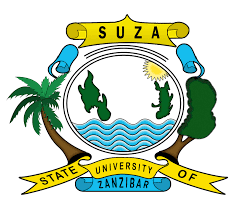
**STATE UNIVERSITY OF ZANZIBAR (SUZA)**



SCHOOL OF COMPUTING, COMMUNICATION AND MEDIA (SCCM)

DEPARTMENT OF COMPUTER SCIENCE AND IT

CLASS ACTIVITY - 24/01/2023

| **INSTRUCTIONS TO STUDENT** |
| --- |

1. Read Carefully, the previous Examination for INF 2112.
2. Use this time to solve the Examination
3. Which question(s) have you found difficult?
4. What do you feel about this paper?
5. Write a small paragraph, writing your reflection on the questions set on the Examination.
6. Submit your work by 27th January, 2023

**SECTION A (24 marks)**

**Answer all questions**

|  | Multiple Choice Question: Choose the most correct answer: **[0.5 mark each]**   1. A \_\_\_\_\_\_\_\_\_\_ is a special kind of a stored procedure that executes in response to certain actions on the table like insertion, deletion or updation of data.   A. Procedures  **B. Triggers**  C. Functions  D. None of the mentioned   1. Management Studio is \_\_\_\_\_\_\_\_ used for development of SQL Server databases. 2. **Front end tool** 3. Back end tool 4. Database engine 5. SQL Server 6. Which of the following statements is true?    1. Views could be looked as an additional layer on the table which enables us to protect intricate or sensitive data based upon our needs    2. Views are virtual tables that are compiled at run time    3. Creating views can improve query response time    4. **All of the Mentioned** 7. What is the meaning of LIKE ‘%0%0%’?    1. Feature begins with two 0’s    2. Feature ends with two 0’s    3. Feature has more than two 0’s    4. **Feature has two 0’s in it, at any position** 8. Which of the following constraints does not enforce uniqueness?    1. UNIQUE    2. Primary key    3. **Foreign key**    4. None of the mentioned 9. Which of the following databases cannot be dropped?    1. TEMPDB    2. Master    3. MSDB    4. **All of the mentioned** |  |
| --- | --- | --- |
|  | 1. Briefly explain how to connect to an SQL Server. What is the information required?   To connect to an SQL Server, you need to provide certain information such as the server name or IP address, the authentication type, and your credentials (username and password). The specific steps to connect to an SQL Server can vary depending on the client software you are using, but here is a general outline of the process:   1. Open the client software (such as SQL Server Management Studio or SQL Server Data Tools) on your computer. 2. In the connection dialog box, enter the server name or IP address. This is the address that the SQL Server is running on. 3. Select the authentication type. SQL Server supports two types of authentication: Windows Authentication and SQL Server Authentication. Windows Authentication uses your Windows login credentials to connect to the server, while SQL Server Authentication requires a separate username and password. 4. Enter your credentials (username and password) if you are using SQL Server Authentication. 5. Click the connect button.   It's worth noting that you may need to ensure that your computer can communicate with the SQL Server and that the SQL Server is configured to accept remote connections.  Additionally, if you need to connect to a specific database, you can enter the name of the database in the "Connect to database" field or you can select it from the list of available databases after connecting to the server. | **[4 marks]** |
| 1. What is T-SQL? How is it different from SQL?   b. What is T-SQL? How is it different from SQL?  T-SQL (Transact-SQL) is the proprietary procedural language used with Microsoft SQL Server. It is an extension of SQL (Structured Query Language), which is a standard language used to manage and manipulate relational databases.  While SQL is a standard language that is used by many different relational databases, T-SQL is specific to Microsoft SQL Server. T-SQL includes additional features and capabilities that are not part of the standard SQL language, such as support for variables, control-of-flow statements, and the ability to create stored procedures, triggers and user-defined functions.  T-SQL also has some enhancements to the standard SQL language such as support for TOP, ROW\_NUMBER() and FOR XML clauses.  Another difference between T-SQL and SQL is that T-SQL includes system functions, system stored procedures and system tables that SQL does not have.  In summary, T-SQL is an extended version of SQL and it's specific to Microsoft SQL Server, it includes additional features, enhancements and system objects that are not part of the standard SQL language. | **[2 marks]** |
