



(University of Choice)

MASINDE MULIRO UNIVERSITY OF SCIENCE AND TECHNOLOGY

(MMUST)

MAIN CAMPUS

UNIVERSITY EXAMINATIONS

2021 / 2022 ACADEMIC YEAR

**THIRD YEAR SECOND SEMESTER EXAMINATIONS
FOR THE DEGREE OF BACHELOR**

OF

BIT

COURSE CODE: BIT 327

COURSE TITLE: DATABASE PROGRAMMING

DATE: Friday 22/04/2022

TIME: 12:00-2:00p.m

INSTRUCTIONS TO CANDIDATES

Question ONE (1) is compulsory
Attempt any TWO (2) questions

TIME: 2 Hours

MMUST observes ZERO tolerance to examination cheating

This Paper Consists of 4 Printed Pages. Please Turn Over.

QUESTION ONE [24 MARKS]

- a) Write the SQL CREATE TABLE statement for the **owns** relation between **Skier** and **PairOfSkis**. Make sure that your statement specifies the PRIMARY KEY and any FOREIGN KEYS. Additionally, we would like to enforce the constraint that purchase price be greater than zero. [8 Marks]
- b) Explain why it is not always possible to perform SQL UPDATE/DELETE/INSERT statements on top of a view. [6 Marks]
- c) Write an SQL trigger to carry out the following action: On **delete** of an account, for each owner of the account, check if the owner has any remaining accounts, and if she does not, delete her from the *depositor* relation. [8 Marks]

QUESTION TWO [18 Marks]

Refer to the program listing and Tables (Fig 2.1). This listing represents pseudocode for a trigger that operates on the Tables Parts and ReorderStock

Fig 2.1 Trigger Code and Sample Tables

```
CREATE TRIGGER st_order_trg AFTER
UPDATE OF qty_in_stock ON parts
WHEN (NEW.qty_in_stock <
NEW.reorder_level)
FOR EACH ROW
DECLARE
  x NUMBER;
  y NUMBER;
BEGIN
  SELECT COUNT(*) INTO x
  FROM reorderstock
  WHERE partid = :NEW.partid;
  IF x = 0 THEN
    SELECT reorderqty INTO y
    FROM parts
    WHERE partid = :NEW.partid;
    INSERT INTO reorderstock
    VALUES (:NEW.partid, SYSDATE, y)
  END IF;
END;
```

Table: Parts

PartID	StockDetail	Qty_In_Stock	Reorder_level	Reorder_Qty
1	Drill	450	500	50
2	Hammer	50	60	20
3	Mallet	29	20	5
4	Chisel	149	150	20

Table: ReorderStock

PartID	OrderDate	QtyOrdered
--------	-----------	------------

1	08-12-21	50
3	12-12-21	10

From examination of the trigger code explain:-

- i. What integrity constraint is enforced? **[5 Marks]**
- ii. How the trigger works, giving examples, using the sample Tables, of the circumstances it would execute and enforce the integrity constraint. **[7 Marks]**
- iii. How removing the WHEN clause and performing the test within the trigger code instead, affects the way in which the trigger works. **[6 Marks]**

QUESTION THREE [18 Marks]

- a) Construct a stored procedure, named `usp_GetLastName`, that accepts one input parameter named `EmployeeID` and returns (using a `SELECT Transact-SQL` statement) the last name of the employee. **[6 Marks]**
- b) Explain two major problems with processing update operations expressed in terms of views **[6 Marks]**
- c) Suppose that there is a database system that never fails. Is a recovery manager required for this system, if no why if yes how? **[6 Marks]**

QUESTION FOUR

- a) Given the relations below, answer the following questions
`Employee(EmpId, Emp_Name, Address, CountyId)`
`County(CountyId, County_name)`
 - i. Given that there are 47 counties in kenya, add a **default Constraints** that will display a default value if county is unknown, with the `CountyId` assigned to 48 when the employee county is unknown. **[6 Marks]**
 - ii. What happens when insert commands the `CountyId` in `Employee Null` is passed? Explain your answer **[6 Marks]**
- b) The SQL script below is to create and execute the parameter stored procedure.
What will be the returned when the stored procedure is executed? Explain your answer **[6 Marks]**

```

spGetEmployeeByGenderAndDepartement 1, 'Male'
Create procedure spGetEmployeeByGenderAndDepartement
@Gender nvarchar(20),

```

```

@DepartmentId int
as
Begin
    Select Name, Gender, Department from employeetble where Gender = @Gender
    and DepartmentId= @DepartmentId
End

```

QUESTION FIVE [18 Marks]

Given the table employee with the following attributes

Employeetbl(EmpId, Emp_Name, Gender, Salary)

Create a **after update trigger** that should display the following in EmployeeAuditData table

[18 Marks]

EmpId	AuditData
4	Employee with EmpId = 4 changed Name from Matoke to Ndizi, salary from 199000 to 234000

Note: The After trigger for UPDATE event, makes use of both **inserted** and **deleted** tables. The **inserted** table contains the updated data and the **deleted** table contains the old data

QUESTION SIX [18 Marks]

Write an SQL query, without using a **with** clause, to find all branches where the total account deposit is less than the average total account deposit at all branches,

- i. Using a nested query in the **from** clause.
- ii. Using a nested query in a **having** clause.

[10

Marks]

[8 Marks]