What is a Transaction in SQL?

A **transaction** is a sequence of one or more SQL operations (INSERT, UPDATE, DELETE, etc.) that are treated as a **single logical unit**.

Key Properties (ACID):

- **Atomicity:** All operations succeed or none do
- **Consistency:** Database remains consistent
- **Isolation:** Changes are invisible to other transactions until committed
- **Durability:** Once committed, changes are permanent

Example:

```
BEGIN TRANSACTION;

UPDATE Accounts
SET Balance = Balance - 500
WHERE AccountID = 101;

UPDATE Accounts
SET Balance = Balance + 500
WHERE AccountID = 102;
```

2 Transactions Involving Views

- A **view is a virtual table**; it doesn't store data itself.
- When you perform **DML operations (INSERT, UPDATE, DELETE) on a view**, you are **actually modifying the underlying base table(s)**.
- Therefore, you **can include a view in a transaction**, just like a table.

- **Transactions in SQL**
  - 12
  - A **transaction** in SQL is a sequence of operations performed as a single unit of work. It ensures that either all operations within the transaction are successfully executed or none of them are applied, maintaining the **integrity and consistency** of the database. Transactions are crucial for scenarios like bank transfers, where partial updates can lead to inconsistencies.
  -

1 What is a Stored Procedure?

A **stored procedure** is a **precompiled collection of SQL statements** that can:

- Perform queries (SELECT)
- Modify data (INSERT, UPDATE, DELETE)
- Include logic (IF...ELSE, loops, variables, transactions)
- Accept input parameters and return output

Example:

```
CREATE PROCEDURE GetCustomerAccounts
    @CustomerID INT
AS
BEGIN
    SELECT c.FullName, a.AccountID, a.Balance
    FROM Customers c
    JOIN Accounts a ON c.CustomerID = a.CustomerID
    WHERE c.CustomerID = @CustomerID;
END;
```

What is a Stored Procedure?

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again.

So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.

You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

Stored Procedure Syntax

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

Stored Procedure Example

The following SQL statement creates a stored procedure named "SelectAllCustomers" that selects all records from the "Customers" table:

Example Get your own SQL Server

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