

DATABASE MANAGEMENT SYSTEM – WEEK 02 ASSIGNMENT

Darshana pubudu keerthirathna

ICM 106 OR23106564

Question 01

What are the usages of using SQL constraints?

Constraints are used for specifying rules for the data in the table. Such as **Primary Key, Foreign Key, Unique, Not Null, Default**

Question 02

What is the primary purpose of a primary key in a database table?

The primary purpose of a primary key in a table is to uniquely identify each record or row in that table.

Question 03

Can a primary key consist of multiple columns? If yes, what is it called?

Yes, a primary key can consist of multiple columns, and when it does, it is called a "composite primary key" or "compound primary key."

Question 04

How does a primary key differ from a unique key constraint?

A primary key uniquely identifies each record in a table and cannot contain NULL values, while a unique key only prevents duplicate values in a column and can contain NULL values.

Question 05

Is it possible for a primary key column to have NULL values? Why or why not?

No, a primary key column cannot have NULL values. This is because a primary key must uniquely identify each record in a table.

Allowing NULL values in a primary key would violate this uniqueness constraint.

Question 06

What is a foreign key, and how does it relate to another table? And can a foreign key reference multiple tables?

A foreign key is a column or set of columns in one table that references the primary key of another table. It establishes a relationship between the two tables.

Question 07

How does a unique constraint differ from a primary key constraint?

Same Answer for Question 04

Question 08

Is it possible to have multiple unique constraints on a single table? If yes, explain with an example.

yes, you can have multiple unique constraints on a single table. These constraints ensure that no two rows have the same value for a specified column or set of columns. For example, in a students table, you could have unique constraints on email, student_id_number, phone_number.

Question 09

What is the purpose of a default constraint in SQL?

A default constraint in SQL is used to automatically assign a default value to a column if no value is provided when inserting a new record. This helps prevent NULL values from entering the table.

Question 10

Can a column have both a default constraint and a NOT NULL constraint at the same time?

Yes, a column can have both a default constraint and a NOT NULL constraint at the same time. The default constraint ensures that a value is always assigned to the column, while the NOT NULL constraint prevents NULL values from being inserted.

Question 11

How do default constraints affect existing data when added to a table?

Adding a default constraint to a table only affects new rows. Existing rows retain their values.

Question 12

What is the function of a check constraint in a table?

After creating the table, we can check all the constraints from DESC *"table name"*.

Question 13

Imagine you're creating a database for a bookstore. The database should include tables for books, authors, and categories.

Establish relationships between these tables using appropriate foreign key constraints. Write the complete SQL codes for the given task.

```
CREATE DATABASE book_store;
```

```
USE book_store;
```

```
CREATE TABLE authors(  
    AuthorID INT NOT NULL,  
    AuthorName VARCHAR(30) NOT NULL UNIQUE,  
    CONSTRAINT PRIMARY KEY(AuthorID)  
);
```

```
CREATE TABLE categories(  
    CategoryID INT NOT NULL,  
    CategoryName VARCHAR(20) NOT NULL UNIQUE,  
    CONSTRAINT PRIMARY KEY(CategoryID)  
);
```

```
CREATE TABLE books(  
    BookID INT NOT NULL,  
    Title VARCHAR(50) NOT NULL UNIQUE,  
    AuthorID INT NOT NULL,  
    CategoryID INT NOT NULL,  
    CONSTRAINT PRIMARY KEY(BookID),  
    CONSTRAINT FOREIGN KEY(AuthorID) REFERENCES authors(AuthorID),  
    CONSTRAINT FOREIGN KEY(CategoryID) REFERENCES categories(CategoryID)  
);
```

```
INSERT INTO authors VALUES(1,'J.K. Rowling');
INSERT INTO authors VALUES(2,'Stephen King');
INSERT INTO authors VALUES(3,'Harper Lee');
INSERT INTO authors VALUES(4,'George Orwell');
```

```
INSERT INTO categories VALUES(1,'Fantasy');
INSERT INTO categories VALUES(2,'Horror');
INSERT INTO categories VALUES(3,'Classics');
INSERT INTO categories VALUES(4,'Dystopian');
```

```
INSERT INTO books VALUES(1,'Harry Potter and the Philosopher`s Stone',1,1);
INSERT INTO books VALUES(2,'The Shining',2,2);
INSERT INTO books VALUES(3,'To Kill a Mockingbird',3,3);
INSERT INTO books VALUES(4,'1984',4,4);
```

```
mysql> SELECT * FROM authors;
+-----+-----+
| AuthorID | AuthorName |
+-----+-----+
|         4 | George Orwell |
|         3 | Harper Lee   |
|         1 | J.K. Rowling |
|         2 | Stephen King |
+-----+-----+
4 rows in set (0.00 sec)

mysql> █
```

```
mysql> SELECT * FROM categories;
+-----+-----+
| CategoryID | CategoryName |
+-----+-----+
|          3 | Classics     |
|          4 | Dystopian    |
|          1 | Fantasy      |
|          2 | Horror       |
+-----+-----+
4 rows in set (0.00 sec)

mysql> █
```

```
mysql> USE book_store;
Database changed
mysql> SELECT * FROM books;
+-----+-----+-----+-----+
| BookID | Title                                     | AuthorID | CategoryID |
+-----+-----+-----+-----+
|       1 | Harry Potter and the Philosopher`s Stone |         1 |          1 |
|       2 | The Shining                             |         2 |          2 |
|       3 | To Kill a Mockingbird                   |         3 |          3 |
|       4 | 1984                                    |         4 |          4 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> █
```

