E commerce database

Let's take an example, suppose I have an e-commerce business, and have data relating to customers, items, and supply. I have practiced my knowledge of SQL to create a database, called BUSINESS to store this information in a structured way. The task is to create the database and 3 tables, and then add data to the relevant tables. For this example, I will be using MySQL and MySQL Workbench.

DATABASE – BUSINESS

TABLE 1 - Customer

C_ID	C_NAME	SURNAME	LOCATION
C1	ANNA	SHELL	LONDON
C2	JOHN	SMITH	MANCHESTER
C3	TOM	SHARP	ROME
C4	GEMMA	BROWN	PARIS
C5	JACK	WILLIAMS	PARIS
C6	EMILY	JOHNSON	ROME
C7	OLIVIA	CLARK	LONDON

TABLE 2 – Item

I_ID	I_NAME	PRICE
I1	TOTE BAG	15
I2	SCRUNCHIE	4
I3	PILLOW	25
I4	HEADBAND	6
I5	VASE	35

TABLE 3 – Supply

I_ID	QUANTITY	LOCATION
I1	200	ROME
I2	650	LONDON
I3	350	PARIS
I4	500	PARIS
I5	150	MANCHESTER

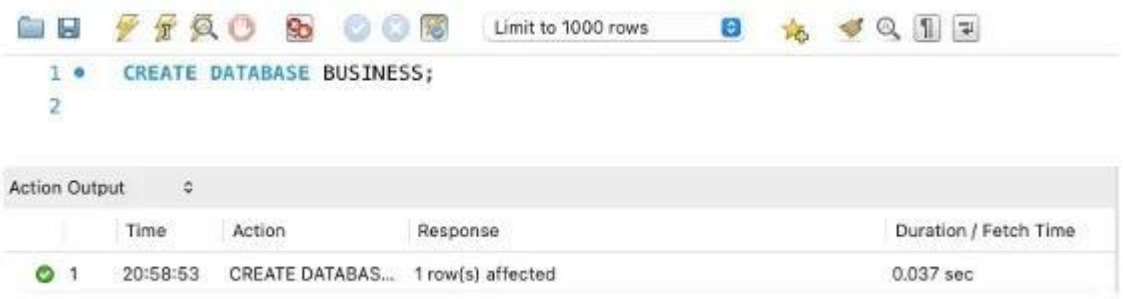
Step 1 — Creating a database

The syntax for creating a database is:

CREATE DATABASE <database_name>;

Following the creation of a database, state:

USE DATABASE <database_name>;



Step 2 — Creating a table

The syntax for creating a table in a database is:

CREATE TABLE <table_name>

(column_name Type,

column_name2 Type,column_name3 Type);

```
1     -- CREATE DATABASE BUSINESS;
2
3 •   USE BUSINESS;
4 •   CREATE TABLE CUSTOMER
5     (C_ID char(2),
6     C_NAME varchar(20),
7     SURNAME varchar(20),
8     LOCATION varchar(30));
9
10 •  CREATE TABLE ITEM
11    (I_ID char(2),
12    I_NAME varchar(20),
13    PRICE INTEGER);
14
15 •  CREATE TABLE SUPPLY
16    (I_ID char(2),
17    QUANTITY INTEGER,
18    LOCATION varchar(30));
19
```

Step 3 — Add data into the table

The syntax for this is:

INSERT INTO <table_name>

(column_name1, column_name2, column_name3)

VALUES

(value1, value2, value3),

(value1, value2, value3);

```
19
20 •  INSERT INTO CUSTOMER
21    (C_ID, C_NAME, SURNAME, LOCATION)
22    VALUES
23    ('C1', 'ANNA', 'SHELL', 'LONDON'),
24    ('C2', 'JOHN', 'SMITH', 'MANCHESTER'),
25    ('C3', 'TOM', 'SHARP', 'ROME'),
26    ('C4', 'GEMMA', 'BROWN', 'PARIS'),
27    ('C5', 'JACK', 'WILLIAMS', 'PARIS'),
28    ('C6', 'EMILY', 'JOHNSON', 'ROME'),
29    ('C7', 'OLIVIA', 'CLARK', 'LONDON');
30
31 •  INSERT INTO ITEM
32    (I_ID, I_NAME, PRICE)
33    VALUES
34    ('I1', 'TOTE BAG', 15),
35    ('I2', 'SCRUNCHIE', 4),
36    ('I3', 'PILLOW', 25),
37    ('I4', 'HEADBAND', 6),
38    ('I5', 'VASE', 35);
39
```

```
40
41 •   INSERT INTO SUPPLY
42     (I_ID, QUANTITY, LOCATION)
43     VALUES
44     ('I1', 200, 'ROME'),
45     ('I2', 650, 'LONDON'),
46     ('I3', 350, 'PARIS'),
47     ('I4', 500, 'PARIS'),
48     ('I5', 150, 'MANCHESTER');
49
```

For checking If the data has been added in the way we have anticipated by running a SELECT query.

```
1 •   SELECT * FROM BUSINESS.CUSTOMER;
```

100% 1:1

Result Grid Filter Rows: Search Export:

C_ID	C_NAME	SURNAME	LOCATION
C1	ANNA	SHELL	LONDON
C2	JOHN	SMITH	MANCHESTER
C3	TOM	SHARP	ROME
C4	GEMMA	BROWN	PARIS
C5	JACK	WILLIAMS	PARIS
C6	EMILY	JOHNSON	ROME
C7	OLIVIA	CLARK	LONDON

We have successfully created a database, multiple tables, and added relevant data to allow you to store the information

Now let me describe and take an example for Primary and Foreign key

Primary Key:-A Primary Key is a column (or a set of columns) in a table that uniquely identifies each row in that table. It ensures that no two rows have the same identifier, and it cannot contain null values.

A Foreign Key is a column (or a set of columns) in one table that references the Primary Key of another table. It establishes a relationship between the two tables and enforces referential integrity, ensuring that data in the foreign key column matches data in the referenced primary key column.

Now let's take an example, Let's say you are managing a school's database.

1. `Students` Table: Stores student details with a unique `StudentID` as the Primary Key.

2. `Courses` Table: Tracks which students are enrolled in which courses, using `StudentID` as a Foreign Key to reference the `Students` table.

`Students` Table

StudentID	Name	Age	Class
1	Alice	14	9A
2	Bob	15	10B
3	Charlie	14	9C

`Courses` Table

CourseID	CourseName	StudentID
101	Mathematics	1
102	Physics	2
103	Chemistry	1

Here:

- `StudentID` in the `Students` table is the Primary Key.

- `StudentID` in the `Courses` table is the Foreign Key referencing the `Students` table.

This setup ensures:

1. No duplicate or invalid students in the `Students` table.

2. Only valid `StudentID`s can be added to the `Courses` table.