Table values

SQL Query Clause Order:

```
START TRANSACTION;

INSERT INTO Sales (SaleID, CustomerID, Product, Amount, Date, Region) VALUES
(1, 'C001', 'Widget', 100, '2024-01-01', 'North'),
(2, 'C002', 'Gadget', 200, '2024-01-02', 'South'),
(3, 'C001', 'Widget', 150, '2024-01-03', 'North'),
(4, 'C003', 'Gadget', 300, '2024-01-04', 'East'),
(5, 'C002', 'Widget', 250, '2024-01-05', 'South'),
(6, 'C001', 'Gadget', 180, '2024-01-06', 'North'),
(7, 'C004', 'Widget', 220, '2024-01-07', 'West'),
(8, 'C003', 'Widget', 130, '2024-01-08', 'East'),
(9, 'C004', 'Gadget', 260, '2024-01-09', 'West'),
(10, 'C002', 'Gadget', 300, '2024-01-10', 'South');

SELECT * FROM Sales;

ROLLBACK;
```

Quiz 2: 02/18/2025

```
-- Create the Publisher table
CREATE TABLE Publisher (
    PublisherID INT AUTO_INCREMENT PRIMARY KEY,
    PublisherName VARCHAR(255),
    PublisherAddress VARCHAR(255),
    PublisherPhone VARCHAR(20)
);

-- Create the Book table with foreign keys to Author and Publisher
CREATE TABLE Book (
    BookID INT PRIMARY KEY,
    Title VARCHAR(255),
    ISBN VARCHAR(20),
    PublicationYear INT,
    Genre VARCHAR(100),
    AuthorID INT,
    PublisherID INT,
    FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID),
    FOREIGN KEY (PublisherID) REFERENCES Publisher(PublisherID)
);
```

```
-- Create the Author table
CREATE TABLE Author (
   AuthorID INT AUTO_INCREMENT PRIMARY KEY,
   AuthorName VARCHAR(255),
   AuthorEmail VARCHAR(255)
);

-- Insert data into Publisher table
INSERT INTO Publisher (PublisherName, PublisherAddress, PublisherPhone) VALUES
('Acme Publishing', '123 Main Street, Cityville', '555-1234'),
('Sunrise Books', '456 Sunrise Blvd, Townsville', '555-5678'),
('Global Reads', '789 Global Rd, Metropolis', '555-9012'),
('Historical House', '101 History Ln, Oldtown', '555-3456'),
('Novel Ideas', '202 Novel St, Fictionville', '555-7890'),
('Economic Press', '303 Finance Ave, Economica', '555-2345'),
('Thumb Publications', '404 Garden Rd, Floral City', '555-6789'),
('Tech Press', '505 Code St, Silicon Valley', '555-4321');

-- Insert data into Author table
INSERT INTO Author (AuthorName, AuthorEmail) VALUES
('John Doe', 'johndoe@example.com'),
('Jane Smith', 'jane.smith@example.com'),
('Alice Johnson', 'alicej@example.com'),
('Cathy Lee', 'cathylee@example.com'),
('Mark Brown', 'markbrown@example.com'),
('Lucy Gray', 'lucy.gray@example.com'),
('Robert Green', 'robert.green@example.com'),
('Emma White', 'emma.white@example.com'),
('David Black', 'david.black@example.com');

INSERT INTO Book (BookID, Title, ISBN, PublicationYear, Genre, AuthorID, PublisherID)
VALUES
(1, 'Book Title 1', '9783161484100', 2015, 'Fiction', 3, 5),
(2, 'Book Title 2', '9783161484101', 2008, 'Mystery', 7, 2),
(3, 'Book Title 3', '9783161484102', 2020, 'Science', 2, 8),
(4, 'Book Title 4', '9783161484103', 2012, 'Adventure', 5, 1),
(5, 'Book Title 5', '9783161484104', 2001, 'Cooking', 8, 4),
(6, 'Book Title 6', '9783161484105', 2003, 'Economics', 1, 7),
