# Views and Procedures

**Eng. Lina Hammad**

**August 19th, 2022**

# SQL View

- In SQL, a view is a virtual table based on the result-set of an SQL statement.
- A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.
- You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.

- **Table vs View**:
  - **Table**: Table is a preliminary storage for storing data and information in RDBMS. A table is a collection of related data entries, and it consists of columns and rows.
  - **View**: A view is a virtual table whose contents are defined by a query. (SELECT query that has been given a name and saved in the database).

- **Advantages over table are:**
- We can combine columns/rows from multiple table or another view and have a consolidated view.
- Views can be used as security mechanisms by letting users access data through the view, without granting the users permissions to directly access the underlying base tables of the view.
- Views reduces the effort for writing queries to access specific columns every time Instead of hitting the complex query to database every time, we can use view
- It acts as abstract layer to downstream systems, so any change in schema is not exposed and hence the downstream systems doesn't get affected.

# SQL View

- **Syntax:**

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

- **Note**: A view always shows up-to-date data! The database engine recreates the data, using the view's SQL statement, every time a user queries a view.

# SQL Views

- We can query the view above as follows:

```
SELECT * FROM view_name;
```

- A view can be updated with the CREATE OR REPLACE VIEW command.

```
CREATE OR REPLACE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

- A view is deleted with the DROP VIEW command.

```
DROP VIEW view_name;
```

# SQL Stored Procedures

- 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 vs View:**
- A view represents a virtual table. You can join multiple tables in a view and use the view to present the data as if the data were coming from a single table.
- A stored procedure uses parameters to do a function... whether it is updating and inserting data or returning single values or data sets.

# SQL Stored Procedures

- **Stored Procedure Syntax:**

```
CREATE PROCEDURE procedure_name()
sql_statement;
```

- To Execute a Stored Procedure:

```
CALL procedure_name();
```

- A procedure can be updated with the CREATE OR REPLACE VIEW command.

```
CREATE OR REPLACE PROCEDURE procedure_name()
sql_statement;
```

# Stored Procedure With One Parameter

- **Stored Procedure with one parameter Syntax:**

```
CREATE PROCEDURE procedure_name(IN parameter_name data_type)
sql_statement
WHERE column_name = parameter_name;
```

- To call the stored procedure with one parameter:

```
Set @variable_name = value;
CALL procedure_name(@variable_name);
```

Or:

```
CALL procedure_name(value);
```

# Stored Procedure With One Parameter

- A stored procedure can be updated by click on it then update the procedure name or the SQL statement then click on GO.
- A stored procedure is deleted with the DROP stored procedure command:

```
DROP PROCEDURE procedure_name;
```