| 1. Mention any two ways that T-SQL statements can be written and submitted to the Database engine.   c. Mention any two ways that T-SQL statements can be written and submitted to the Database engine.   1. T-SQL statements can be written and submitted to the Database engine using the SQL Server Management Studio (SSMS). This is a graphical user interface (GUI) that allows you to connect to a SQL Server instance, create and execute T-SQL statements, and manage the database objects. SSMS also provides features such as syntax highlighting, IntelliSense and debugging that can help you write and troubleshoot T-SQL code. 2. T-SQL statements can also be written and submitted to the Database engine using the sqlcmd command-line utility. This is a command-line tool that allows you to connect to a SQL Server instance and execute T-SQL statements. The sqlcmd utility can be run from the command prompt or a batch file, and it supports a variety of options for specifying connection parameters, input files, and output files.   Both of these ways can be used to write and submit T-SQL statements to the Database engine, depending on the user's preference and the context of the task. | **[1 mark]** |
|  | 1. List out four advantages of having more than one SQL Server instance running on a single machine.   List out four advantages of having more than one SQL Server instance running on a single machine.   1. Resource isolation: Running multiple SQL Server instances on a single machine allows for better resource isolation between different workloads. Each instance can be configured with its own memory, CPU, and disk resources, which can help prevent one workload from impacting the performance of another. 2. Security isolation: Having multiple SQL Server instances on a single machine can help provide better security isolation between different databases and applications. Each instance can be configured with its own authentication and authorization settings, which can help prevent unauthorized access to sensitive data. 3. Version compatibility: Running multiple SQL Server instances on a single machine allows you to use different versions of SQL Server for different workloads. This can be useful when you have applications that are not compatible with the latest version of SQL Server but still need to be supported. 4. Cost-effective: Multiple instances of SQL Server running on a single machine can be cost-effective as it allows you to get the most out of your hardware resources. Instead of having multiple servers running different instances of SQL Server, you can use one machine to run multiple instances, which can reduce the costs of hardware, electricity, and maintenance.   It's worth noting that this approach does have some limitations and considerations, such as increased management complexity, and the need for more disk space and memory to run multiple instances. | **[4 marks]** |
| 3. | Assume another person has created a  **student** table with the structure  student(ID,firstname,lastname,email,city\_id,salary)  Write queries to answer the questions: |  |
|  | 1. Add a new column gender in the table.   ALTER TABLE student ADD gender VARCHAR(255); | **[2 marks]** |
|  | 1. Delete last name column from the table.   **ALTER TABLE student DROP COLUMN lastname;** | **[2 marks]** |
|  | 1. Allow the gender column to only accept two values: Male and Female.   ALTER TABLE student ADD CONSTRAINT chk\_gender CHECK (gender in ('Male', 'Female')); | **[3 marks]** |
|  | 1. Populate the data from **student** table to another existing table **registeredStudent** with the following schema:   registeredStudent (ID, firstname, email, salary)  INSERT INTO registeredStudent (ID, firstname, email, salary)  SELECT ID, firstname, email, salary FROM student; | **[3 marks]** |