(7, 'Book Title 7', '9783161484106', 2018, 'History', 9, 6),
(8, 'Book Title 8', '9783161484107', 2005, 'Gardening', 4, 3),
(9, 'Book Title 9', '9783161484108', 2019, 'Technology', 6, 8),
(10, 'Book Title 10', '9783161484109', 2011, 'Mystery', 2, 1),
(11, 'Book Title 11', '9783161484110', 2007, 'Adventure', 5, 4),
(12, 'Book Title 12', '9783161484111', 2022, 'Cooking', 3, 2),
(13, 'Book Title 13', '9783161484112', 2000, 'Economics', 7, 6),
(14, 'Book Title 14', '9783161484113', 2013, 'Fiction', 9, 3),
(15, 'Book Title 15', '9783161484114', 2002, 'History', 1, 8),
(16, 'Book Title 16', '9783161484115', 2009, 'Science', 4, 5),
(17, 'Book Title 17', '9783161484116', 2016, 'Mystery', 8, 7),
(18, 'Book Title 18', '9783161484117', 2004, 'Adventure', 2, 1),
(19, 'Book Title 19', '9783161484118', 2017, 'Cooking', 6, 4),
(20, 'Book Title 20', '9783161484119', 2010, 'Economics', 3, 2),
(21, 'Book Title 21', '9783161484120', 2006, 'Fiction', 5, 8),
(22, 'Book Title 22', '9783161484121', 2014, 'History', 7, 3),
(23, 'Book Title 23', '9783161484122', 2001, 'Science', 9, 6),
(24, 'Book Title 24', '9783161484123', 2008, 'Mystery', 1, 7),
(25, 'Book Title 25', '9783161484124', 2015, 'Adventure', 4, 2),
(26, 'Book Title 26', '9783161484125', 2003, 'Cooking', 8, 5),
(27, 'Book Title 27', '9783161484126', 2020, 'Economics', 2, 1),
(28, 'Book Title 28', '9783161484127', 2005, 'Fiction', 6, 4),
(29, 'Book Title 29', '9783161484128', 2019, 'History', 3, 8),
(30, 'Book Title 30', '9783161484129', 2012, 'Science', 5, 7),
(31, 'Book Title 31', '9783161484130', 2007, 'Mystery', 7, 2),
(32, 'Book Title 32', '9783161484131', 2021, 'Adventure', 4, 6),
(33, 'Book Title 33', '9783161484132', 2000, 'Cooking', 9, 3),
(34, 'Book Title 34', '9783161484133', 2018, 'Economics', 1, 8),
(35, 'Book Title 35', '9783161484134', 2004, 'Fiction', 3, 5),
(36, 'Book Title 36', '9783161484135', 2010, 'History', 6, 7),
(37, 'Book Title 37', '9783161484136', 2013, 'Science', 8, 1),
(38, 'Book Title 38', '9783161484137', 2009, 'Mystery', 2, 4),
(39, 'Book Title 39', '9783161484138', 2017, 'Adventure', 5, 6),
(40, 'Book Title 40', '9783161484139', 2002, 'Cooking', 7, 3),
(41, 'Book Title 41', '9783161484140', 2006, 'Economics', 9, 8),
(42, 'Book Title 42', '9783161484141', 2014, 'Fiction', 1, 2),
(43, 'Book Title 43', '9783161484142', 2003, 'History', 4, 5),
(44, 'Book Title 44', '9783161484143', 2015, 'Science', 8, 7),
(45, 'Book Title 45', '9783161484144', 2001, 'Mystery', 2, 1),
(46, 'Book Title 46', '9783161484145', 2018, 'Adventure', 6, 4),
(47, 'Book Title 47', '9783161484146', 2007, 'Cooking', 3, 8),
(48, 'Book Title 48', '9783161484147', 2022, 'Economics', 5, 2),
(49, 'Book Title 49', '9783161484148', 2000, 'Fiction', 7, 6),
(50, 'Book Title 50', '9783161484149', 2013, 'History', 9, 3),
(51, 'Book Title 51', '9783161484150', 2008, 'Science', 1, 7),
(52, 'Book Title 52', '9783161484151', 2019, 'Mystery', 4, 2),
(53, 'Book Title 53', '9783161484152', 2005, 'Adventure', 8, 5),
