SQL (Generic):

**CreateTable\_SQL.sql:**

-- Create table script

CREATE TABLE ExampleTable (

ID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Age INT

);

**InsertData\_SQL.sql:**

-- Insert data script

INSERT INTO ExampleTable (ID, FirstName, LastName, Age)

VALUES (1, 'John', 'Doe', 30);

INSERT INTO ExampleTable (ID, FirstName, LastName, Age)

VALUES (2, 'Jane', 'Smith', 25);

**QueryData\_SQL.sql:**

-- Query data script

SELECT \* FROM ExampleTable;

**UpdateData\_SQL.sql:**

-- Update data script

UPDATE ExampleTable

SET Age = 31

WHERE ID = 1;

**DeleteData\_SQL.sql**:

-- Delete data script

DELETE FROM ExampleTable

WHERE ID = 2;

**StoredProcedure\_SQL.sql:**

-- Stored procedure script

CREATE PROCEDURE GetAge(IN id INT, OUT age INT)

BEGIN

SELECT Age INTO age

FROM ExampleTable

WHERE ID = id;

END;

**Transactions\_SQL.sql:**

-- Transaction script

BEGIN TRANSACTION;

-- SQL statements here

COMMIT;

**ExceptionHandling\_SQL.sql:**

-- Exception handling script

BEGIN

-- SQL statements here

EXCEPTION

WHEN others THEN

-- Handle exceptions here

END;

**DynamicSQL\_SQL.sql:**

-- Dynamic SQL script

DECLARE

sql\_query VARCHAR(100);

result INT;

BEGIN

sql\_query := 'SELECT COUNT(\*) FROM ExampleTable';

EXECUTE IMMEDIATE sql\_query INTO result;

DBMS\_OUTPUT.PUT\_LINE('Number of rows: ' || result);

END;

/

**Oracle:**

**1. CreateTable\_Oracle.sql:**

-- Create table script for Oracle

CREATE TABLE ExampleTable (

ID NUMBER PRIMARY KEY,

FirstName VARCHAR2(50),

LastName VARCHAR2(50),

Age NUMBER

);

**2. InsertData\_Oracle.sql:**

-- Insert data script for Oracle

INSERT INTO ExampleTable (ID, FirstName, LastName, Age)

VALUES (1, 'John', 'Doe', 30);

INSERT INTO ExampleTable (ID, FirstName, LastName, Age)

VALUES (2, 'Jane', 'Smith', 25);

**3. QueryData\_Oracle.sql:**

-- Query data script for Oracle

SELECT \* FROM ExampleTable;

**4. UpdateData\_Oracle.sql:**

-- Update data script for Oracle

UPDATE ExampleTable

SET Age = 31

WHERE ID = 1;

**5. DeleteData\_Oracle.sql**

-- Delete data script for Oracle

DELETE FROM ExampleTable

WHERE ID = 2;

**6. StoredProcedure\_Oracle.sql:**

-- Stored procedure script for Oracle

CREATE OR REPLACE PROCEDURE GetAge(p\_id IN NUMBER, p\_age OUT NUMBER)

IS

BEGIN

SELECT Age INTO p\_age

FROM ExampleTable

WHERE ID = p\_id;

END;

/

**7. Transactions\_Oracle.sql:**

-- Transaction script for Oracle

BEGIN

-- SQL statements here

COMMIT;

EXCEPTION

WHEN others THEN

ROLLBACK;

END;

**8. ExceptionHandling\_Oracle.sql:**

-- Exception handling script for Oracle

BEGIN

-- SQL statements here

EXCEPTION

WHEN others THEN

-- Handle exceptions here

END;

/

**DynamicSQL\_Oracle.sql:**

-- Dynamic SQL script for Oracle

DECLARE

sql\_query VARCHAR2(100);

result NUMBER;

BEGIN

sql\_query := 'SELECT COUNT(\*) FROM ExampleTable';

EXECUTE IMMEDIATE sql\_query INTO result;

DBMS\_OUTPUT.PUT\_LINE('Number of rows: ' || result);

END;

/

**SQL (Generic):**

**1. CreateTable\_SQL.sql:**

This script creates a table named ExampleTable with columns ID, FirstName, LastName, and Age. The ID column is the primary key.

**2. InsertData\_SQL.sql:**

Inserts two rows of data into the ExampleTable table, providing values for the ID, FirstName, LastName, and Age columns.

**3. QueryData\_SQL.sql:**

Retrieves all rows from the ExampleTable table using a simple SELECT \* statement.

**4. UpdateData\_SQL.sql:**

Updates the Age column to 31 for the row where ID is 1 in the ExampleTable table.

**5. DeleteData\_SQL.sql:**

Deletes the row where ID is 2 from the ExampleTable table.

**6. StoredProcedure\_SQL.sql:**

Defines a simple stored procedure named GetAge that takes an ID as input and returns the corresponding Age using an OUT parameter.

**7. Transactions\_SQL.sql:**

Begins a transaction, executes SQL statements (which you would replace with your actual statements), and commits the transaction if successful. If an error occurs, it rolls back the transaction.

**8. ExceptionHandling\_SQL.sql:**

Demonstrates basic exception handling. If an exception occurs in the enclosed block, it is caught, and you can handle it accordingly.

**9. DynamicSQL\_SQL.sql:**

Declares a dynamic SQL query, executes it using EXECUTE IMMEDIATE, and captures the result. In this case, it counts the number of rows in the ExampleTable.

**Oracle:**

**1. CreateTable\_Oracle.sql:**

Creates a table named ExampleTable with columns ID, FirstName, LastName, and Age. The ID column is the primary key.

**2. InsertData\_Oracle.sql:**

Inserts two rows of data into the ExampleTable table, providing values for the ID, FirstName, LastName, and Age columns.

**3. QueryData\_Oracle.sql:**

Retrieves all rows from the ExampleTable table using a simple SELECT \* statement.

**4. UpdateData\_Oracle.sql:**

Updates the Age column to 31 for the row where ID is 1 in the ExampleTable table.

**5. DeleteData\_Oracle.sql:**

Deletes the row where ID is 2 from the ExampleTable table.

**6. StoredProcedure\_Oracle.sql:**

Defines a simple stored procedure named GetAge that takes an ID as input and returns the corresponding Age using an OUT parameter.

**7. Transactions\_Oracle.sql:**

Begins a transaction, executes SQL statements (which you would replace with your actual statements), and commits the transaction if successful. If an error occurs, it rolls back the transaction.

**8. ExceptionHandling\_Oracle.sql:**

Demonstrates basic exception handling in Oracle. If an exception occurs in the enclosed block, it is caught, and you can handle it accordingly.

**9. DynamicSQL\_Oracle.sql:**

Declares a dynamic SQL query, executes it using EXECUTE IMMEDIATE, and captures the result. In this case, it counts the number of rows in the ExampleTable.