DEPARTMENT OF COMPUTER

SCIENCE AND APPLICATIONS

PANJAB UNIVERSITY,

CHANDIGARH

SQL PROJECT

TOPIC: FRAUD DETECTION

USING SQL

SUBMITTED BY: AMISHA SHARMA

COURSE: MCA (SELF FINANCED)-I

ROLL NUMBER: 56

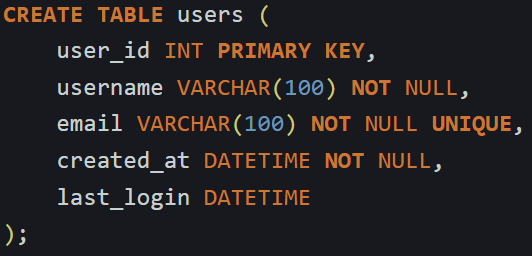
SUBMITTED TO: DR. KAVITA TANEJA

FOR THE TOPIC “FRAUD DETECTION USING SQL” WE WILL USE THREE TABLES – ONE FOR THE USERS NAMED AS ‘USERS’ , SECOND FOR THE TRANSACTIONS MADE BY USERS WHICH IS NAMED AS ‘TRANSACTIONS’ AND LAST FOR THE FRAUD REPORTS REPORTED BY THE USERS WHICH IS NAMED AS ‘FRAUD\_REPORTS’.

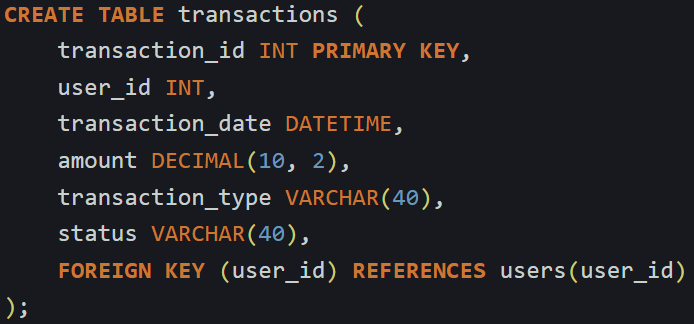
1. CREATE TABLE

HERE, WE WILL CREATE THE REQUIRED THREE TABLES USING CREATE TABLE COMMAND.

