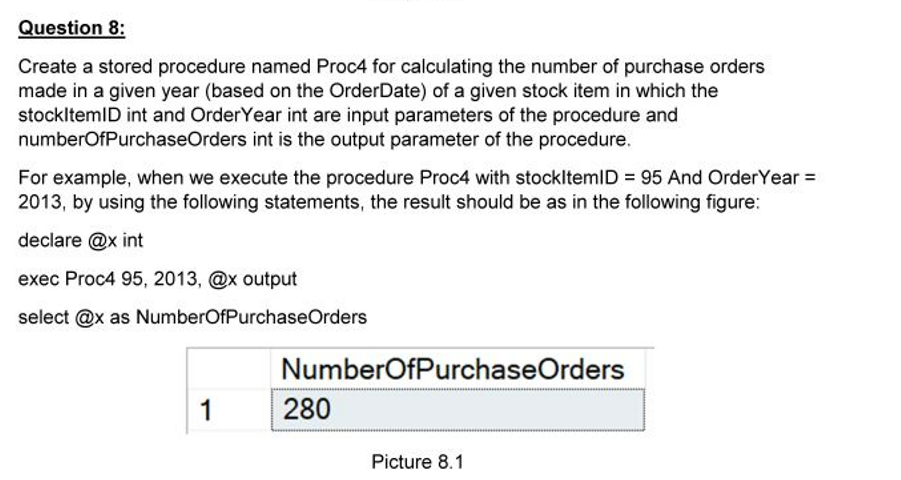
SELECT W.Pno

FROM WORKS\_ON W

WHERE W.Essn = E.Ssn

);

PROC



CREATE PROC Proc4 @stockItemID INT,

@OrderYear INT,

@numberOfPurchaseOrders INT OUTPUT

AS

BEGIN

SELECT @numberOfPurchaseOrders = COUNT(PO.PurchaseOrderID)

FROM PurchaseOrders AS PO

JOIN PurchaseOrderLines AS POL ON PO.PurchaseOrderID = POL.PurchaseOrderID

WHERE POL.StockItemID = @stockItemID AND

YEAR(PO.OrderDate) = @OrderYear

GROUP BY PO.SupplierID

END

--TÍNH SỐ LƯỢNG HÓA ĐƠN CỦA MÃ CHỨNG KHOÁN CO StockID = ?

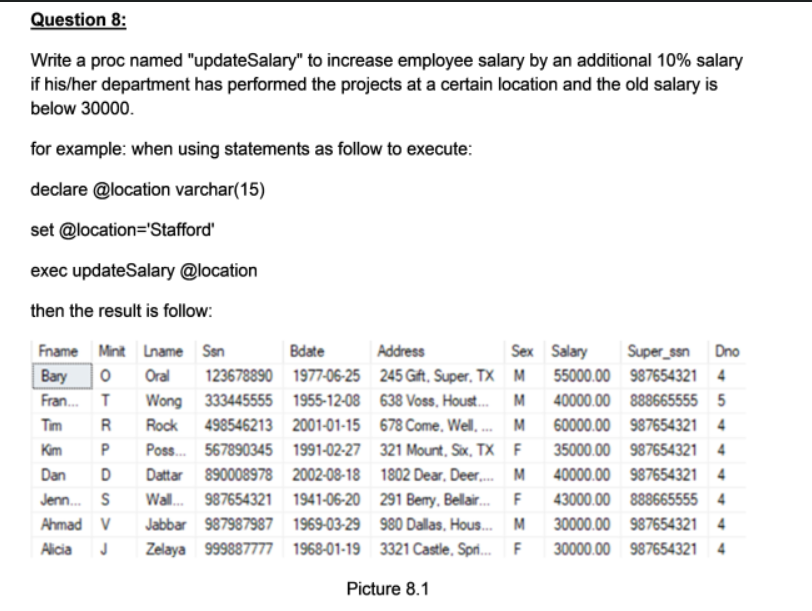
SELECT COUNT(PO.PurchaseOrderID) AS TOTAL

FROM PurchaseOrders AS PO

JOIN PurchaseOrderLines AS POL ON PO.PurchaseOrderID = POL.PurchaseOrderID

WHERE POL.StockItemID = 95 AND YEAR(PO.OrderDate) = 2013

GROUP BY PO.SupplierID



CREATE PROCEDURE updateSalary

@InputLocation VARCHAR(15)

AS

BEGIN

UPDATE EMPLOYEE

SET Salary = Salary \*1.1

WHERE Salary <30000

AND Ssn IN (

SELECT Ssn FROM EMPLOYEE

JOIN WORKS\_ON ON WORKS\_ON.Essn = Ssn

JOIN PROJECT ON PROJECT.Pnumber = WORKS\_ON.Pno

WHERE PROJECT.Plocation = @InputLocation

)

