Dbase protect\_spam

DBMS ASSIGNMENT-1

P. Sai Charan

1602-21-737-045

IT-A

**Abstract:**

The Database Protect-Spam project is a sophisticated database system designed to protect users from spam by providing relevant suggestions based on their input. The system consists of five tables, including User, Category, Input, Keyword and Suggestion which capture and analyse user input data and provide appropriate feedback. Machine learning algorithms help identify patterns and relationships between different user inputs, enabling highly accurate and relevant suggestions. Users can provide feedback on the suggestions provided, further refining and improving the system's performance over time. Overall, the Database Protect-Spam system provides a powerful and user-friendly approach to spam protection, making it an essential tool for anyone looking to stay safe and secure online.

**Requirements:**

Tables that I have identified are: User, Category, Input, Keyword and Suggestion

**1.Users Table:**

|  |  |  |
| --- | --- | --- |
| Attribute | Domain | CONSTARINT |
| User\_id | NUMBER | Primary\_Key |
| Username | VARCHAR | Not Null |
| Password | VARCHAR | Not Null |
| Email | VARCHAR | Not Null |

**2.Category Table:**

|  |  |  |
| --- | --- | --- |
| Attribute | Domain | CONSTRAINT |
| Category\_id | NUMBER | Primary\_Key |
| Name | VARCHAR | Not Null |
| Description | VARCHAR | Not Null |

|  |  |  |
| --- | --- | --- |
| Attribute | Domain | Constraint |
| Input\_id | NUMBER | Primary\_Key |
| User\_id | INT | Foreign Key |
| Input\_text | TEXT | Not Null |
| Timestamp | TIMESTAMP | Not Null |
| Category\_id | INT | Foreign key |
|  |  |  |