(54, 'Book Title 54', '9783161484153', 2011, 'Cooking', 2, 1),
(55, 'Book Title 55', '9783161484154', 2016, 'Economics', 6, 4),
(56, 'Book Title 56', '9783161484155', 2002, 'Fiction', 3, 7),
(57, 'Book Title 57', '9783161484156', 2009, 'History', 7, 2),
(58, 'Book Title 58', '9783161484157', 2017, 'Science', 9, 5),
(59, 'Book Title 59', '9783161484158', 2004, 'Mystery', 1, 8),
(60, 'Book Title 60', '9783161484159', 2012, 'Adventure', 4, 3),
(61, 'Book Title 61', '9783161484160', 2007, 'Cooking', 8, 6),
(62, 'Book Title 62', '9783161484161', 2020, 'Economics', 2, 1),
(63, 'Book Title 63', '9783161484162', 2003, 'Fiction', 5, 4),
(64, 'Book Title 64', '9783161484163', 2018, 'History', 7, 8),
(65, 'Book Title 65', '9783161484164', 2001, 'Science', 9, 3),
(66, 'Book Title 66', '9783161484165', 2015, 'Mystery', 3, 6),
(67, 'Book Title 67', '9783161484166', 2006, 'Adventure', 6, 2),
(68, 'Book Title 68', '9783161484167', 2011, 'Cooking', 1, 7),
(69, 'Book Title 69', '9783161484168', 2022, 'Economics', 4, 5),
(70, 'Book Title 70', '9783161484169', 2009, 'Fiction', 8, 1),
(71, 'Book Title 71', '9783161484170', 2013, 'History', 2, 4),
(72, 'Book Title 72', '9783161484171', 2008, 'Science', 5, 7),
(73, 'Book Title 73', '9783161484172', 2004, 'Mystery', 7, 2),
(74, 'Book Title 74', '9783161484173', 2016, 'Adventure', 9, 5),
(75, 'Book Title 75', '9783161484174', 2010, 'Cooking', 1, 8),
(76, 'Book Title 76', '9783161484175', 2007, 'Economics', 4, 3),
(77, 'Book Title 77', '9783161484176', 2015, 'Fiction', 6, 2),
(78, 'Book Title 78', '9783161484177', 2002, 'History', 8, 7),
(79, 'Book Title 79', '9783161484178', 2009, 'Science', 3, 4),
(80, 'Book Title 80', '9783161484179', 2017, 'Mystery', 5, 1),
(81, 'Book Title 81', '9783161484180', 2003, 'Adventure', 7, 6),
(82, 'Book Title 82', '9783161484181', 2014, 'Cooking', 9, 3),
(83, 'Book Title 83', '9783161484182', 2000, 'Economics', 2, 8),
(84, 'Book Title 84', '9783161484183', 2011, 'Fiction', 4, 1),
(85, 'Book Title 85', '9783161484184', 2008, 'History', 6, 4),
(86, 'Book Title 86', '9783161484185', 2012, 'Science', 8, 7),
(87, 'Book Title 87', '9783161484186', 2005, 'Mystery', 1, 2),
(88, 'Book Title 88', '9783161484187', 2020, 'Adventure', 3, 5),
(89, 'Book Title 89', '9783161484188', 2007, 'Cooking', 5, 8),
(90, 'Book Title 90', '9783161484189', 2004, 'Economics', 7, 3),
(91, 'Book Title 91', '9783161484190', 2016, 'Fiction', 9, 6),
(92, 'Book Title 92', '9783161484191', 2002, 'History', 2, 1),
(93, 'Book Title 93', '9783161484192', 2009, 'Science', 4, 5),
(94, 'Book Title 94', '9783161484193', 2013, 'Mystery', 8, 7),
(95, 'Book Title 95', '9783161484194', 2006, 'Adventure', 3, 2),
(96, 'Book Title 96', '9783161484195', 2010, 'Cooking', 6, 8),
(97, 'Book Title 97', '9783161484196', 2017, 'Economics', 9, 4),
(98, 'Book Title 98', '9783161484197', 2003, 'Fiction', 1, 5),
(99, 'Book Title 99', '9783161484198', 2015, 'History', 4, 7),
(100, 'Book Title 100', '9783161484199', 2000, 'Science', 7, 2);
```

Electronics Store: 02/21/2025

```
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(50) NOT NULL,
    Email VARCHAR(100) UNIQUE NOT NULL,
    Phone VARCHAR(15),
    City VARCHAR(50),
    JoinDate DATE
);

CREATE TABLE Products (
    ProductID INT PRIMARY KEY AUTO_INCREMENT,
    ProductName VARCHAR(50) NOT NULL,
    Category VARCHAR(50),
    Price DECIMAL(10,2) NOT NULL,
    Stock INT,
    Supplier VARCHAR(50)
);

CREATE TABLE Orders (
    OrderID INT PRIMARY KEY AUTO_INCREMENT,
    CustomerID INT,
    ProductID INT,
    OrderDate DATE,
    Quantity INT NOT NULL,
    OrderStatus VARCHAR(20),
    PaymentMethod VARCHAR(20),
    ShippingAddress VARCHAR(255),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);

INSERT INTO Customers (Name, Email, Phone, City, JoinDate)
VALUES
    ('Alice', 'alice@email.com', '123-456-7890', 'New York', '2022-05-01'),
    ('Bob', 'bob@email.com', '234-567-8901', 'Los Angeles', '2021-08-15'),
    ('Charlie', 'charlie@email.com', '345-678-9012', 'Chicago', '2023-01-10'),
    ('David', 'david@email.com', '456-789-0123', 'Houston', '2020-12-20'),
    ('Eve', 'eve@email.com', '567-890-1234', 'Boston', '2022-11-30'),
    ('Frank', 'frank@email.com', '678-901-2345', 'Seattle', '2023-06-15');


INSERT INTO Products (ProductName, Category, Price, Stock, Supplier)
```

```
VALUES
    ('Laptop', 'Electronics', 1200.00, 10, 'TechWorld'),
    ('Phone', 'Electronics', 800.00, 25, 'MobileInc'),
    ('Headphones', 'Accessories', 150.00, 50, 'SoundMax'),
    ('Keyboard', 'Accessories', 100.00, 30, 'KeyPro'),
    ('Monitor', 'Electronics', 300.00, 15, 'DisplayHub'),
    ('Mouse', 'Accessories', 50.00, 40, 'ClickTech');

INSERT INTO Orders (CustomerID, ProductID, OrderDate, Quantity, OrderStatus,
PaymentMethod, ShippingAddress)
VALUES
    (1, 2, '2024-02-01', 1, 'Delivered', 'Credit Card', '123 NY St, NY'),
    (1, 3, '2024-02-05', 2, 'Shipped', 'PayPal', '123 NY St, NY'),
    (2, 1, '2024-02-07', 1, 'Processing', 'Debit Card', '456 LA St, LA'),
    (3, 4, '2024-02-10', 3, 'Delivered', 'Credit Card', '789 CH St, CH'),
    (4, 2, '2024-02-12', 1, 'Cancelled', 'PayPal', '101 HOU St, HOU'),
    (1, 1, '2024-02-15', 1, 'Delivered', 'Credit Card', '123 NY St, NY'),
    (2, 3, '2024-02-18', 2, 'Shipped', 'Cash', '456 LA St, LA'),
    (3, 2, '2024-02-20', 1, 'Processing', 'Debit Card', '789 CH St, CH'),
    (4, 1, '2024-02-22', 1, 'Delivered', 'Credit Card', '101 HOU St, HOU'),
    (1, 4, '2024-02-25', 2, 'Processing', 'PayPal', '123 NY St, NY'),
    (2, 4, '2024-02-28', 1, 'Shipped', 'Cash', '456 LA St, LA'),
    (3, 1, '2024-03-02', 1, 'Delivered', 'Debit Card', '789 CH St, CH');
```