Question 14

For a small shop, you should create tables for products, sales, and customers. The products table will store information about available items, the sales table will track transactions, and the customers table will hold information about buyers. Write the complete SQL codes for the given task.

```
CREATE TABLE products(
```

```
    ProductID INT NOT NULL,
```

```
    ProductName VARCHAR(30) NOT NULL,
```

```
    Price DOUBLE(10,2),
```

```
    StockQuantity INT,
```

```
    CONSTRAINT PRIMARY KEY(ProductID)
```

```
);
```

```
CREATE TABLE customer(
```

```
    CustomerID INT NOT NULL,
```

```
    CustomerName VARCHAR(20) NOT NULL,
```

```
    Email VARCHAR(20) UNIQUE,
```

```
    CONSTRAINT PRIMARY KEY(CustomerID)
```

```
);
```

```
CREATE TABLE sales(
```

```
    SaleID INT NOT NULL,
```

```
    ProductID INT NOT NULL,
```

```
    CustomerID INT NOT NULL,
```

```
    Quantity INT NOT NULL,
```

```
    TotalAmount DOUBLE(10,2),
```

```
    SaleDate DATE,
```

```
    CONSTRAINT PRIMARY KEY(SaleID),
```

```
    CONSTRAINT FOREIGN KEY(ProductID) REFERENCES products(ProductID),
```

```
    CONSTRAINT FOREIGN KEY(CustomerID) REFERENCES customer(CustomerID)
```

```
);
```

```
INSERT INTO customer VALUES(1,'John Doe','john@example.com');
INSERT INTO customer VALUES(2,'Jane Smith','jane@example.com');
INSERT INTO customer VALUES(3,'Emily Johnson','emily@example.com');
```

```
INSERT INTO products VALUES(1,'T-Shirt',19.99,50);
INSERT INTO products VALUES(2,'Jeans',39.99,30);
INSERT INTO products VALUES(3,'Sneakers',29.99,40);
INSERT INTO products VALUES(4,'Dress',49.99,20);
```

```
INSERT INTO sales VALUES(101,1,1,2,39.98,'2023-11-15');
INSERT INTO sales VALUES(102,2,2,1,39.99,'2023-11-16');
INSERT INTO sales VALUES(103,3,3,3,89.97,'2023-11-17');
INSERT INTO sales VALUES(104,4,1,1,49.99,'2023-11-18');
```

```
mysql> SELECT*FROM customer;
+-----+-----+-----+
| CustomerID | CustomerName | Email |
+-----+-----+-----+
| 1 | John Doe | john@example.com |
| 2 | Jane Smith | jane@example.com |
| 3 | Emily Johnson | emily@example.com |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> 
```

```
mysql> SELECT*FROM products;
+-----+-----+-----+-----+
| ProductID | ProductName | Price | StockQuantity |
+-----+-----+-----+-----+
| 1 | T-Shirt | 19.99 | 50 |
| 2 | Jeans | 39.99 | 30 |
| 3 | Sneakers | 29.99 | 40 |
| 4 | Dress | 49.99 | 20 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> 
```

```
mysql> SELECT*FROM sales;
+-----+-----+-----+-----+-----+-----+
| SaleID | ProductID | CustomerID | Quantity | TotalAmount | SaleDate |
+-----+-----+-----+-----+-----+-----+
| 101 | 1 | 1 | 2 | 39.98 | 2023-11-15 |
| 102 | 2 | 2 | 1 | 39.99 | 2023-11-16 |
| 103 | 3 | 3 | 3 | 89.97 | 2023-11-17 |
| 104 | 4 | 1 | 1 | 49.99 | 2023-11-18 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> 
```

