Queries with Tables & Constraints

1-Tut: SQL Query - 1

Send Feedback

Problem Statement :

Write a query for creating a table named **People**, which contains information given in the table below:

Attribute	Data Type
PID	INT (Primary Key)
LastName	VARCHAR
FirstName	VARCHAR
Address	VARCHAR
City	VARCHAR

Print the Table schema once created as follows:

```
SELECT table_name, column_name, data_type
FROM information_schema.columns
WHERE table_name =<TABLE_NAME>
ORDER BY column_name;
```

Note: Write keywords of syntax in uppercase alphabets.

```
CREATE TABLE People (PID INT UNIQUE,
                     FirstName VARCHAR(255),
                     LastName VARCHAR(255),
                     Address VARCHAR(255),
                     City VARCHAR(255));
```

```
SELECT
table_name,
column_name,
data_type
FROM
information_schema.columns
WHERE
table_name = 'People'
   ORDER BY
   column_name;
```

2-Tut: **SQL Query -2**

Send Feedback

Problem Statement :

Write a query for creating a table named **Patients**, which contains the attribute given in the table below:

Attribute	Data Type
Patient_id	INT (Primary Key)
Patient_title	CHAR (NOT NULL)
Patient_name	CHAR (NOT NULL)
admit_date	DATE

Print the Table Schema once created as follows:

```
SELECT table_name, column_name, data_type
 FROM information_schema.columns
 WHERE table_name = <TABLE_NAME>
 ORDER BY column_name;
```

Note: Write keywords of syntax in uppercase alphabets.

```
CREATE TABLE Patients (Patient_id INT NOT NULL,
                       Patient_title CHAR(255) NOT NULL,
                       Patient_name CHAR(255) NOT NULL,
                       admit_date DATE,PRIMARY KEY(Patient_id));
```

```
SELECT
table_name,
column_name,
data_type
FROM
information_schema.columns
WHERE
table_name = 'Patients'
   ORDER BY
   column_name ;
```

3-Tut : MCQ - 1

Send Feedback

Which of the following can accept NULL values?

Options

This problem has only one correct answer

Primary Key
Unique Key
Both
None of the above

Correct Answer : B

Solution Description

A primary key defines the columns that uniquely identify rows in a table. When you create a primary key constraint, none of the columns included in the primary key can have NULL constraints; so Primary Key does not permit NULL values.

However, You can insert NULL values into columns with the UNIQUE constraint because NULL is there due to the absence of a value, so actually, it is never equal to other NULL values and hence not considered a duplicate value

4-Tut : MCQ - 2

Send Feedback

Which of the following is not a Key in MySQL?

Options

This problem has only one correct answer

Primary
Secondary
Alternate
Foreign

Correct Answer : B

Solution Description

There is no “secondary” key type in SQL, instead, there is an “Alternate” key type.

5-Tut: MCQ - 3

Send Feedback

Which of the following statements regarding Foreign Key is Incorrect.

Statement - 1 : Foreign key is used to link two tables together.

Statement - 2 : Foreign key allows inserting NULL values.

Statement - 3 : Foreign key doesn't allow inserting NULL values.

Statement - 4 : Foreign key uniquely identifies tuples of a relation.

Options

This problem has only one correct answer

3 & 4
2 & 4
2, 3 & 4
1, 3 & 4

Correct Answer : A

Solution Description

The foreign key is used to link two tables, via adjoining common column.
The foreign key is the primary key of another table, and it is possible to have more than one foreign key in a particular table.
Though NULL values are not allowed in the case of the primary key, it is allowed in the case of the foreign key.

The Primary key uniquely identifies tuples of a relation. If there is another key uniquely identifying the tuples of relation, than that key will be called the Candidate key.

6-Tut : SQL Query - 4

Send Feedback

Problem Statement:

Consider the tables given below:

1. The table **users** contains features like *id, full_name, enabled, last_login.* The attribute *id* here will be the *primary key*.

Attribute	Datatype
id	INT (Primary key)
full_name	VARCHAR
enabled	CHAR
last_login	VARCHAR

Attribute List(s):

2.The table **addresses** contains features like *user_id, street, city, state*. The *user_id* here will be the *primary key* as well as *foreign key* that refers to *id(users table)*. The attributes *street, city and state* should be declared *NOT NULL*.
Attribute List(s):

Attribute	Datatype
user_id	INT (Primary key)
street	VARCHAR
city	VARCHAR
state	VARCHAR

Formulate a SQL query to create these tables with all the constraints given along with the table. Also, add constraints to correlate these tables.

Print the Table Schema for each table created as follows:

```
SELECT table_name, column_name, data_type
FROM information_schema.columns
WHERE table_name = <TABLE_NAME>;
```

Note - 1: First print table **users** and then **addresses**. Position the above command just after the CREATE TABLE command for each table.

Note - 2: Write keywords of syntax in uppercase alphabets.

```
CREATE TABLE users(id INT UNIQUE NOT NULL,
                   full_name VARCHAR(255),
                   enabled CHAR,
                   last_login DATE);

SELECT
table_name,
column_name,
data_type
FROM
information_schema.columns
WHERE
table_name = 'users';
```

```
CREATE TABLE addresses(user_id INT PRIMARY KEY,
                FOREIGN KEY(user_id) REFERENCES users(id),
                street VARCHAR(255) NOT NULL,
                city VARCHAR(255) NOT NULL,
                state VARCHAR(255) NOT NULL);
SELECT
table_name,
column_name,
data_type
FROM
information_schema.columns
WHERE
table_name = 'addresses';
```

7-Tut : SQL Query - 5

Send Feedback

Problem Statement:

Consider the tables given below and formulate a SQL query to create these tables with all the constraints.

1. The table **books** contain features like *id, title, author, publish_date, isbn*. The *id* should be declared as *PRIMARY KEY*. The *isbn* should be declared *UNIQUE*. The attributes *title, author and publish_date* should be declared *NOT NULL*.

Attribute	Data type
id	INT (Primary Key)
title	VARCHAR (100)
author	VARCHAR (100)
publish_date	TIMESTAMP
isbn	CHAR (12)

2. The Table **reviews** contains features like *id, book_id, reviewer_name, content, rating, published_date* . The *id* should be declared *PRIMARY KEY* . The *bookid should be declared FOREIGN KEY referred to id (table **book**). The attribute bookid* should be declared *NOT NULL* .

Attribute	Data type
id	INT (Primary Key)
book_id	INT
reviewer_name	VARCHAR (255)
content	VARCHAR (255)
rating	INT
publish_date	TIMESTAMP

Print the Table Schema for each table created as follows:

```
SELECT table_name, column_name, data_type
FROM information_schema.columns
WHERE table_name = <TABLE_NAME>;
```

Note - 1: First print table **books** and then **reviews**. Position the above command just after the CREATE TABLE command for each table.

Note - 2: Write keywords of syntax in uppercase alphabets.

```
CREATE TABLE books(id int Primary Key,
                   title varchar(255) NOT NULL,
                   author varchar(255) NOT NULL,
                   published_date TIMESTAMP NOT NULL,
                   isbn CHAR UNIQUE);
SELECT
table_name,
column_name,
data_type
FROM
information_schema.columns
WHERE
table_name = 'books';
```

```
CREATE TABLE reviews(id int Primary Key,
                book_id int NOT NULL,
                foreign key(book_id) references books(id),
                reviewer_name varchar,
                content varchar,
                rating int,
                published_date TIMESTAMP );

SELECT
table_name,
column_name,
data_type
FROM
information_schema.columns
WHERE
table_name = 'reviews';
```

8-Tut : SQL Query - 3

Send Feedback

Problem Statement:

Write multiple queries to create two tables named customer and contacts:

1. The table **customer** contains features like ID, Name, City where all columns are never NULL. The ID here will be the primary key.

Attribute	Data Type
ID	INT (Primary Key)
Name	VARCHAR (50)
City	VARCHAR (50)

2. The Table **contacts** contain features like ID, Customer_Id, Customer_info, Type. Customer_id being a foreign key that refers to ID(customer table). Also Customer_info and Type are never NULL.

