**Advanced SQL Database final Lab Part 2**

Ibrahim hassan Ibrahim Mohammed

SD - PD

**1. Batch, Script, and Transaction:**

**Batch:**

* A batch is a collection of one or more SQL statements that are executed as a single unit.
* Batches are often used to group related statements or to execute multiple statements together.

**Script:**

* A script is a set of SQL statements that are saved in a file and executed together.
* Scripts can contain one or more batches.

**Transaction:**

* A transaction is a sequence of one or more SQL statements that are executed as a single unit of work.
* Transactions ensure that either all the statements within the transaction are executed, or none of them are. They follow the ACID properties (Atomicity, Consistency, Isolation, Durability).

**2. Trigger and Stored Procedure:**

**Trigger:**

* A trigger is a special type of stored procedure that is automatically executed (or "triggered") in response to certain events, such as INSERT, UPDATE, or DELETE operations on a table.
* Triggers are used to enforce business rules, maintain data integrity, or perform additional actions based on changes to the data.

**Stored Procedure:**

* A stored procedure is a precompiled collection of one or more SQL statements that can be executed as a single unit.
* Stored procedures are explicitly called by the user or application and can accept parameters.

**3. Stored Procedure and Functions:**

**Stored Procedure:**

* A stored procedure is a set of SQL statements that can be executed as a single unit.
* Stored procedures may or may not return values, and they can contain input and output parameters.

**Function:**

* A function is a special type of stored procedure that returns a value.
* Functions are designed to be used in SQL statements, such as SELECT queries, and can be part of expressions.

**4. Drop, Truncate, and Delete Statement:**

**DROP:**

* The DROP statement is used to remove database objects, such as tables, views, or indexes.
* Once an object is dropped, it is permanently deleted from the database.

**TRUNCATE:**

* The TRUNCATE statement is used to remove all rows from a table but retains the table structure for future use.
* TRUNCATE is faster than DELETE, but it cannot be used if the table is referenced by a foreign key constraint.

**DELETE:**

* The DELETE statement is used to remove rows from a table based on a condition.
* DELETE is slower than TRUNCATE, but it offers more flexibility as it allows the use of WHERE clause to specify conditions for deletion.

**5. Select and Select Into Statement:**

**SELECT:**

* The SELECT statement is used to retrieve data from one or more tables in a database.
* It is often used in conjunction with other clauses like WHERE, ORDER BY, GROUP BY, etc., to filter, sort, and organize the retrieved data.

**SELECT INTO:**

* The SELECT INTO statement is used to create a new table and insert the result set of a SELECT query into that table in a single statement.
* It is often used to create a temporary or backup table based on the result of a query.

**6. Local and Global Variables:**

**Local Variables:**

* Local variables are declared and used within a specific scope, such as a stored procedure, function, or a batch of statements.
* They have limited visibility and are typically used for temporary storage of data within a specific block of code.

**Global Variables:**

* Global variables, in the context of databases, often refer to variables that can be accessed from anywhere within a program or session.
* SQL Server doesn't have truly global variables, but you can use session-level variables or system functions to achieve a similar effect.

**7. Convert and Cast Statements:**

**CONVERT:**

* The CONVERT function is used to explicitly convert an expression from one data type to another.
* It allows for more flexibility in specifying the target data type and the format of the result.

**CAST:**

* The CAST function is used for explicit conversion of an expression to a specified data type.
* It provides a simpler syntax compared to CONVERT but may have limitations in terms of format customization.

**8. DDL, DML, DCL, DQL, and TCL:**

**DDL (Data Definition Language):**

* DDL statements are used for defining, altering, and dropping database objects such as tables, indexes, and views.
* Examples include CREATE, ALTER, and DROP statements.

**DML (Data Manipulation Language):**

* DML statements are used for manipulating data stored in the database.
* Examples include SELECT, INSERT, UPDATE, and DELETE statements.

**DCL (Data Control Language):**

* DCL statements are used for controlling access to data within the database.
* Examples include GRANT and REVOKE statements.

**DQL (Data Query Language):**

* DQL refers to the subset of SQL that deals with queries, primarily the SELECT statement.
* It is used for retrieving data from the database.

**TCL (Transaction Control Language):**

* TCL statements are used to manage transactions within a database.
* Examples include COMMIT, ROLLBACK, and SAVEPOINT.

**9. FOR XML RAW and FOR XML AUTO:**

**FOR XML RAW:**

* FOR XML RAW is used in SQL Server to return query results as raw XML, with each row represented as an XML element.
* It provides a simple XML structure where each column is represented as an attribute.

**FOR XML AUTO:**

* FOR XML AUTO is used to return query results as XML, with a more hierarchical structure.
* It automatically generates XML elements and attributes based on the table structure and relationships.

**10. Table-Valued and Multi-Statement Function:**

**Table-Valued Function:**

* A Table-Valued Function (TVF) is a function that returns a table as its result.
* It can be used in the FROM clause of a SELECT statement, similar to a regular table.

**Multi-Statement Function:**

* A Multi-Statement Function is a function that contains multiple SQL statements in its body.
* It can perform complex operations and includes control-of-flow language (BEGIN...END), allowing for more advanced logic.

These comparisons provide an overview of the differences and use cases for each concept in SQL Server.

**11. Varchar(50) and Varchar(max):**

**Varchar(50):**

* Varchar(50) is a variable-length character data type that can store up to 50 characters.
* The number within the parentheses (50 in this case) represents the maximum number of characters allowed for a specific column.

**Varchar(max):**

* Varchar(max) is a variable-length character data type that can store up to 2^31-1 (2,147,483,647) characters.
* It is used when the length of the data is unknown or when the data can be very large.

*Note:* The use of **max** comes with some considerations, and it's important to be mindful of potential performance implications when dealing with very large data.

**12. Datetime, Datetime2(7), and Datetimeoffset(7):**

**Datetime:**

* Datetime is a data type that represents a date and time from January 1, 1753, to December 31, 9999, with an accuracy of 3.33 milliseconds.

**Datetime2(7):**

* Datetime2 is an extension of the Datetime data type with a larger date range and fractional seconds precision.
* Datetime2(7) indicates that it can store up to 7 digits of fractional seconds.

**Datetimeoffset(7):**

* Datetimeoffset is a data type that includes a time zone offset along with date and time information.
* Datetimeoffset(7) indicates that it can store up to 7 digits of fractional seconds.

**13. Default Instance and Named Instance:**

**Default Instance:**

* In SQL Server, the default instance refers to the installation of SQL Server that uses the default settings without specifying an instance name.
* The default instance is identified by the machine name or IP address.

**Named Instance:**

* A named instance is an installation of SQL Server that is assigned a specific instance name during setup.
* It allows for multiple instances of SQL Server to run on the same machine, each with its own configuration and databases.

**14. SQL and Windows Authentication:**

**SQL Authentication:**

* SQL Authentication is a method of authentication where users provide a username and password to connect to the SQL Server.
* User credentials are stored in the SQL Server database, and the authentication is independent of Windows user accounts.

**Windows Authentication:**

* Windows Authentication, also known as Integrated Security, relies on the Windows operating system to authenticate users.
* Users are authenticated based on their Windows login credentials, and SQL Server trusts the Windows security infrastructure for user authentication.

**15. Clustered and Non-Clustered Index:**

**Clustered Index:**

* Defines the physical order of data rows in a table.
* The leaf nodes of the clustered index contain the actual data pages.
* There can be only one clustered index per table.

**Non-Clustered Index:**

* Does not affect the physical order of data rows.
* The leaf nodes of the non-clustered index contain pointers to the actual data pages.
* Tables can have multiple non-clustered indexes.

Choosing between clustered and non-clustered indexes depends on factors such as the type of queries, data distribution, and the overall usage pattern of the database.

**16. Group by ROLLUP and Group by CUBE:**

**ROLLUP:**

* Used for creating subtotals and grand totals in result sets.
* Generates result sets that represent hierarchies of values in the columns listed in the GROUP BY clause.
* Provides a more limited form of grouping compared to CUBE.

**CUBE:**

* Similar to ROLLUP but more powerful.
* Generates result sets that represent all possible combinations of values in the columns listed in the GROUP BY clause.
* Useful for creating cross-tabular reports.

**17. Sequence Object and Identity:**

**Sequence Object:**

* Introduced in SQL Server 2012.
* A user-defined object that generates a sequence of numeric values.
* More flexible than the IDENTITY property, as it can be used independently of a specific table.

Identity:

* A property that is used to automatically generate numeric values for a column.
* Typically associated with a specific table.
* Limited to a single column in a table.

**18. Inline Function and View:**

**Inline Function:**

* A user-defined function that returns a table data type.
* Accepts parameters and can be used in the FROM clause of a query.
* Allows for more dynamic behavior as it can take parameters.

View:

* A virtual table based on the result set of a SELECT statement.
* Does not accept parameters and is essentially a saved query.
* Provides a static representation of data.

**19. Table Variable and Temporary Table:**

**Table Variable:**

* Created and stored in memory.
* Scope is limited to the batch, stored procedure, or function in which it is declared.
* Automatically cleared when the batch or session ends.

**Temporary Table:**

* Created in the tempdb database.
* Can be explicitly dropped or is dropped automatically when the session ends.
* Can have indexes and statistics.

**20. ROW\_NUMBER() and DENSE\_RANK() Function:**

**ROW\_NUMBER():**

* Assigns a unique number to each row within a partition of a result set.
* The numbering is consecutive without gaps.

**DENSE\_RANK():**

* Similar to ROW\_NUMBER() but assigns ranks without any gaps, even if there are tied values.
* Provides a ranking without skipping rank values.

These comparisons should give you a good overview of the differences and use cases for each concept in SQL Server.