Question 15

For a small cafe, you should create tables for menu items, orders, and customers. The menu items table will store information about available food/drinks, the orders table will track customer orders, and the customers table will hold information about café visitors. Write the complete SQL codes for the given task.

```
CREATE TABLE menu_items(  
    MenuItemID INT NOT NULL,  
    ItemName VARCHAR(30) NOT NULL,  
    Price DOUBLE(10,2),  
    CONSTRAINT PRIMARY KEY(MenuItemID)  
);
```

```
CREATE TABLE customers(  
    CustomerID INT NOT NULL,  
    CustomerName VARCHAR(20) NOT NULL,  
    Email VARCHAR(20) UNIQUE,  
    CONSTRAINT PRIMARY KEY(CustomerID)  
);
```

```
CREATE TABLE orders(  
    OrderID INT NOT NULL,  
    MenuItemID INT NOT NULL,  
    CustomerID INT NOT NULL,  
    Quantity INT NOT NULL,  
    TotalAmount DOUBLE(10,2),  
    OrderDate DATE,  
    CONSTRAINT PRIMARY KEY(OrderID),  
    CONSTRAINT FOREIGN KEY(MenuItemID) REFERENCES products(ProductID),  
    CONSTRAINT FOREIGN KEY(CustomerID) REFERENCES customers(CustomerID)  
);
```



```
INSERT INTO menu_items VALUES(1,'Fish Bun',110.00);
```

```
INSERT INTO menu_items VALUES(2,'Chicken Bun',130.00);
```

```
INSERT INTO menu_items VALUES(3,'Black Coffee',120.00);
```

```
INSERT INTO menu_items VALUES(4,'Milk Tea',150.00);
```

```
INSERT INTO customers VALUES(1,'Nimal','Nimal@example.com');
```

```
INSERT INTO customers VALUES(2,'Kamal','Kamal@example.com');
```

```
INSERT INTO customers VALUES(3,'Sunimal','Sunimal@example.com');
```

```
INSERT INTO customers VALUES(4,'Rathmal','Rathmal@example.com');
```

```
INSERT INTO orders VALUES(120,1,1,2,220.00,'2023-11-17');
```

```
INSERT INTO orders VALUES(121,2,2,3,390.00,'2023-11-19');
```

```
INSERT INTO orders VALUES(122,3,3,1,120.00,'2023-11-23');
```

```
INSERT INTO orders VALUES(123,4,4,1,150.00,'2023-11-28');
```

```
mysql> SELECT*FROM menu_items;
+-----+-----+-----+
| MenuItemID | ItemName | Price |
+-----+-----+-----+
| 1 | Fish Bun | 110.00 |
| 2 | Chicken Bun | 130.00 |
| 3 | Black Coffee | 120.00 |
| 4 | Milk Tea | 150.00 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> 
```

```
mysql> SELECT*FROM customers;
+-----+-----+-----+
| CustomerID | CustomerName | Email |
+-----+-----+-----+
| 1 | Nimal | Nimal@example.com |
| 2 | Kamal | Kamal@example.com |
| 3 | Sunimal | Sunimal@example.com |
| 4 | Rathmal | Rathmal@example.com |
+-----+-----+-----+
4 rows in set (0.02 sec)

mysql> 
```

```
mysql> SELECT*FROM orders;
+-----+-----+-----+-----+-----+-----+
| OrderID | MenuItemID | CustomerID | Quantity | TotalAmount | OrderDate |
+-----+-----+-----+-----+-----+-----+
| 120 | 1 | 1 | 2 | 220.00 | 2023-11-17 |
| 121 | 2 | 2 | 3 | 390.00 | 2023-11-19 |
| 122 | 3 | 3 | 1 | 120.00 | 2023-11-23 |
| 123 | 4 | 4 | 1 | 150.00 | 2023-11-28 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> 
```