01.1 Referential Integrity Full Code

SQL/Day 2/01 DDL

08 Referential Integrity

Example 1: CASCADE

```sql
-- Create the Customers_Cascade table
CREATE TABLE Customers_Cascade (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(100)
);

-- Create the Orders_Cascade table
CREATE TABLE Orders_Cascade (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customers_Cascade(CustomerID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

-- Insert data into Customers_Cascade table
INSERT INTO Customers_Cascade (CustomerID, CustomerName)
VALUES (1, 'Alice'), (2, 'Bob');

-- Insert data into Orders_Cascade table
INSERT INTO Orders_Cascade (OrderID, CustomerID, OrderDate)
VALUES (100, 1, '2023-01-15'), (101, 2, '2023-01-16');
```

Example 2: SET NULL

```sql
-- Create the Customers_SetNull table
CREATE TABLE Customers_SetNull (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(100)
);

-- Create the Orders_SetNull table
CREATE TABLE Orders_SetNull (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customers_SetNull(CustomerID)
        ON DELETE SET NULL
        ON UPDATE SET NULL
);

-- Insert data into Customers_SetNull table
INSERT INTO Customers_SetNull (CustomerID, CustomerName)
VALUES (1, 'Alice'), (2, 'Bob');
```

```
-- Insert data into Orders_SetNull table
INSERT INTO Orders_SetNull (OrderID, CustomerID, OrderDate)
VALUES (100, 1, '2023-01-15'), (101, 2, '2023-01-16');
```

Example 3: SET DEFAULT

Not supported by MySQL
Try, PostgreSQL, Oracle, SQL Server

```
-- Create the Customers_SetDefault table
CREATE TABLE Customers_SetDefault (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(100)
);

-- Create the Orders_SetDefault table
CREATE TABLE Orders_SetDefault (
    OrderID INT PRIMARY KEY,
    CustomerID INT DEFAULT 0,
    OrderDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customers_SetDefault(CustomerID)
        ON DELETE SET DEFAULT
        ON UPDATE SET DEFAULT
);

-- Insert data into Customers_SetDefault table
INSERT INTO Customers_SetDefault (CustomerID, CustomerName)
VALUES (1, 'Alice'), (2, 'Bob');

-- Insert data into Orders_SetDefault table
INSERT INTO Orders_SetDefault (OrderID, CustomerID, OrderDate)
VALUES (100, 1, '2023-01-15'), (101, 2, '2023-01-16');
```

Example 4: NO ACTION

```
-- Create the Customers_NoAction table
CREATE TABLE Customers_NoAction (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(100)
);

-- Create the Orders_NoAction table
CREATE TABLE Orders_NoAction (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customers_NoAction(CustomerID)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
);

-- Insert data into Customers_NoAction table
INSERT INTO Customers_NoAction (CustomerID, CustomerName)
VALUES (1, 'Alice'), (2, 'Bob');

-- Insert data into Orders_NoAction table
```

```sql
INSERT INTO Orders_NoAction (OrderID, CustomerID, OrderDate)
VALUES (100, 1, '2023-01-15'), (101, 2, '2023-01-16');
```

Example 5: RESTRICT

```sql
-- Create the Customers_Restrict table
CREATE TABLE Customers_Restrict (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(100)
);

-- Create the Orders_Restrict table
CREATE TABLE Orders_Restrict (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customers_Restrict(CustomerID)
        ON DELETE RESTRICT
        ON UPDATE RESTRICT
);

-- Insert data into Customers_Restrict table
INSERT INTO Customers_Restrict (CustomerID, CustomerName)
VALUES (1, 'Alice'), (2, 'Bob');

-- Insert data into Orders_Restrict table
INSERT INTO Orders_Restrict (OrderID, CustomerID, OrderDate)
VALUES (100, 1, '2023-01-15'), (101, 2, '2023-01-16');
```

Testing Referential Integrity Constraints

Test 1: DELETE CASCADE

```sql
-- Delete customer with CustomerID = 1 from Customers_Cascade
DELETE FROM Customers_Cascade WHERE CustomerID = 1;

-- Check Orders_Cascade table to verify the cascading delete
SELECT * FROM Orders_Cascade;
```

Test 2: UPDATE CASCADE

```sql
-- Update CustomerID from 2 to 3 in Customers_Cascade
UPDATE Customers_Cascade SET CustomerID = 3 WHERE CustomerID = 2;

-- Check Orders_Cascade table to verify the cascading update
SELECT * FROM Orders_Cascade;
```

Test 3: DELETE SET NULL

```sql
-- Delete customer with CustomerID = 1 from Customers_SetNull
DELETE FROM Customers_SetNull WHERE CustomerID = 1;

-- Check Orders_SetNull table to verify the set null on delete
SELECT * FROM Orders_SetNull;
```

Test 4: UPDATE SET NULL

```sql
-- Update CustomerID from 2 to 3 in Customers_SetNull
UPDATE Customers_SetNull SET CustomerID = 3 WHERE CustomerID = 2;

-- Check Orders_SetNull table to verify the set null on update
SELECT * FROM Orders_SetNull;
```

Test 5: DELETE SET DEFAULT

```sql
-- Delete customer with CustomerID = 1 from Customers_SetDefault
DELETE FROM Customers_SetDefault WHERE CustomerID = 1;

-- Check Orders_SetDefault table to verify the set default on delete
SELECT * FROM Orders_SetDefault;
```

Test 6: UPDATE SET DEFAULT

```sql
-- Update CustomerID from 2 to 3 in Customers_SetDefault
UPDATE Customers_SetDefault SET CustomerID = 3 WHERE CustomerID = 2;

-- Check Orders_SetDefault table to verify the set default on update
SELECT * FROM Orders_SetDefault;
```

Test 7: DELETE NO ACTION

```sql
-- Delete customer with CustomerID = 1 from Customers_NoAction
DELETE FROM Customers_NoAction WHERE CustomerID = 1;

-- Check Orders_NoAction table to verify no action on delete
SELECT * FROM Orders_NoAction;
```

Test 8: UPDATE NO ACTION

```sql
-- Update CustomerID from 2 to 3 in Customers_NoAction
UPDATE Customers_NoAction SET CustomerID = 3 WHERE CustomerID = 2;

-- Check Orders_NoAction table to verify no action on update
SELECT * FROM Orders_NoAction;
```

Test 9: DELETE RESTRICT

```sql
-- Delete customer with CustomerID = 1 from Customers_Restrict
DELETE FROM Customers_Restrict WHERE CustomerID = 1;

-- Check Orders_Restrict table to verify restrict on delete
SELECT * FROM Orders_Restrict;
```

Test 10: UPDATE RESTRICT

```sql
-- Update CustomerID from 2 to 3 in Customers_Restrict
UPDATE Customers_Restrict SET CustomerID = 3 WHERE CustomerID = 2;
```

```
-- Check Orders_Restrict table to verify restrict on update
SELECT * FROM Orders_Restrict;
```

Expected Results

- **CASCADE**: Deleting a customer will delete their orders; updating a customer ID will update the corresponding orders.
- **SET NULL**: Deleting a customer will set the `CustomerID` in orders to NULL; updating a customer ID will set the `CustomerID` in orders to NULL.
- **SET DEFAULT**: Deleting a customer will set the `CustomerID` in orders to the default value (e.g., 0); updating a customer ID will set the `CustomerID` in orders to the default value.
- **NO ACTION**: Deleting or updating a customer will be rejected if there are corresponding orders.
- **RESTRICT**: Similar to NO ACTION, deleting or updating a customer will be rejected if there are corresponding orders. The difference is mainly in the timing of the enforcement.