SQL Data Integrity and Constraints Examples

This document provides SQL examples for demonstrating data integrity concepts and constraint applications.

Data Integrity and Data Integrity Constraints

Data integrity refers to the accuracy and consistency of data stored in a database. Data integrity constraints are rules that are enforced on data to ensure its accuracy and reliability. These constraints prevent invalid data from being entered into the database.

Domain Integrity

Domain integrity ensures that data values in a column are valid. It defines the permissible values for an attribute.

Domain Integrity - SQL Data Types

SQL data types restrict the type of data that can be stored in a column.

```
Unset
-- Example: Creating a table with various data types
CREATE TABLE Products (
    ProductID INT,
    ProductName VARCHAR(255),
    Price DECIMAL(10, 2),
    StockQuantity INT,
    IsActive BOOLEAN,
    DateAdded DATE
);
```

Domain Integrity - NOT NULL and DEFAULT Constraints

NOT NULL ensures that a column cannot have a NULL value. DEFAULT provides a default value for a column if no value is specified.

```
Unset
-- Example: Adding NOT NULL and DEFAULT constraints
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(255) NOT NULL,
    LastName VARCHAR(255) NOT NULL,
    Department VARCHAR(255) DEFAULT 'General'
);
```

Entity Integrity

Entity integrity ensures that each row in a table is uniquely identifiable. It primarily involves the use of primary keys.

Entity Integrity - Primary Keys

A primary key uniquely identifies each record in a table.

```
Unset
-- Example: Creating a table with a primary key
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(255),
    City VARCHAR(255)
);
```

Entity Integrity - Primary Keys for Group of Columns

A composite primary key uses multiple columns to uniquely identify a record.

```
Unset
-- Example: Creating a table with a composite primary key
CREATE TABLE OrderItems (
    OrderID INT,
    ProductID INT,
    Quantity INT,
    PRIMARY KEY (OrderID, ProductID)
);
```

Referential Integrity - Foreign Key Constraints

Referential integrity ensures that relationships between tables are consistent. Foreign keys are used to establish these relationships.

```
Unset
-- Example: Creating a table with a foreign key
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

User-Defined Constraints

User-defined constraints are custom rules created by users to enforce business rules that cannot be enforced by other constraints.

```
Unset
-- Example: Creating a check constraint for a salary
CREATE TABLE Salary (
    EmployeeID INT PRIMARY KEY,
    Salary DECIMAL(10, 2),
    CONSTRAINT CHK_Salary CHECK (Salary >= 0)
);
```