SELECT \* FROM EMPLOYEE

WHERE Ssn

IN (

SELECT Ssn FROM EMPLOYEE

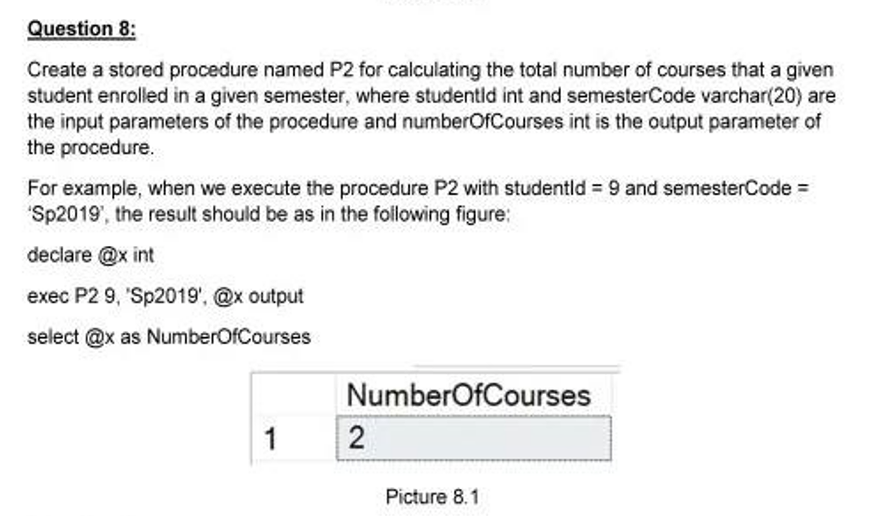
JOIN WORKS\_ON ON WORKS\_ON.Essn = Ssn

JOIN PROJECT ON PROJECT.Pnumber = WORKS\_ON.Pno

WHERE PROJECT.Plocation = @InputLocation

)

END



CREATE PROC P2 @studentId INT,

@semesterCode VARCHAR(20),

@numberOfCourses INT OUTPUT

AS

BEGIN

SELECT @numberOfCourses = COUNT(ST.id)

FROM enroll AS E

JOIN Students AS ST ON E.studentId = ST.id

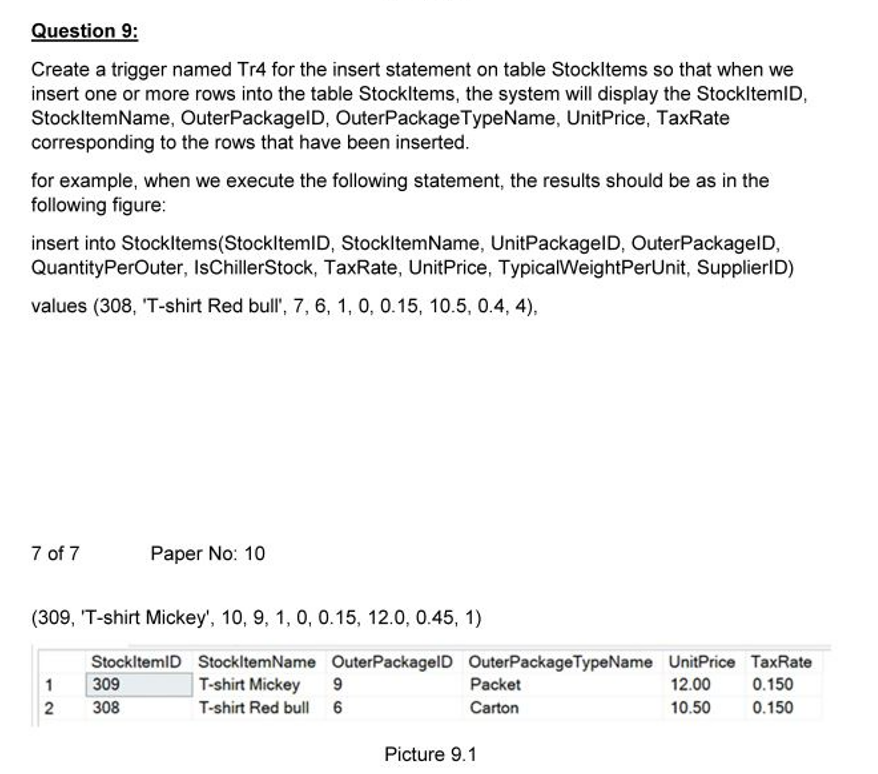
JOIN semesters AS SM ON E.semesterId = SM.id