**SECTION B (36 marks)**

***Answer any THREE* *questions, each question carries 12 marks.***

| 4. | 1. What is the main difference between a view and a user-defined function in MS SQL Server?   In MS SQL Server, a view is a virtual table that is based on the result of a SELECT statement. It can be thought of as a "saved" or "predefined" SELECT statement. A user-defined function, on the other hand, is a programmable object that performs a specific action and returns a value. While a view returns a result set, a user-defined function returns a single value. Additionally, views cannot accept input parameters, while user-defined functions can. | **[2 marks]** |
| --- | --- | --- |
|  | 1. Consider the database schema below:   city(id, city\_name)  program(prog\_id, programName, Description)  student(student\_id, firstname,lastname, city\_id, program\_id)  Use T-SQL to create a view vWStudentDetails that stores students first name, last name, program name and the name of the city that the student is coming from.  CREATE VIEW vWStudentDetails AS  SELECT student.firstname, student.lastname, program.programName, city.city\_name  FROM student  INNER JOIN program ON student.program\_id = program.prog\_id  INNER JOIN city ON student.city\_id = city.id;  This view will join the student, program and city tables together on the program\_id and city\_id columns respectively, and select the first name, last name, program name and city name from each table.  Top of Form | **[4 marks]** |
|  | 1. Write a query to display the number of students in each city, grouped by city names.   To display the number of students in each city, grouped by city names, you can use the following query:  SELECT city.city\_name, COUNT(student.student\_id) as 'Number of Students'  FROM student  INNER JOIN city ON student.city\_id = city.id  GROUP BY city.city\_name;  This query joins the student and city tables together on the city\_id column, and then uses the COUNT function to count the number of student\_id's in each city. The query then groups the results by city name. | **[3 marks]** |
|  | 1. Create a view to display the student full name and the program from which the student is admitted, ordered by programs.   CREATE VIEW student\_program\_view AS SELECT CONCAT(firstname, ' ', lastname) AS full\_name, programName FROM student JOIN program ON student.program\_id = program.prog\_id ORDER BY programName; | **[3 marks]** |
|  |  |  |
| 5. | 1. What is the difference between a stored procedure and a function in MS SQL Server?   In MS SQL Server, a stored procedure is a precompiled collection of Transact-SQL statements that can take input parameters, perform a series of actions, and return a single value or multiple values. They are typically used to perform tasks such as data manipulation or data validation, and can also be used to return a result set.  A function, on the other hand, is also a precompiled collection of Transact-SQL statements, but it is designed to return a single value. Functions can also take input parameters, but they cannot perform any data manipulation or data validation tasks, and they cannot return a result set. Functions are typically used to perform calculations or to retrieve a single value from a table or a view.  Another difference between Stored procedure and function is that Stored procedure does not return any value, where as function returns a single value. Also stored procedure can perform DDL and DCL statements, where as function can not. | **[2 marks]** |
|  | 1. Create a stored procedure called spCountries1 to list out all the countries with ContinentId equal to 1, in alphabetical order   Here's an example of how you might create a stored procedure called **spCountries1** in MS SQL Server to list out all the countries with a **ContinentId** equal to 1, in alphabetical order:  CREATE PROCEDURE spCountries1  AS  BEGIN  SELECT CountryName  FROM Countries  WHERE ContinentId = 1  ORDER BY CountryName  END | **[4 marks]** |
|  | 1. Modify the procedure which modifies the above one,to list out the CountryId as well.   CREATE PROCEDURE spCountries @ContinentId INT  AS  BEGIN  SELECT CountryName  FROM Countries  WHERE ContinentId = @ContinentId  ORDER BY CountryName  END | **[2 marks]** |
|  | 1. Create a function called myAge that accepts DateofBirth and returns Age of the person. Use the function to display the firstname,lastname,DateofBirth and Age of student in a student table with a schema:   student(firstname, lastname,DOB)  CREATE TABLE student (firstname VARCHAR(255), lastname VARCHAR(255),DOB DATE);  INSERT INTO student (firstname, lastname, DOB) VALUES ('John', 'Doe', '1995-12-22');  SELECT firstname,lastname,DOB,DATEDIFF(NOW(),DOB) as age FROM student; | **[4 marks]** |
|  |  |  |
| 6. | 1. What do you understand by the term transaction in T-SQL?   In T-SQL (Transact-SQL), a transaction is a sequence of one or more SQL statements that are executed as a single, atomic unit of work. This means that if any of the statements within the transaction fail, the entire transaction is rolled back and the database is returned to its previous state. Transactions ensure that data integrity is maintained in the event of an error or failure, and they allow multiple operations to be grouped together in a logical unit of work. | **[2 marks]** |
|  | 1. Write a **transaction** which executes an insert statement to add a new event with EventName called My DOB , with your date of birth as the Eventdate column and an appropriate entry for the EventDetails column.   The transaction should test if an Eventname of **My DOB** exists already, adding it if it doesn't and giving a message if it does as shown in Fig. 6(a) and Fig. 6(b):    Fig. 6(a):Adding an event    Fig. 6(b): Result if the event is already added | **[4 marks]** |
|  | 1. The snapshot of the tblCountry is as shown:     Use an UPDATE statement on the tblCountry table to set the CountryName column to include the words (My Country) next to the country you call home (i.e. Tanzania) | **[2 marks]** |
|  | 1. Write a transaction that add (Holiday Destination) to all other countries after running the update statement which:    1. rolls back the transaction if 1 + 1 = 2 (it does); or    2. commit it otherwise. | **[4 marks]** |
