



**NAMIBIA UNIVERSITY
OF SCIENCE AND TECHNOLOGY**

**FACULTY OF COMPUTING AND INFORMATICS
SCHOOL OF COMPUTING**

DEPARTMENT OF SOFTWARE ENGINEERING

QUALIFICATIONS: BACHELOR OF COMPUTER SCIENCE; BACHELOR OF INFORMATICS; BACHELOR OF GEOINFORMATION TECHNOLOGY	
QUALIFICATION CODES: 07BCMS; 07BACS 07BAIT; 07BAIF; 07GITB	LEVEL: 6
COURSE CODE: DPG621S	COURSE NAME: DATABASE PROGRAMMING
SESSION: NOVEMBER 2023	PAPER: (PAPER 1)
DURATION: 3 HOURS	MARKS: 100

FIRST OPPORTUNITY EXAMINATION QUESTION PAPER	
EXAMINER:	DR GEREON KOCH KAPUIRE
MODERATOR:	MS ROSETHA KAYS

INSTRUCTIONS	
<ol style="list-style-type: none">1. Answer ALL the questions.2. Read all the questions carefully before answering.3. Number the answers clearly	

THIS QUESTION PAPER CONSISTS OF 5 PAGES (Including this front page)

Section A – Theory (40 Marks)

Question 1

Choose True or False

[10 Marks]

No	Questions	True (T)	False (F)
1	The CASE statement in SQL Server can only be used in SELECT and WHERE clauses.		
2	Temporary tables created within a stored procedure are automatically dropped when the procedure finishes execution.		
3	Transactions cannot be explicitly started and committed within a stored procedure.		
4	Input parameters in SQL Server stored procedures can be modified within the procedure.		
5	The order in which parameters are defined in a SQL Server stored procedure must match the order in which they are passed during the procedure call.		
6	The EXECUTE statement in SQL Server is used to invoke a stored procedure.		
7	Triggers can be used to audit changes to a table by logging information about INSERT, UPDATE, and DELETE operations.		
8	Parameters in SQL Server stored procedures can have default values.		
9	Cursors in SQL Server are used to iterate through a result set row by row.		
10	SQL Server supports a traditional FOR loop similar to those found in procedural programming languages.		

Question 2

[6 Marks]

Compare the usage of stored procedures and triggers in SQL Server. When would you prefer one over the other?

Question 3

[4 Marks]

What is the fundamental difference between a BEFORE trigger and an AFTER trigger in SQL Server?

Question 4

[4 Marks]

Compare local variables and parameters in terms of scope and usage within stored procedures.

Question 5

[4 Marks]

Discuss the differences between IF statements and the TRY...CATCH construct in SQL Server.

Question 6

[4 Marks]

Compare the use of a cursor and a regular SELECT statement for iterating over a result set.

Question 7

[4 Marks]

Explain the difference between a committed transaction and a rolled-back transaction in SQL Server.

Question 8

[4 Marks]

Discuss the benefits of using the SAVEPOINT statement within a transaction in SQL Server.

Section B – Practical (60 Marks)

(Use SQL Server Management System (SSMS) on your computer for this section)

Instructions: Copy and Paste your final code onto notepad and save it on the exam drive (z). The drive is the drive with your student number.

Question 1

[4 Marks]

Scenario 1: Basic Parameterized Stored Procedure

```
CREATE TABLE Employees (  
    EmployeeID INT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    Salary INT,  
    JobTitle VARCHAR(50)  
);  
INSERT INTO Employees (EmployeeID, FirstName, LastName, Salary, JobTitle)  
VALUES  
    (1, 'John', 'Doe', 65000, 'Manager'),  
    (2, 'Jane', 'Smith', 55000, 'Developer'),  
    (3, 'Bob', 'Johnson', 70000, 'Manager'),  
;
```

Create a stored procedure that retrieves employees with a salary greater than a specified amount.

Question 2

[4 Marks]

Scenario 2: Stored Procedure

Write a SQL query using the GetEmployeesBySalaryAndTitle stored procedure to retrieve all employees with a salary greater than \$60,000 and a job title of 'Manager'. Ensure that the query handles cases where the job title is not specified. Provide the necessary SQL code for this query. Use the table created in question 1 (scenario 1).

Question 3

[3 Marks]

Scenario 3: Handling Default Parameter Values

Make the job title parameter optional by setting a default value. Update the stored procedure created in Question 2 accordingly.

Question 4

[10 Marks]

Scenario 4: User Authentication

```
CREATE TABLE Users (  
    UserID INT PRIMARY KEY,  
    Username VARCHAR(50),  
    Password VARCHAR(64)  
);  
  
INSERT INTO Users (UserID, Username, Password)  
VALUES  
    (1, 'john_doe', 'password123'),  
    (2, 'jane_smith', 'securepassword');
```

You are tasked with implementing user authentication using stored procedures. How would you design a stored procedure to verify user credentials?

Question 5

[5 Marks]

Scenario 5: Bulk Data Insertion

```
CREATE TYPE dbo.EmployeeType AS TABLE  
(  
    EmployeeID INT,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50)  
);
```

You need to insert a large amount of data into a table efficiently. How would you design a stored procedure to handle bulk data insertion?

Question 6

[14 Marks]

Scenario 6: Cursor for Iterating Over Rows

```
CREATE TABLE EmployeeTable (  
    EmployeeID INT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    Salary DECIMAL(18, 2),  
    DepartmentID INT  
);  
  
INSERT INTO EmployeeTable (EmployeeID, FirstName, LastName, Salary,  
    DepartmentID)  
VALUES  
    (1, 'John', 'Doe', 50000.00, 1),  
    (2, 'Jane', 'Smith', 60000.00, 1),  
    (3, 'Bob', 'Johnson', 75000.00, 2),  
    ;
```

You have a table with employee information, and you need to create a stored procedure to calculate the total salary expense for each department. How would you approach this using a cursor? Make use of a explicit cursor.

Question 7

[6 Marks]

Scenario 7: IF Statement

Given a table containing employee details, if you aim to extract employees whose salary exceeds a specific threshold, how might you employ an IF statement to refine the outcomes? Refer to the table mentioned in Question 1 and formulate a procedure accordingly.

Question 8

[6 Marks]

Scenario 8: Exception Handling

```
CREATE TABLE YourTable (  
    RecordID INT PRIMARY KEY,  
    YourColumn VARCHAR(50)  
);  
  
INSERT INTO YourTable (RecordID, YourColumn) VALUES  
(1, 'Value1'),  
(2, 'Value2'),  
(3, 'Value3');
```

You have a stored procedure that updates a record in a table. How would you handle an exception if the record does not exist?

Question 9

[6 Marks]

Scenario 9: Basic Transaction

```
CREATE TABLE YourTable (  
    ID INT PRIMARY KEY,  
    YourColumn INT  
);  
INSERT INTO YourTable (ID, YourColumn)  
VALUES  
(1, 10),  
(2, 20),  
(3, 30),  
(4, 40);
```

You have a stored procedure requiring multiple updates, and you aim to guarantee that either all updates are successful, or none of them take effect. How would you employ a transaction to accomplish this objective?

Question 10

[2 Marks]

Write a statement to remove a stored procedure in SQL Server.

<<<<<<End of Exam Paper>>>>>>