WHERE SM.code = @semesterCode AND ST.id = @studentId

GROUP BY ST.id

END

TRIGGER



CREATE TRIGGER Tr4

ON StockItems

AFTER INSERT

AS

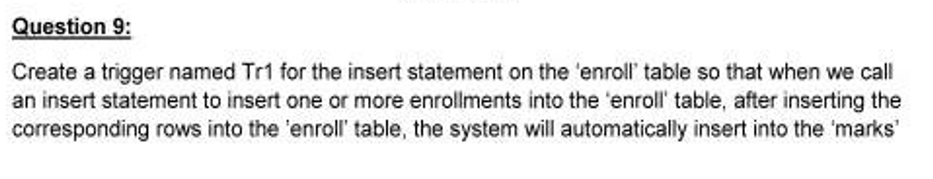
BEGIN

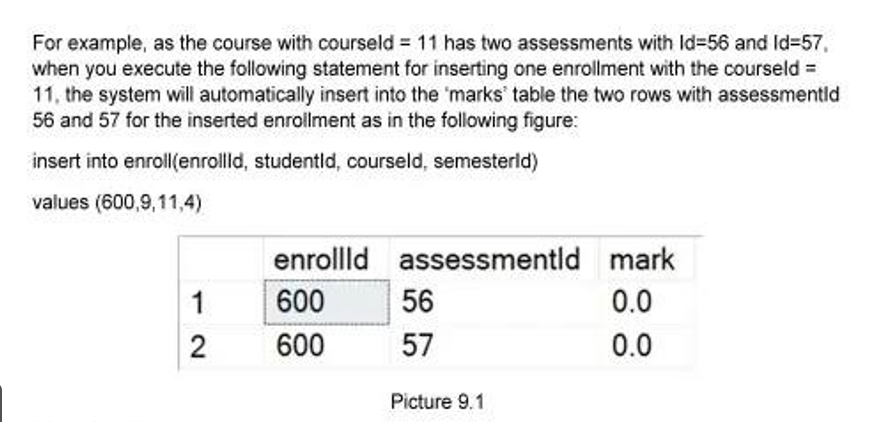
SELECT I.StockItemID, I.StockItemName, I.OuterPackageID, PT.PackageTypeName AS OuterPackageTypeName, I.UnitPrice, I.TaxRate

FROM inserted AS I

JOIN PackageTypes AS PT ON I.OuterPackageID = PT.PackageTypeID

END





CREATE TRIGGER Tr1

ON enroll

AFTER INSERT

AS

BEGIN

INSERT INTO marks(enrollId, assessmentId, mark)

SELECT I.enrollId, A.id AS assessmentId, 0

FROM inserted AS I

JOIN Assessments AS A ON I.courseId = A.courseId

DECLARE @enrollId INT

SELECT @enrollId = enrollId FROM inserted

SELECT \*

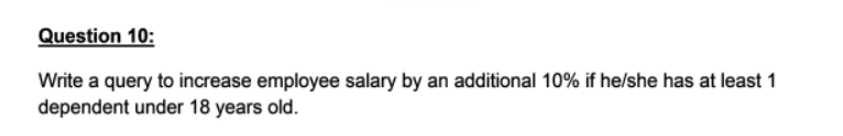
FROM marks AS M

WHERE M.enrollId = @enrollId

END

FUNCTION

UPDATE



UPDATE EMPLOYEE

SET Salary = Salary \* 1.10

WHERE Ssn IN (

SELECT E.Ssn

FROM EMPLOYEE E

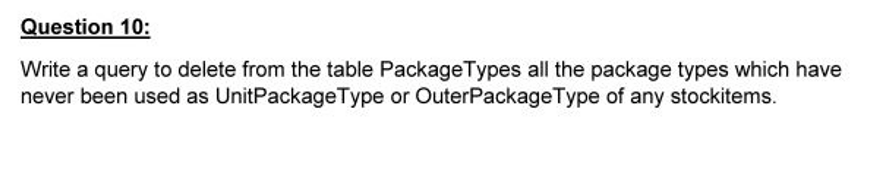
JOIN DEPENDENT D ON E.Ssn = D.Essn

WHERE YEAR(GETDATE()) - YEAR(D.Bdate) < 18

)

INSERT

DELETE



DELETE FROM PackageTypes

WHERE PackageTypeID NOT IN (

SELECT UnitPackageID

FROM StockItems

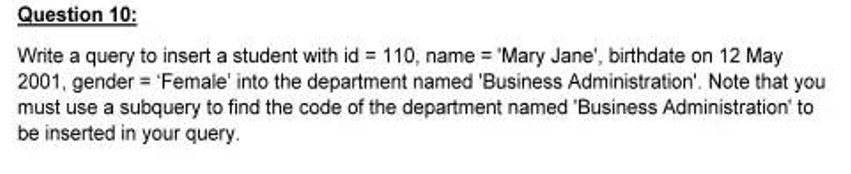
)

AND PackageTypeID NOT IN (

SELECT OuterPackageID

FROM StockItems

)



DECLARE @departmentCode VARCHAR(50)

SELECT @departmentCode = D.Code

FROM Departments AS D

WHERE D.Name = 'Business Administration'

INSERT INTO Students(id, name, birthdate, gender, department)

VALUES(110, 'Mary Jane', '2001-05-12', 'Female', @departmentCode)