**3.Input Table:**

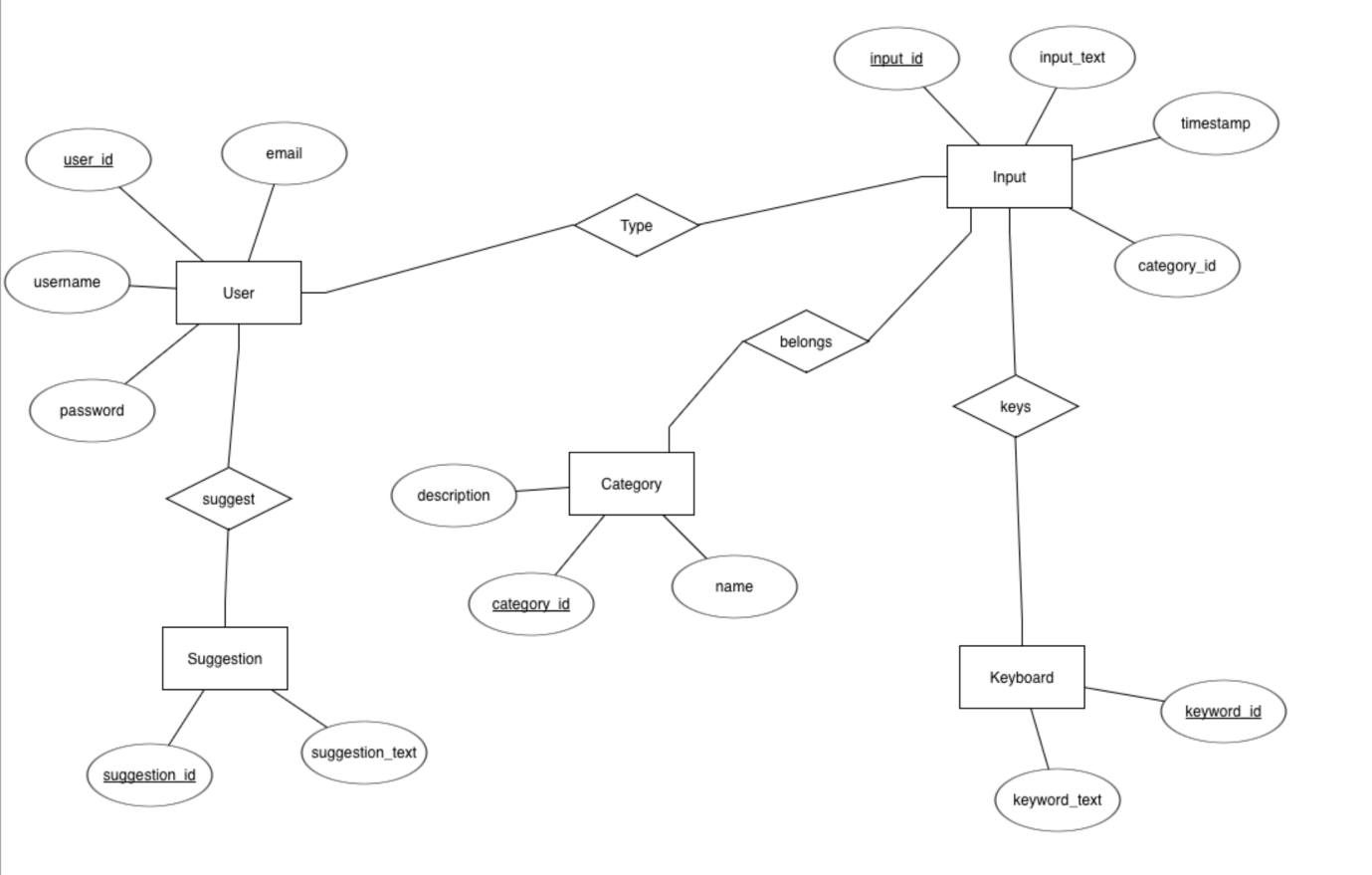
**4.Keyword Table:**

|  |  |  |
| --- | --- | --- |
| Attribute | Domain | CONSTRAINT |
| Keyword\_id | NUMBER | Primary\_Key |
| Input\_id | INT | Foreign Key |
| Keyword\_text | TEXT | Not Null |

**5.Suggestion Table**

|  |  |  |
| --- | --- | --- |
| Attribute | Domain | CONSTRAINT |
| Suggestion\_id | INT | Foreign\_Key |
| Input\_id | INT | Foreign Key |
| Suggestion\_text | TEXT | Not Null |

**ER Diagram:**

****

**DDL COMMANDS:**

1.Creating table for Users with constraints :

**QUERY:** create table Users(

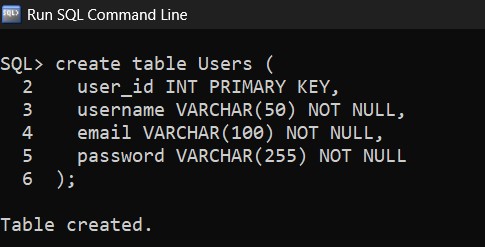
2 user\_id int,

3 username varchar(50),

4 email varchar(100),

5 password varchar(225),

6 primary key(user\_id));



2. Creating Category table:

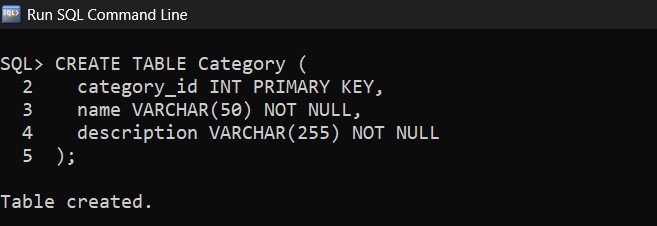
**QUERY:** create table Category(

2 category\_id number(2),

3 name varchar(50) not null,

4 description varchar(255),

4 primary key(category\_id));



3.Creating Input table:

**QUERY:** create table Input(

2 input\_id number(4) primary key,

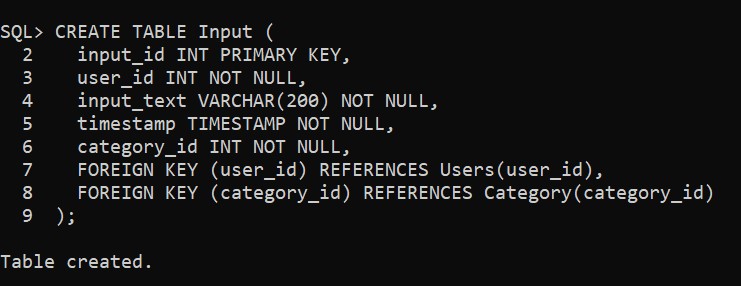
3 user\_id int,

4 input\_text varchar(200),

5 timestamp varchar(20),

6 category\_id int,

7 foreign key(user\_id) references Users;



4.Creating Keyword table:

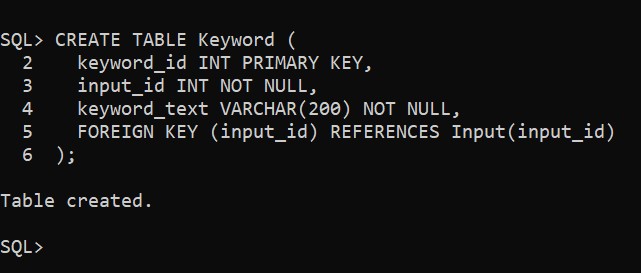
**QUERY:** create table Keyword(

2 keyword\_id int primary key,

3 input\_id int,

4 keyword\_text varchar(200),

5 foreign key(input\_id) references Input;



5.Creating Suggestion table:

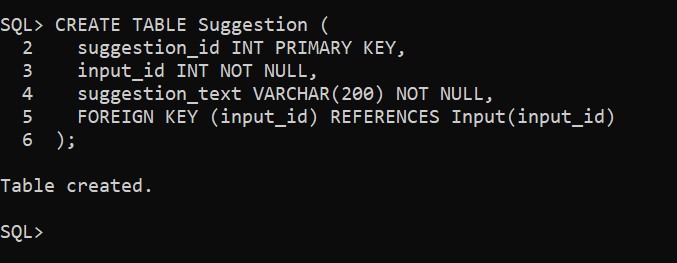
**QUERY:** create table Suggestion(

2 suggestion\_id int primary key,

3 input\_id int,

4 suggestion\_text varchar(200),

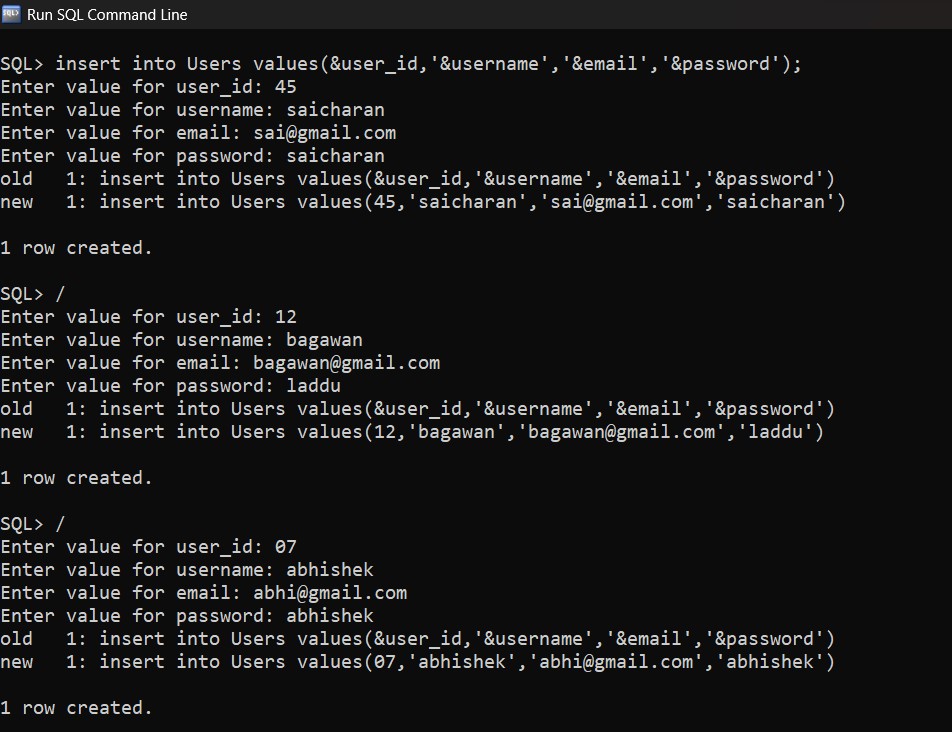
5 foreign key(input\_id) references Input;