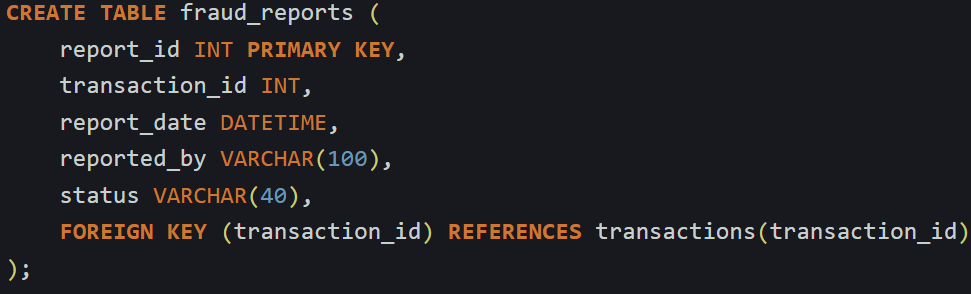
1. USERS TABLE



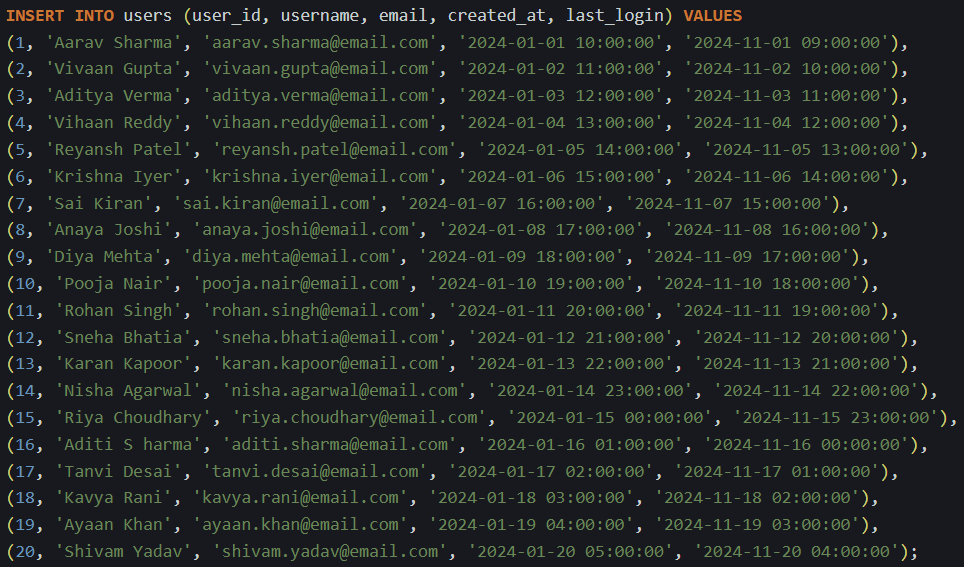
1. TRANSACTIONS TABLE



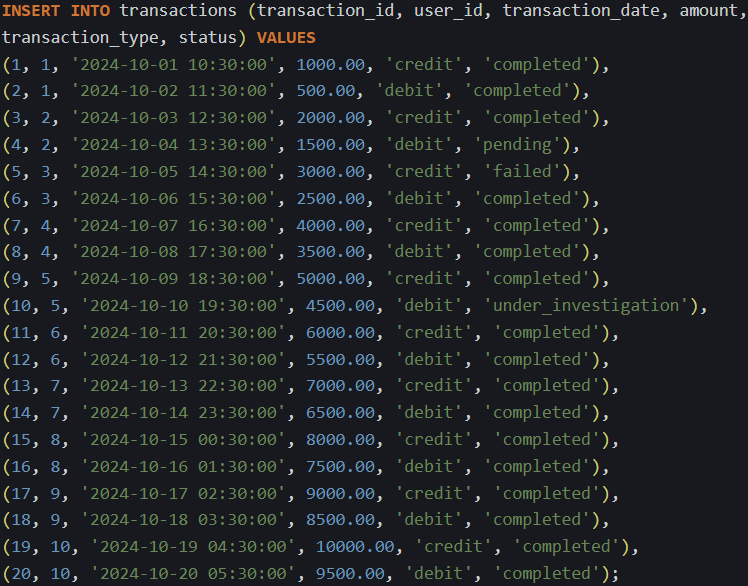
1. FRAUD REPORTS TABLE



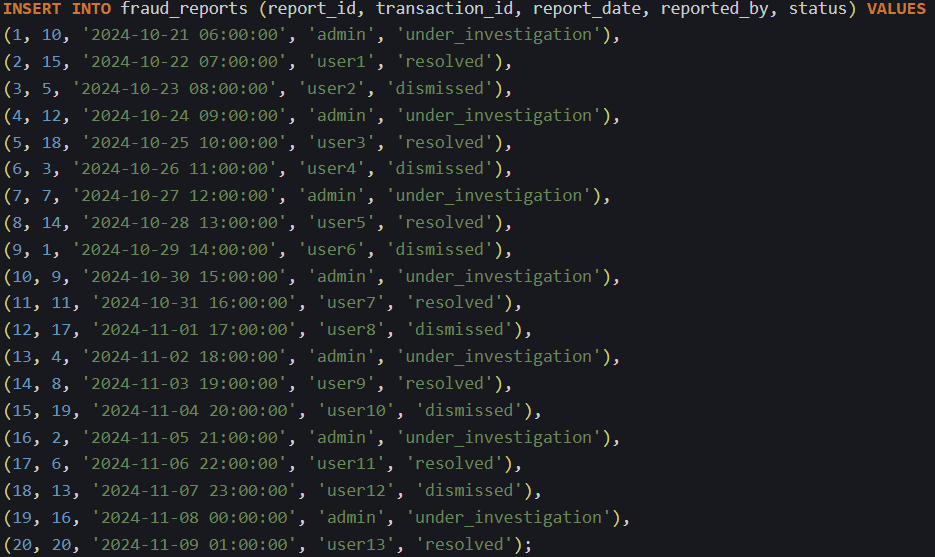
1. INSERTING VALUES INTO TABLES
2. USERS TABLE



1. TRANSACTIONS TABLE



1. FRAUD REPORTS TABLES



1. SELECT COMMAND
2. USERS TABLES



OUTPUT IS:



1. TRANSACTIONS TABLE



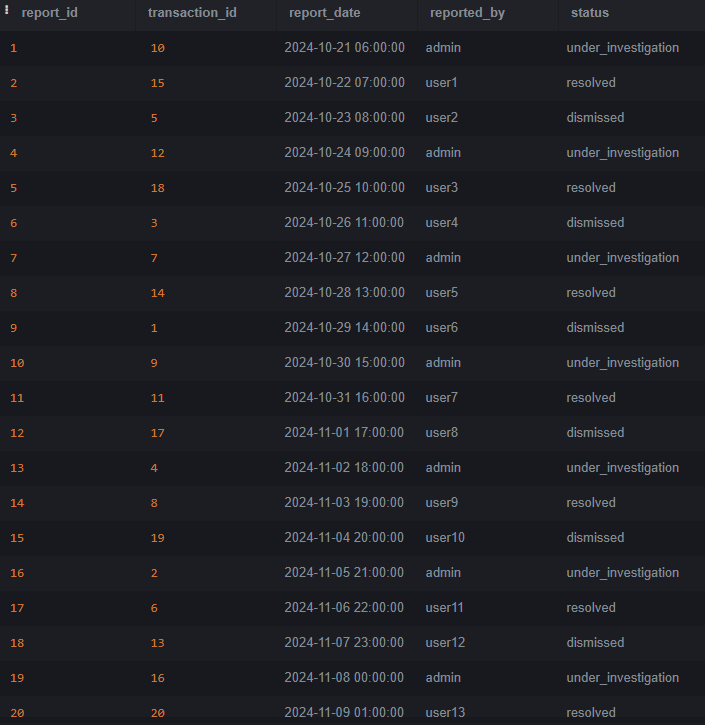
OUTPUT IS:



1. FRAUD REPORTS TABLE



OUTPUT IS:



1. QUERY DATABASE TABLE

IT IS DONE WITH THE HELP OF SELECT COMMAND WHICH GIVES THE USERS WHO HAVE REPORTED FRAUD:

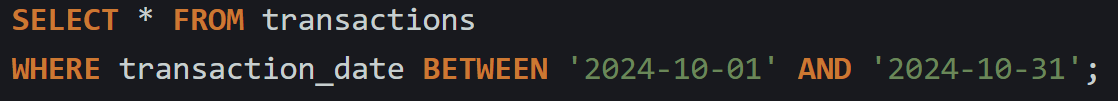


OUTPUT IS:



1. CONDITIONAL RETRIEVEL OF ROWS

IT CAN BE DONE BY EXECUTING SELECT COMMAND ALONG WITH THE WHERE CLAUSE. HERE, WE WILL GET THE TRANSACTIONS BETWEEN TWO DATES AS:



OUTPUT IS:



6. **Queries for Working with NULL Values**

IT IS USED WITH THE HELP OF IS NULL CONDITION AND HERE IT HELPS IN RETRIEVING THE REPORTS THAT ARE NOT REPORTED AS FRAUD BY THE USER WHICH IS AS FOLLOWS:



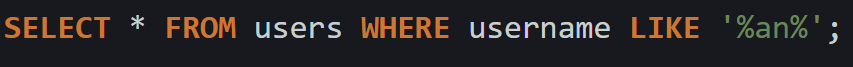
OUTPUT IS:



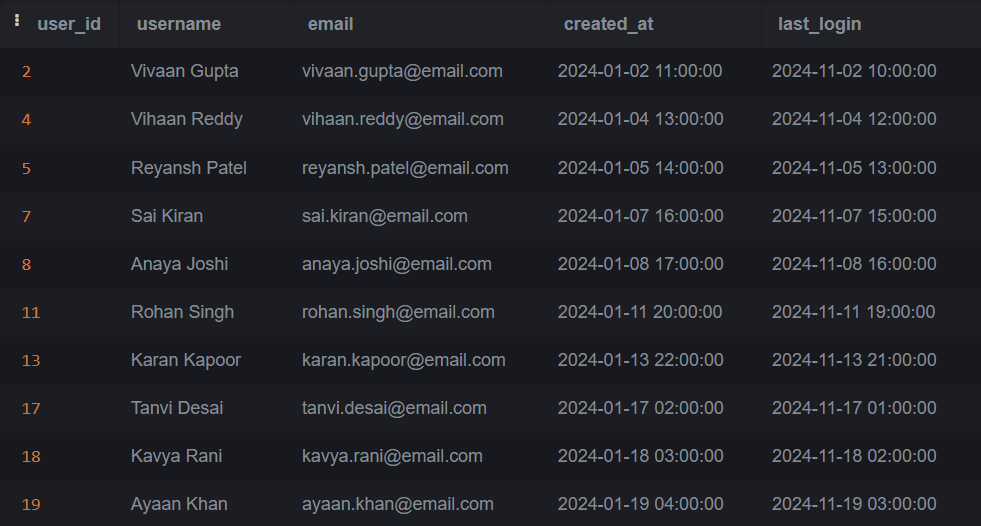
THE OUTPUT GIVES NO RESULT BECAUSE NOT NULL CONSTRAINT IS APPLIED ON THE REPORTED\_BY COLUMN OF THE TABLE.

7. **Queries for Pattern Matching**

IT CAN BE DONE BY USING THE LIKE OPERATOR. HERE, WE WILL GET USERS WITH A USERNAME CONTAINING A SPECIFIC SUBSTRING AS FOLLOWS:



OUTPUT IS:

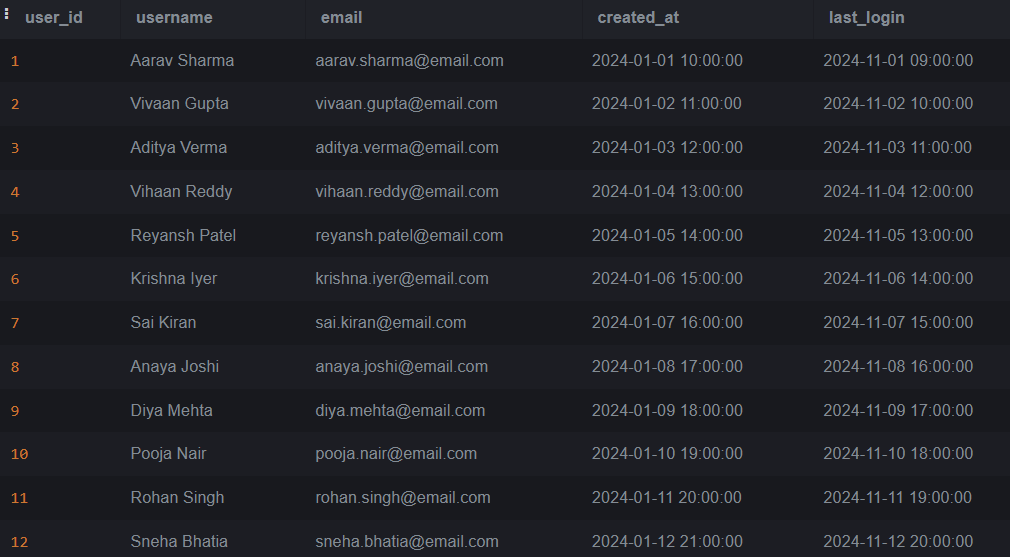


8. **Queries for Ordering Results**

THESE CAN BE ACHIEVED BY USING ORDER BY CLAUSE WHICH HELPS IN SEQUENCING RESULT IN ASCENDING OR DESECENDING ORDER. HERE, WE WILL GET USERS ORDERED BY CREATED DATE AND LAST LOGIN AS FOLLOWS:



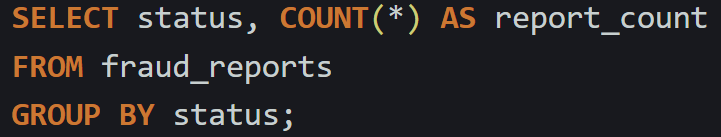
OUTPUT IS:



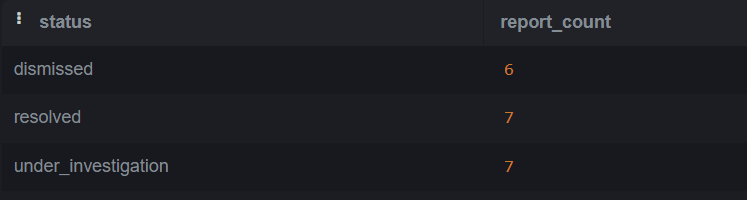
9. **Queries Using Aggregate Functions**

THE AGGREGATE FUNCTIONS COUNT (), SUM (), AVG (), MIN () AND MAX () ARE USED HERE TO GET VARIOUS OUTPUTS AS FOLLOWS:

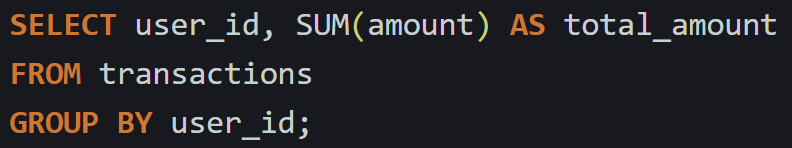
1. COUNT ()



OUTPUT IS:



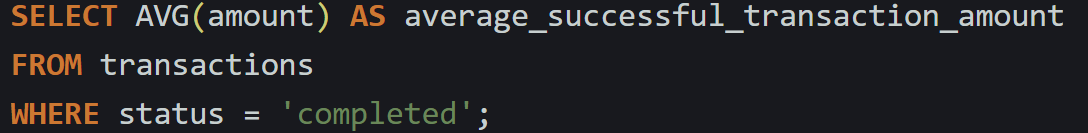
1. SUM ()



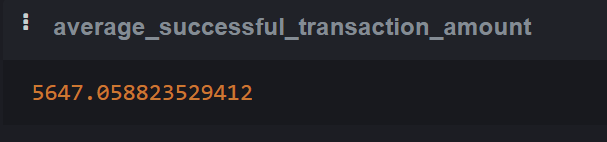
OUTPUT IS:



1. AVG ()



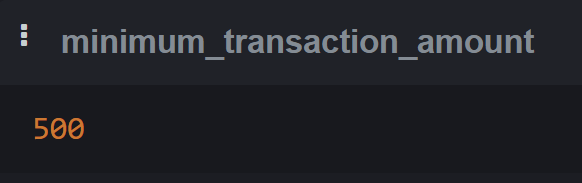
OUTPUT IS:



1. MIN ()



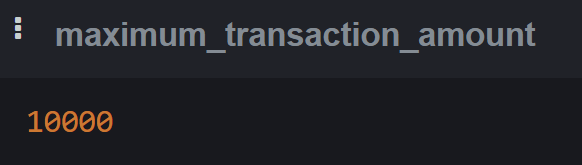
OUTPUT IS:



1. MAX ()

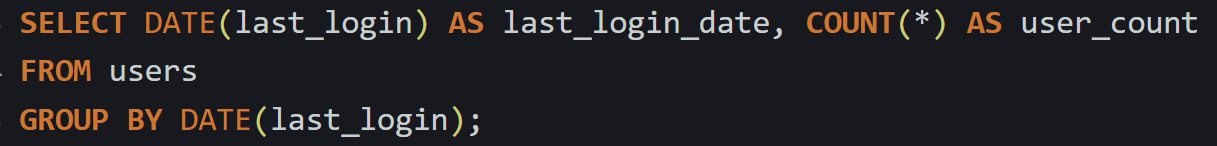


OUTPUT IS:



10. **Queries for Grouping Results**

IN THIS, WE WILL USE GROUP BY CLAUSE WHICH WILL GIVE COUNT OF NUMBER OF USERS WITH EACH LOGGED IN ON EACH DATE. THIS IS AS FOLLOWS:

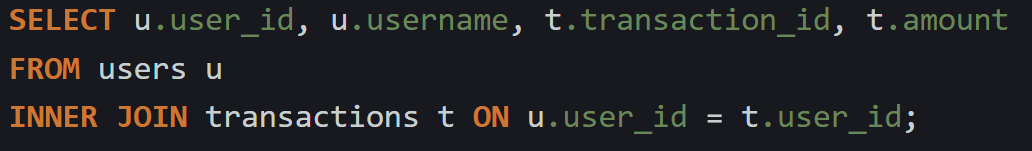


OUTPUT IS:

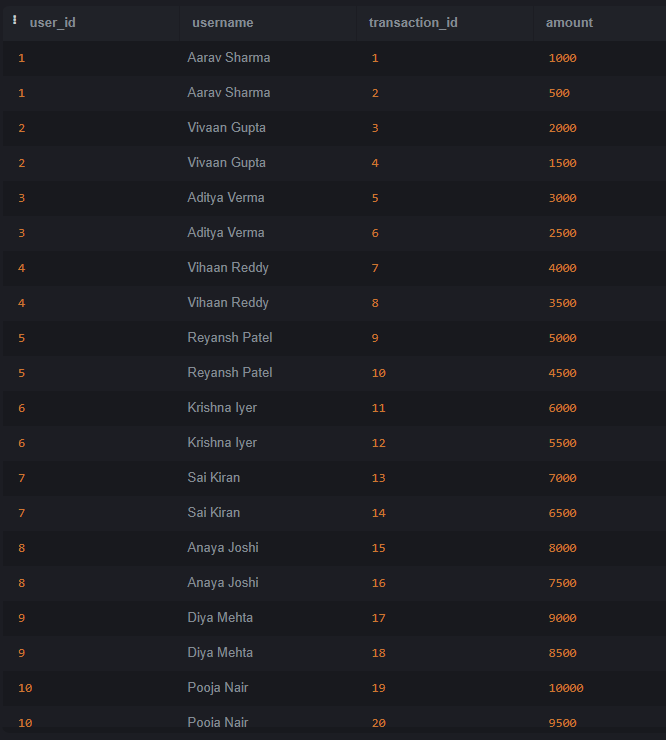


11.**Query for Inner Join**

HERE, WE WILL USE INNER JOIN OPERATION TO RETRIEVE SOME COLUMNS OR FIELDS FROM THE USERS TABLE, WHICH IS AS FOLLOWS:

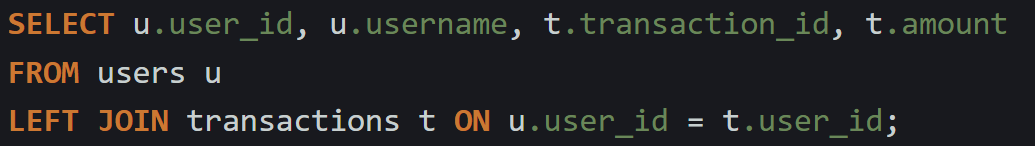


OUTPUT IS:

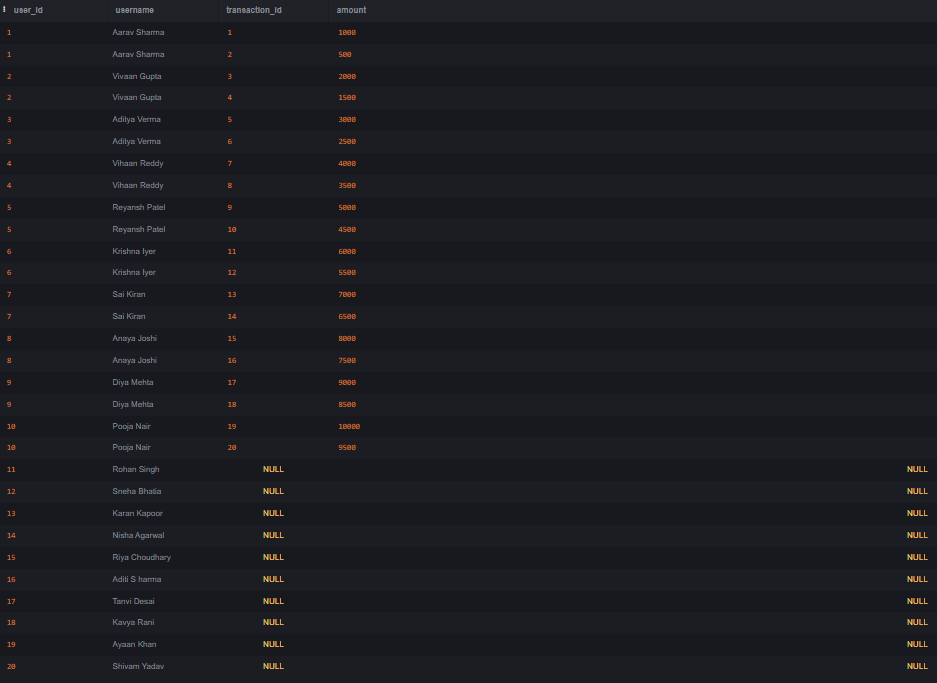
****

12.**Query for Left Join**

HERE, WE ARE USING LEFT JOIN TO RETRIEVE ALL USERS AND THEIR TRANSACTIONS AS FOLLOWS:

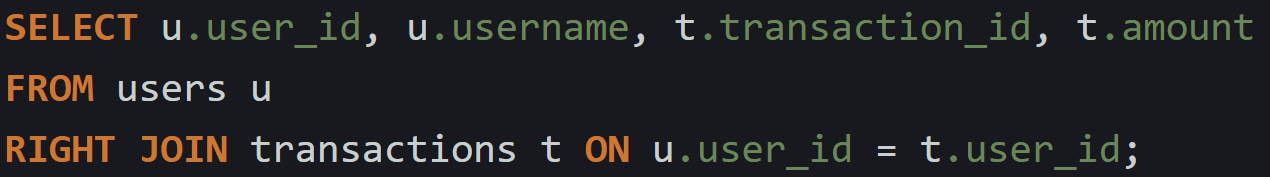


OUTPUT IS:

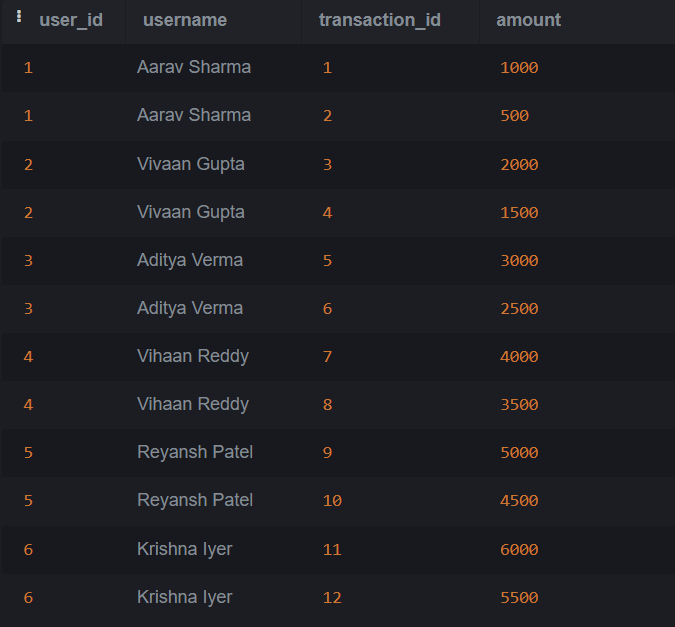


13.**Query for Right Join**

HERE, WE WILL USE RIGHT JOIN OPERATION TO RETRIEVE ALL TRANSACTIONS AND THEIR ASSOCIATED USERS AS FOLLOWS:

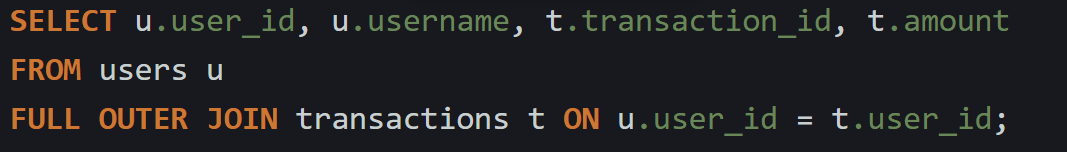


OUTPUT IS:

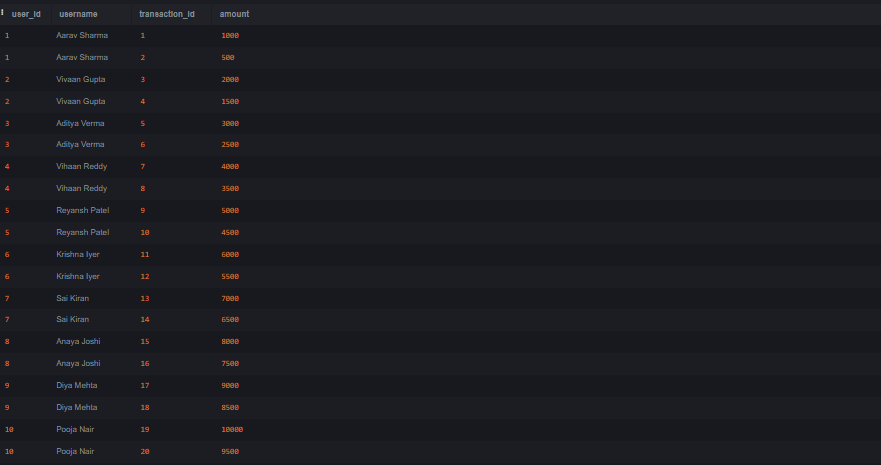


**14. Query for Full Outer Join**

HERE WE WILL USE FULL OUTER JOIN WHICH HELPS IN RETRIEVING ALL THE RECORDS FROM TWO TABLES AS FOLLOWS:

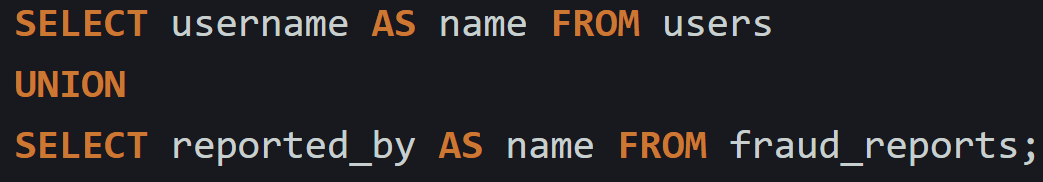


OUTPUT IS:



15.**Query for UNION**

IN THIS CASE, **UNION** operator retrieves a combined list of unique usernames from the **users** table and the names of users who reported fraud from the **fraud\_reports** table AS FOLLOWS:



OUTPUT IS:



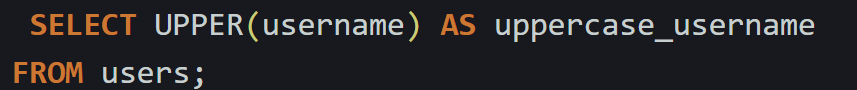




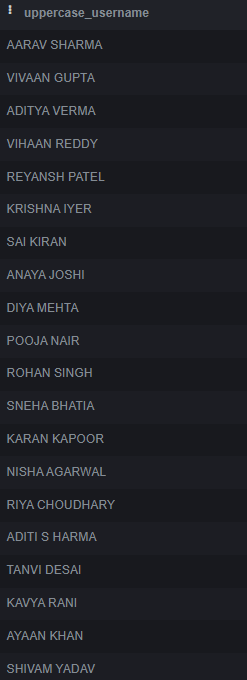


16. **Queries for Character Functions**

IN THIS CASE, CHARACTER FUNCTION IS USED TO GET THE NAMES IN UPPER CASE AS FOLLOWS:

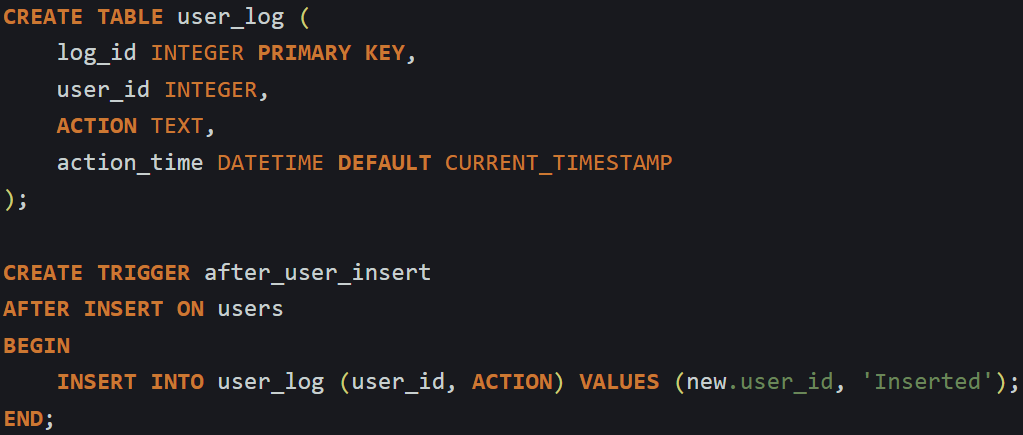


OUTPUT IS:

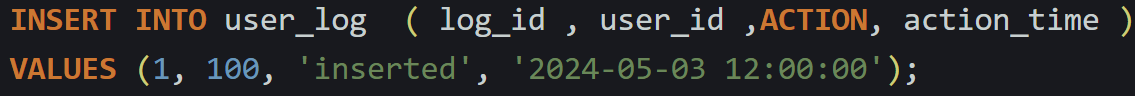


**17. CREATING a trigger**

THIS COMMAND creates a trigger AS FOLLOWS:



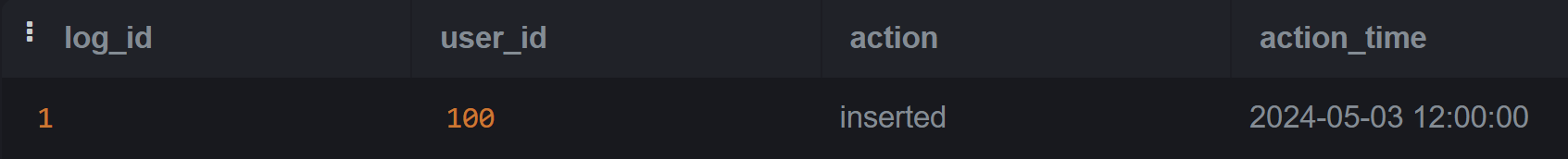
NOW TO ADD SOME DATA IN THE CREATED TRIGGER WE CAN DO THE FOLLOWING STEP:



TO SEE THE RESULT OF THIS OPERATION WE WILL USE SELECT STATEMENT AS:

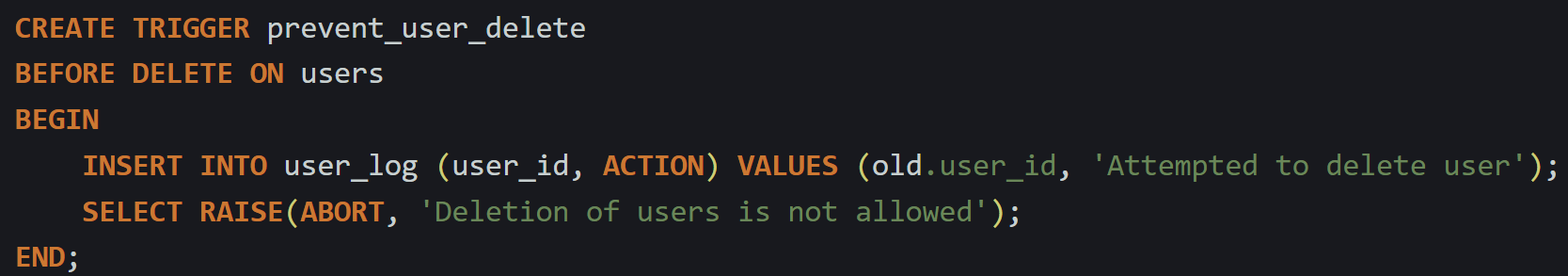


OUTPUT IS:

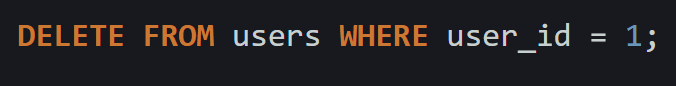


**18.Trigger to Prevent Deletion from USERS Table**

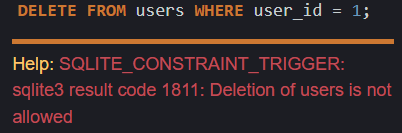
TO DO THIS, WE WILL PERFORM THE FOLLOWING:



NOW, WE WILL CHECK IF THIS COMMAND WORKS AS FOLLOWS:



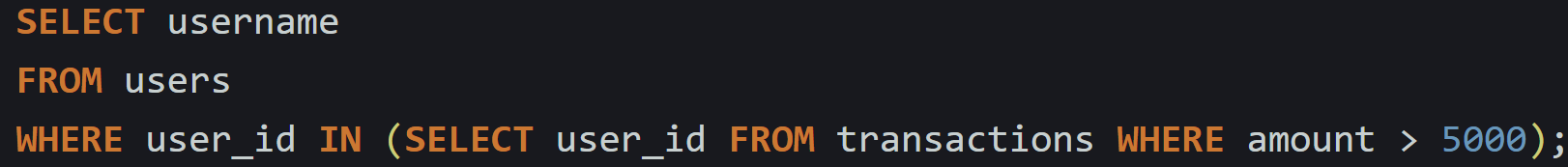
OUTPUT OF THIS WILL BE:



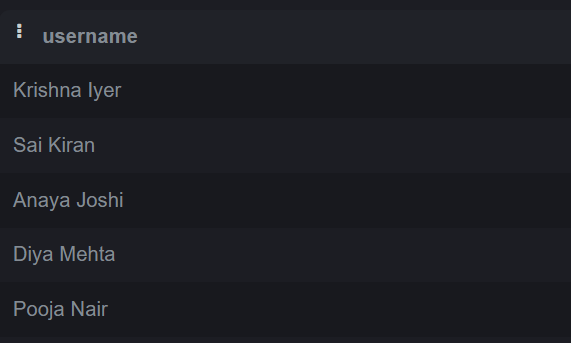
HERE, THE OUTPUT SHOWS THE ERROR WHICH DOES NOT ALLOWS THE DELETION OPERATION FROM USERS TABLE.

19.**Query Using a Subquery**

THIS IS EXECUTED AS FOLLOWS:

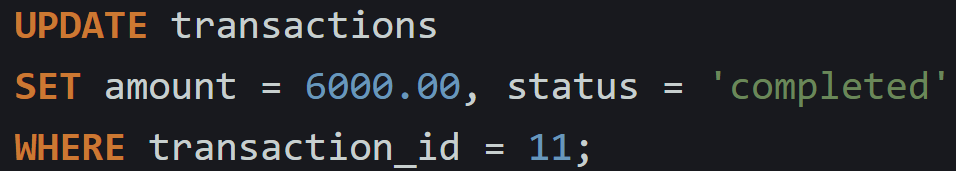


OUTPUT IS:



20.UPDATE QUERY

THIS COMMAND IS IMPLEMENTED AS:



OUTPUT IS:

