ROUTINE

- "routine" refers to a predefined set of instructions. To be used to automatically process data in the database. In general, routines in a database may include:

ROUTINES

- Stored Procedures
- Functions
- Triggers:

STORED PROCEDURES

- Stored Procedures are pre-compiled sets of SQL statements stored in a database
- They are a part of the SQL language that are stored in the database

BENEFITS OF STORED PROCEDURE

- Performance
- Reduce data transfer between application and database
- Execute faster as they are pre-compiled and run on the database server
- Security
- Prevent direct access to database tables
- Allow controlled access to data through defined procedures
- Better Management
- Group related SQL statements together
- Easier to maintain and update

STEPS TO CREATE

- Use the CREATE PROCEDURE statement
- Specify the name of the Stored Procedure
- Write the desired SQL statements
- Define parameters (if any)

STORED PROCEDURE SYNTAX

```
CREATE PROCEDURE procedure_name
AS
sql_statement
GO;
```

EXAMPLE

```
CREATE PROCEDURE SelectAllCustomers
AS
SELECT * FROM Customers
GO;
```

REFERENCE

- https://www.w3schools.com/sql/sql_stored_procedures.asp

https://dev.mysql.com/doc/refman/8.3/en/create-procedure.html

TRIGGER

- **Definition**
- Triggers are special types of stored procedures in a database
- They are automatically executed when a specific event occurs on a table
- Triggers can be used to enforce data integrity, security, and business rules

BENEFITS OF TRIGGERS

- Data Integrity
- Automatically validate and enforce data rules
- Ensure data consistency across related tables

BENEFITS OF TRIGGERS

- Auditing and Logging
- Capture changes made to data
- Log important events and user activities

BENEFITS OF TRIGGERS

- Automation and Workflow
- Automatically perform tasks based on data changes
- Streamline business processes and operations

TRIGGER EVENTS

- INSERT: Fires when a new record is added to the table
- UPDATE: Fires when an existing record is modified
- DELETE: Fires when a record is deleted from the table

TRIGGER TYPES

- BEFORE: Executes the trigger code before the operation
- AFTER: Executes the trigger code after the operation

TRIGGER STRUCTURE

```
CREATE TRIGGER trigger_name
ON table_name
FOR INSERT, UPDATE, DELETE
AS
BEGIN
    -- Trigger code goes here
END
```

TRIGGER EXAMPLE

```
CREATE TRIGGER LogEmployeeUpdates
ON Employees
FOR UPDATE
AS
BEGIN
    INSERT INTO EmployeeAudit (EmployeeID, OldValue, NewValue, UpdatedBy, UpdatedDate)
    SELECT i.EmployeeID, d.FirstName, i.FirstName, SYSTEM_USER, GETDATE()
    FROM inserted i
    JOIN deleted d ON i.EmployeeID = d.EmployeeID
    WHERE i.FirstName <> d.FirstName
END
```

FUNCTIONS IN DATABASES

- Definition
- Functions are reusable programming units in a database
- They encapsulate a specific logic or calculation
- Functions can accept input parameters and return a value

BENEFITS OF FUNCTIONS

- Code Reusability
- Write a function once and use it multiple times
- Promote code organization and maintainability

BENEFITS OF FUNCTIONS

- Readability and Clarity
- Encapsulate complex logic into a named function
- Improve the understandability of your database code

BENEFITS OF FUNCTIONS

- Performance Optimization
- Functions can be optimized and cached by the database
- Reduce the need for repetitive calculations

FUNCTION STRUCTURE

```
CREATE FUNCTION function_name(
    @parameter1 datatype,
    @parameter2 datatype
)
RETURNS return_datatype
AS
BEGIN
    -- Function code goes here
    RETURN @result
END
```

EXAMPLE: GETFULLNAME FUNCTION

```
CREATE FUNCTION GetFullName(
    @FirstName VARCHAR(50),
    @LastName VARCHAR(50)
)
RETURNS VARCHAR(100)
AS
BEGIN
    RETURN @FirstName + ' ' + @LastName
END
```

CALLING A FUNCTION

```
SELECT dbo.GetFullName('John', 'Doe')
```

REFERENCE

- https://www.javatpoint.com/mysql-stored-function
- https://dev.mysql.com/doc/search/?d=333&p=1&q=stor+proce durce