**SQL Queries**

The user table is as follows.,

1. Select \* from user;

Graphical user interface

Description automatically generated

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2. select fname, lname, dob, timestampdiff(year, dob, curdate() )as age from user;

Text

Description automatically generated

// I have derived the age from the date of birth from the user table and so the dob is the derived attribute.,

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

3. select \* from user order by dob;

Graphical user interface, text

Description automatically generated

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

4. select \* from quotes where language = ‘english’;

Table

Description automatically generated

//once the system understands the users taste buds of musical language, the application will start throwing quotes in their appropriate language with few other logical coding’s, and in default it will be displayed in English

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

5. For instance, if our application finds that the listener listens to Japanese songs most of the times, then the related quotes will be displayed in his background once he successfully logs in., and it is as follows.,

Table

Description automatically generated

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

6.// Retrieving the user preference language from weak entity table by referencing the primary key from user table.,

select user.fname, preference.language from user inner join preference on preference.user\_id = user.user\_id;

Text

Description automatically generated

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7. Retrieving the user’s DOB, firstname and login role using cross join between user and login table

select user.fname, user.dob, login.role from user cross join login on login.user\_id = user.user\_id;

Text

Description automatically generated with medium confidence

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8. Retrieving first names of user that starts with “J” using Like function.,

Select fname, name, dob from user where fname like ‘j%’;

Table

Description automatically generated

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

9.Retrieving all users born in 1990’s, it might be helpful in data visualisation.,

Select fname, name, dob from user where dob like ‘199%’;

Table

Description automatically generated with medium confidence

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.Retrieving particular music composer songs that is composed in 21st century.,

Table

Description automatically generated

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NOW let us see some real time hustlings with the query’s in the “pathus” database.,

11. Let’s assume that the user John Vinodh is feeling guilty with some of his previous actions and logs into our web application by accident, just types in his feeling in the textbox provided and below is what all will be displayed to match up his feelings.,

A picture containing diagram

Description automatically generated

//next system tries to find the best solution to match John’s guilt emotion.,

Diagram

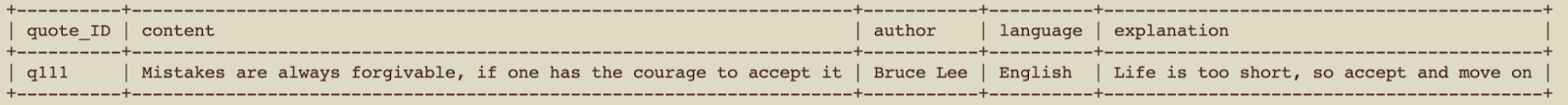
Description automatically generated

//best matching lyrics is also being displayed

Diagram

Description automatically generated

// also corresponding quotes will be displayed



**So, user John after seeing the above database values, somewhat feels normal and better than before and thanks us in return :)**

\*\*\*\*\*\*\*\*\*\*\*\*Thank\_You\*\*\*\*\*\*\*\*\*\*\*\*