Attribute	Data Type
ID	INT (Primary Key)
Customer_Id	INT
Customer_Info	VARCHAR (50)
Type	VARCHAR (50)

Note: Describe both the tables as well. The syntax for describe is given below. Describe the Customer table first and then Contacts Table.

```
DESC <TABLE_NAME>;
```

Note: Write keywords of syntax in uppercase alphabets.

```
create table customer(id int PRIMARY KEY,
                name varchar(50) NOT NULL,
                city varchar(50) NOT NULL);

create table contacts(id int,
                customer_id int,
                foreign key(customer_id) references customer(id),
                customer_info varchar(50) NOT NULL,
                type varchar(50) NOT NULL);
DESC customer;
desc contacts;
```

9-Tut : SQL Query - 6

Send Feedback

Problem Statement :

Given a table named **members**, write a query to add a column named cc_number (Datatype - VARCHAR).

Note : Print the Table Schema once created as follows:

```
SELECT table_name, column_name, data_type
FROM information_schema.columns
WHERE table_name = <TABLE_NAME>
```

```
   ORDER BY column_name;
```

Note: Write keywords of syntax in uppercase alphabets.

```
ALTER TABLE members add cc_number VARCHAR(20);

SELECT
table_name,
column_name,
data_type
FROM
information_schema.columns
WHERE
table_name = 'members'
   ORDER BY
   column_name;
```

10-Tut : **SQL Query - 7**

Send Feedback

Problem Statement :

Given a table named Bank, write a query to change the existing column person_id to Pid VARCHAR(50).

table **Bank**

Attribute	Data type
person_id	INT (Primary Key)
full_name	VARCHAR (30)
acc_no	INT
last_trans	VARCHAR (200)
phone_no	VARCHAR (200)

Syntax to describe the above table.

```
DESC <TABLE_NAME>;
```

Note: Write keywords of syntax in uppercase alphabets.

```
ALTER TABLE Bank CHANGE column person_id Pid varchar(50);
DESC Bank;
```

11-Tut : MCQ - 4

Send Feedback

Which statement is used to delete all rows in a table without logging the individual row deletions?

Options

This problem has only one correct answer

DELETE
REMOVE
DROP
TRUNCATE

Correct Answer : D

Solution Description

TRUNCATE Command removes all rows from a table or specified partitions of a table without logging the individual row deletions.

12-Tut : SQL Query - 8

Send Feedback

Problem Statement :

Given a table named members, write a query to remove a column named member_dob.

Print the Table Schema once created as follows:

```
SELECT table_name, column_name, data_type
FROM information_schema.columns
WHERE table_name = <TABLE_NAME>
ORDER BY column_name;
```

Note: Write keywords of syntax in uppercase alphabets.

```
alter table members drop member_dob;
SELECT
  table_name,
  column_name,
  data_type
FROM
  information_schema.columns
WHERE
  table_name = 'members'
ORDER BY
  column_name ;
```

13-Tut : SQL Query - 9

Send Feedback

Problem Statement :

Write a query to rename the Table consumers to Consumer_Data.

Print the Table Schema once created as follows:

```
SELECT table_name, column_name, data_type
FROM information_schema.columns
WHERE table_name = <TABLE_NAME>
ORDER BY column_name;
```

Note: Write keywords of syntax in uppercase alphabets.

```
Alter table consumers rename to Consumer_data;
SELECT
  table_name,
  column_name,
  data_type
FROM
  information_schema.columns
WHERE
  table_name = 'consumer_data'
ORDER BY
  column_name;
```

14-Tut : SQL Query

Send Feedback

Problem Statement :

Write a SQL query to remove the attribute 'last_login' and rename the 'full_name' to 'customer_name' in the customer table.

Table **customer**

Attribute	Data type
Id	INT (Primary Key)
full_name	VARCHAR (30)
totalOrders	INT
last_login	DATE
phone_no	VARCHAR (200)

Syntax to describe the table.

`DESC <TABLE_NAME>;`

Note: Write keywords of syntax in uppercase alphabets.

```
ALTER TABLE customer drop COLUMN last_login;
ALTER TABLE customer RENAME COLUMN full_name to customer_name;
DESC customer;
```