**DML COMMANDS:**

1.Insert values into Users:

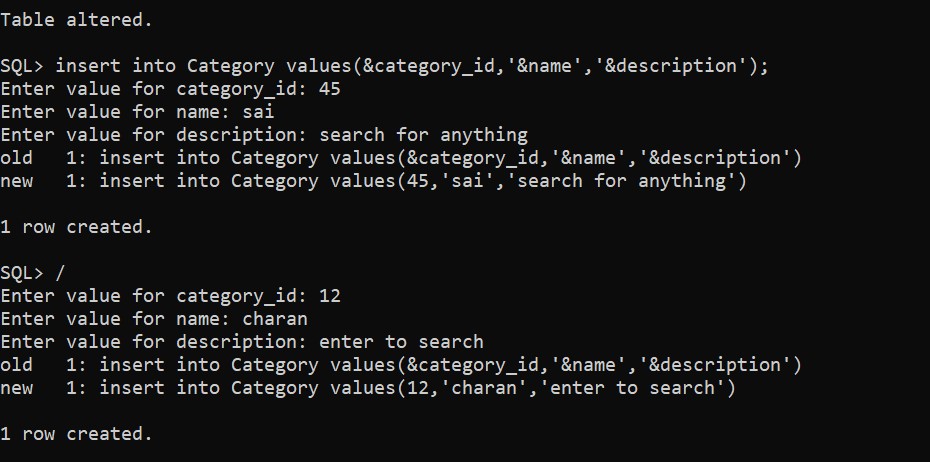
**QUERY:** insert into Users values(&user\_id,'&username','&email','&password');

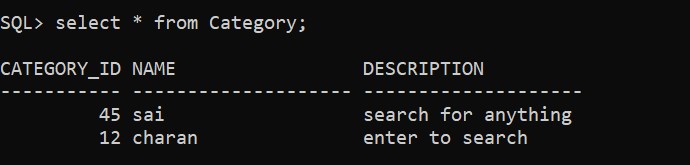




2.Insert values into Category:

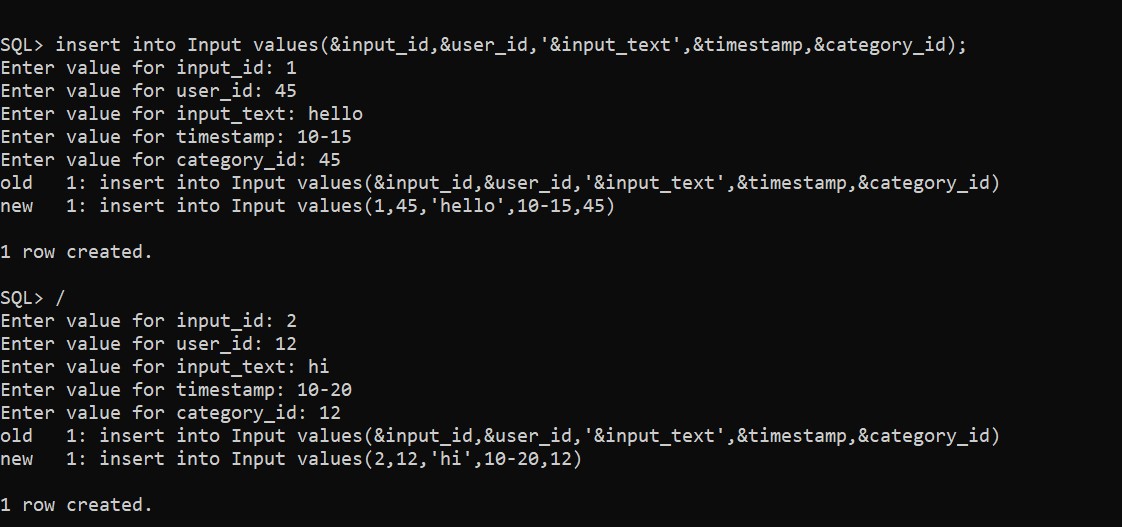
**QUERY:** insert into Category values(&category\_id,'&name','&description');

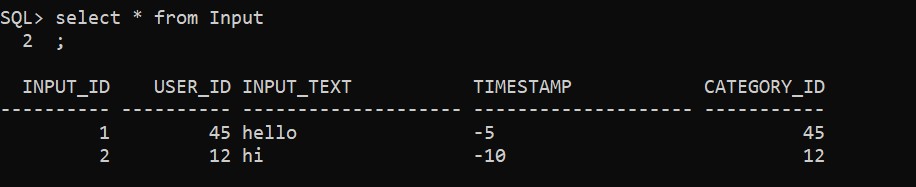




3.Insert values into Input:

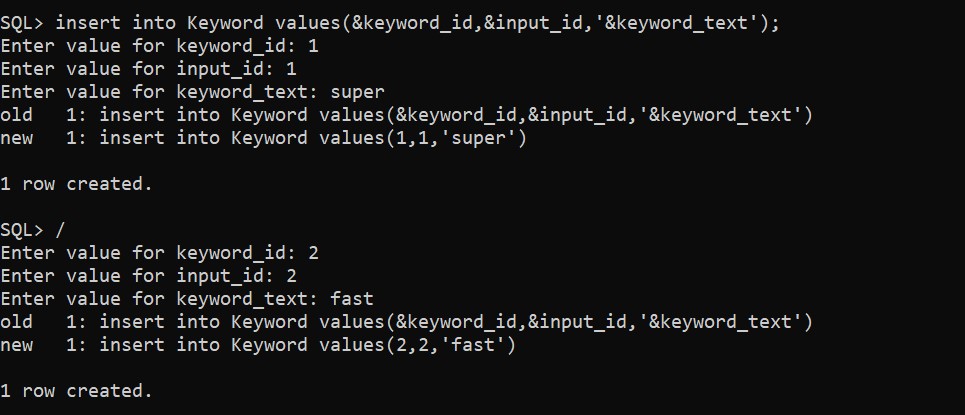
**QUERY**: insert into Input values(&input\_id,&user\_id,'&input\_text',&timestamp,&category\_id);

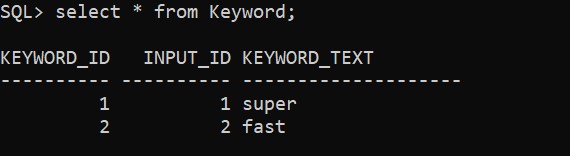




4.Insert values into Keyword:

**QUERY:** insert into Keyword values(&keyword;&input\_id,'&keyword\_text');





5.Insert values into Suggestion:

**QUERY:** insert into Suggestion table(&suggestion\_id,&input\_id,'&suggestion\_text');