—--

02/28/2025

```
-- Drop tables if they already exist for a clean setup (optional)
DROP TABLE IF EXISTS Passport;
DROP TABLE IF EXISTS Person;

CREATE TABLE Person (
    person_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(100) NOT NULL,
    last_name VARCHAR(100) NOT NULL
);

CREATE TABLE Passport (
    passport_id INT AUTO_INCREMENT PRIMARY KEY,
    passport_number VARCHAR(50) NOT NULL,
    person_id INT NOT NULL,
    UNIQUE KEY (person_id),
    CONSTRAINT fk_person_passport
```

```
        FOREIGN KEY (person_id)
        REFERENCES Person(person_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

-- Insert into Person
INSERT INTO Person (first_name, last_name)
VALUES ('John', 'Doe'),
       ('Jane', 'Smith'),
       ('Alex', 'Brown');

-- Insert into Passport
-- Each passport references exactly one person
INSERT INTO Passport (passport_number, person_id)
VALUES ('P123456', 1),
       ('P654321', 2),
       ('P121234', 3);

-- One to Many

DROP TABLE IF EXISTS Product;
DROP TABLE IF EXISTS Category;


CREATE TABLE Category (
    category_id INT AUTO_INCREMENT PRIMARY KEY,
    category_name VARCHAR(100)
);

CREATE TABLE Product (
    product_id INT AUTO_INCREMENT PRIMARY KEY,
    product_name VARCHAR(100),
    price DECIMAL(10,2),
    category_id INT NOT NULL,
    FOREIGN KEY (category_id) REFERENCES Category(category_id)
);


-- Insert into Category
INSERT INTO Category (category_name)
VALUES ('Electronics'),
       ('Books'),
       ('Clothing');
```

```
-- Insert into Product
-- Each product references one category
INSERT INTO Product (product_name, price, category_id)
VALUES ('Smartphone', 699.99, 1),
       ('Laptop', 999.99, 1),
       ('Science Fiction Novel', 19.99, 2),
       ('T-Shirt', 9.99, 3),
       ('Jeans', 39.99, 3);


-- Many to Many

DROP TABLE IF EXISTS StudentCourse;
DROP TABLE IF EXISTS Course;
DROP TABLE IF EXISTS Student;

CREATE TABLE Student (
   student_id INT AUTO_INCREMENT PRIMARY KEY,
   student_name VARCHAR(100)
);

CREATE TABLE Course (
   course_id INT AUTO_INCREMENT PRIMARY KEY,
   course_name VARCHAR(100)
);

CREATE TABLE StudentCourse (
   student_id INT NOT NULL,
   course_id INT NOT NULL,
   PRIMARY KEY (student_id, course_id),
   FOREIGN KEY (student_id) REFERENCES Student(student_id),
   FOREIGN KEY (course_id) REFERENCES Course(course_id)
);

-- Insert into Student
INSERT INTO Student (student_name)
VALUES ('Alice'),
       ('Bob'),
       ('Charlie');

-- Insert into Course
INSERT INTO Course (course_name)
VALUES ('Mathematics'),
       ('Physics'),
       ('Literature');

-- Insert into StudentCourse (junction table)
-- Demonstrating many-to-many enrollments
INSERT INTO StudentCourse (student_id, course_id)
VALUES (1, 1),  -- Alice -> Mathematics
       (1, 2),  -- Alice -> Physics
       (2, 1),  -- Bob -> Mathematics
       (2, 3),  -- Bob -> Literature
       (3, 2),  -- Charlie -> Physics
       (3, 3);  -- Charlie -> Literature
```