|  |  |  |
| 7. | 1. What are DML triggers? | **[1 mark]** |
|  |  |  |
|  | 1. Briefly explain the difference between AFTER triggers and instead of triggers. | **[2 marks]** |
|  |  |  |
|  | 1. Consider the structure of the students table and AuditTable below:   Student(ID, firstname, lastname, email, city\_id, salary, gender)  StudentAudit(id, AuditInfo) |  |
|  | As a DBA, you are required to use DML triggers to create a simple Auditing Information to let you understand when the data in the table student has been inserted or deleted.  The output of the StudentAudit table should look similar to the one below: | **[6 marks]** |
|  | 1. Write a trigger that prevents the salary of the student to be updated, if the updated value is less than the existing value in the table.   The output should look like this one, when the user performs an update that results in the decrease of the existing salary. | **[3 marks]** |
|  |  |  |
| 8. | 1. Briefly explain any four types of SQL Server Backups. 2. Full Backup: A full backup is a complete copy of the entire database. It includes all the data and transaction log files. This type of backup is typically used as the foundation for all other types of backups. 3. Differential Backup: A differential backup contains only the data that has changed since the last full backup. This type of backup is useful for restoring the database to a specific point in time. 4. Transaction Log Backup: A transaction log backup contains only the transaction log files. This type of backup is useful for recovering the database to a specific point in time or for maintaining database consistency after a failure. 5. File and Filegroup Backup: A file and filegroup backup allows you to back up a specific file or filegroup rather than the entire database. This type of backup is useful when you need to restore a single file or filegroup without affecting the rest of the database. 6. Copy-only Backup: A copy-only backup is a type of backup that can be taken without affecting the backup chain and without affecting the ability to restore the database to a specific point in time. This type of backup is useful when you need to take a backup without affecting the existing backup schedule.   It's worth noting that the types of backups you choose will depend on your specific requirements and the nature of your data. It's always good practice to have a backup strategy in place that includes multiple types of backups to ensure that you can recover your data in the event of an unexpected failure. | **[4 marks]** |
|  | 1. Create an SQL Authenticated server Login “Instructor’. The password has to be changed when it is first used. Create a user associated with this login.   USE master;  GO  CREATE LOGIN Instructor WITH PASSWORD = 'password123', CHECK\_POLICY = ON, CHECK\_EXPIRATION = ON;  GO  To create a user associated with this login, you would use the following command:  USE YourDatabase;  GO  CREATE USER Instructor FOR LOGIN Instructor;  GO | **[2 marks]** |
|  | 1. Allow the user to view the data on the customer table and be able to allow another user to view the same.   To allow the user "Instructor" to view the data on the "customer" table, you can use the GRANT SELECT statement as follows:  GRANT SELECT ON customer TO Instructor;  GO  To allow another user, let's say "Student" to view the same data on the customer table, you can use the GRANT SELECT statement as follows:  GRANT SELECT ON customer TO Student;  GO | **[2 marks]** |
|  | 1. Briefly explain any two fixed database roles. 2. db\_owner: This role is a member of the fixed database roles and provides the user with the highest level of permissions on a database. The user is able to perform any action on the database, including creating and modifying objects, modifying security settings, and managing permissions. The user can also grant and revoke permissions to other users. 3. db\_datareader: This role is a member of the fixed database roles and provides the user with the ability to read data from all tables and views in a database. The user can also execute SELECT statements, but they cannot make any changes to the data or the schema. This role is commonly used for reporting and data analysis. 4. db\_datawriter: This role is a member of the fixed database roles and provides the user with the ability to insert, update, and delete data in all tables and views in a database. However, the user cannot make changes to the schema or the security settings. This role is commonly used for applications that need to insert or update data in a database. 5. db\_securityadmin: The db\_securityadmin role is a fixed database role that allows a user to manage the security settings of a SQL Server database. A user in this role can create, alter, or drop logins, create or alter roles, and set or change passwords.   It's important to remember that the roles you choose will depend on the specific requirements of your database, and it's always good practice to grant the minimum necessary permissions to users to maintain the security of your database. | **[2 marks]** |
|  | 1. Differentiate between Logins and Users as used in MS SQL Server.   In MS SQL Server, logins and users are two different concepts.  A login is used to authenticate a user who is trying to connect to the SQL Server instance. A login is created at the server level and can be used to access any database on the server. Logins can be either Windows-authenticated or SQL Server-authenticated. Windows-authenticated logins are based on Windows user accounts, while SQL Server-authenticated logins are based on a username and password combination.  A user, on the other hand, is used to grant or deny access to specific database objects within a database. A user is created within a specific database and is associated with a login. The user is used to control the access to the specific database objects for the associated login.  In short, a Login is used to connect to the SQL Server, whereas a User is used to control access to specific database objects within a database. A user is associated with a login, and a login can have multiple users associated with it, one user per database. | **[2 marks]** |