03/04/2025

```
-- 1. Drop child tables first (due to foreign key dependencies), then drop Project
DROP TABLE IF EXISTS Task;
DROP TABLE IF EXISTS Milestone;
DROP TABLE IF EXISTS TeamMember;
DROP TABLE IF EXISTS Project;

-- 2. Create the 'Project' table (central table)
CREATE TABLE Project (
    project_id INT AUTO_INCREMENT PRIMARY KEY,
    project_name VARCHAR(100) NOT NULL,
    project_description TEXT,       -- Detailed description of the project
    start_date DATE,                -- Project start date
    end_date DATE,                  -- Project end date
    status VARCHAR(50)              -- e.g., 'Planned', 'In Progress', 'Completed'
);

-- 3. Create the 'Task' table
--    Each Task references the Project table
CREATE TABLE Task (
    task_id INT AUTO_INCREMENT PRIMARY KEY,
    task_name VARCHAR(100) NOT NULL,
    task_description TEXT,          -- Detailed description of the task
    due_date DATE,                  -- Task due date
    status VARCHAR(50),             -- e.g., 'Not Started', 'In Progress', 'Done'
    project_id INT NOT NULL,
    CONSTRAINT fk_task_project
        FOREIGN KEY (project_id)
        REFERENCES Project(project_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

-- 4. Create the 'Milestone' table
--    Each Milestone references the Project table
CREATE TABLE Milestone (
    milestone_id INT AUTO_INCREMENT PRIMARY KEY,
    milestone_name VARCHAR(100) NOT NULL,
    milestone_description TEXT,     -- Detailed description of the milestone
    milestone_date DATE,            -- Date when the milestone is expected or achieved
    status VARCHAR(50),             -- e.g., 'Pending', 'Achieved'
    project_id INT NOT NULL,
    CONSTRAINT fk_milestone_project
        FOREIGN KEY (project_id)
        REFERENCES Project(project_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

-- 5. Create the 'TeamMember' table
--    Each TeamMember references the Project table
CREATE TABLE TeamMember (
    member_id INT AUTO_INCREMENT PRIMARY KEY,
    member_name VARCHAR(100) NOT NULL,
    email VARCHAR(100),             -- Email address of the team member
    role VARCHAR(50),               -- Role in the project, e.g., 'Developer', 'Manager'
    project_id INT NOT NULL,
    CONSTRAINT fk_member_project
        FOREIGN KEY (project_id)
        REFERENCES Project(project_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

03/21/2025

```sql
CREATE TABLE Advisor (
    AdvisorID VARCHAR(10) PRIMARY KEY,
    AdvisorName VARCHAR(50) NOT NULL,
    AdvisorRoom VARCHAR(10) NOT NULL
);


CREATE TABLE Student (
    StudentID INT PRIMARY KEY,
    AdvisorID VARCHAR(10),
    FOREIGN KEY (AdvisorID) REFERENCES Advisor(AdvisorID)
);


CREATE TABLE Class (
    ClassID VARCHAR(10) PRIMARY KEY
);

-- StudentClass junction table
CREATE TABLE StudentClass (
    StudentID INT,
    ClassID VARCHAR(10),
    PRIMARY KEY (StudentID, ClassID),
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
    FOREIGN KEY (ClassID) REFERENCES Class(ClassID)
);

-- Insert values into Advisor
INSERT INTO Advisor (AdvisorID, AdvisorName, AdvisorRoom) VALUES
('123A', 'James', '555'),
('123B', 'Smith', '467');

-- Insert values into Student
INSERT INTO Student (StudentID, AdvisorID) VALUES
(123, '123A'),
(124, '123B');

-- Insert values into Class
INSERT INTO Class (ClassID) VALUES
('102-8'),
('104-9'),
('209-0');
```

```
-- Insert values into StudentClass
INSERT INTO StudentClass (StudentID, ClassID) VALUES
(123, '102-8'),
(123, '104-9'),
(124, '209-0'),
(124, '102-8');
```