**Implementing Security Features - Beta University Annual Fund**

**Step 8.1 – Step 8.4 - Database Security Features for the Team Project.**

For each step:

- Use your Team’s Oracle User ID to run the SQL statements in the database. (Do not use SYS or SYSTEM).

- Write out the SQL statements that will be executed in the database (in addition to showing the SQL Statements in the screenshot).

- Execute the SQL statements in the database. Show your work by providing screenshots that include the current user id, current system date, executing the SQL statements in the database, and the results. Any screenshots that do not include this information will receive no credit. To display the current user and current date, run the following statement before running each SQL statement:

SQL> select user||' '||to\_char(sysdate, 'DD-MON-YYYY HH24:MI:SS') from dual;

**Step 8.1 – Design a value-independent view that hides some sensitive/private information. Indicate (write out) what sensitive/private information you are hiding from the value-independent view. Then create the value-independent view in the database.**

CREATE VIEW PublicDonorInfo AS

SELECT

PD.donorID,

PD.firstName,

PD.lastName,

PD.country,

PD.donorCircle,

MC.matchCorpName,

E.eventName,

E.eventDate,

P.pledgeNumber,

P.pledgeDate,

PM.datePaid,

PM.paymentMethod,

PM.creditCardType

FROM

PotentialDonor PD

LEFT JOIN

Pledge P ON PD.donorID = P.donorId

LEFT JOIN

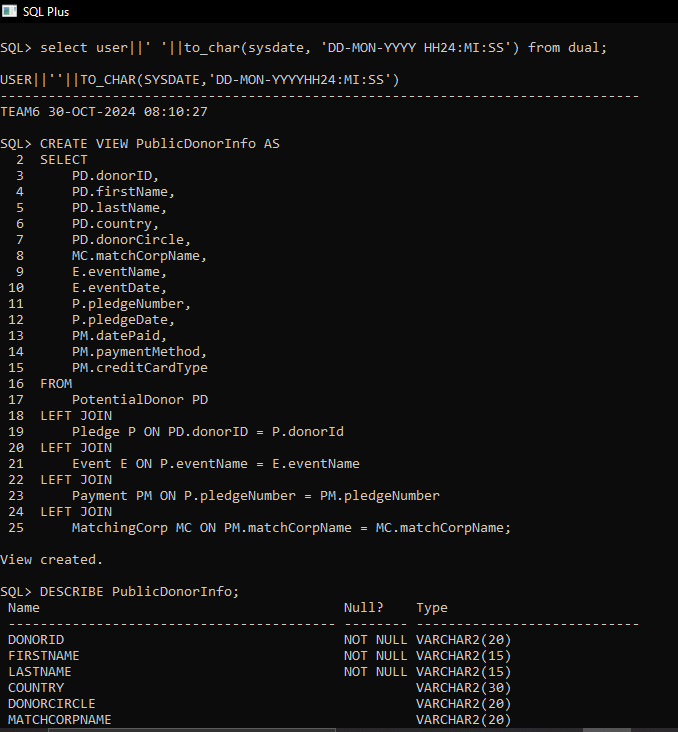
Event E ON P.eventName = E.eventName

LEFT JOIN

Payment PM ON P.pledgeNumber = PM.pledgeNumber

LEFT JOIN

MatchingCorp MC ON PM.matchCorpName = MC.matchCorpName;



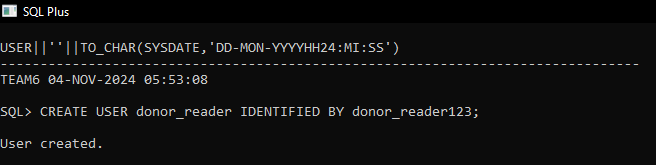
**Step 8.2 - Create a user and authorize that person to read the view. Connect to the database as the new user and demonstrate (provide screenshots) that the new user can read the view. Using Microsoft Visio, draw an authorization graph showing the privileges given.**

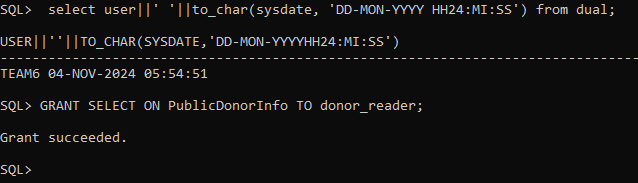
Note: Hand-written authorization graphs are not acceptable submissions.

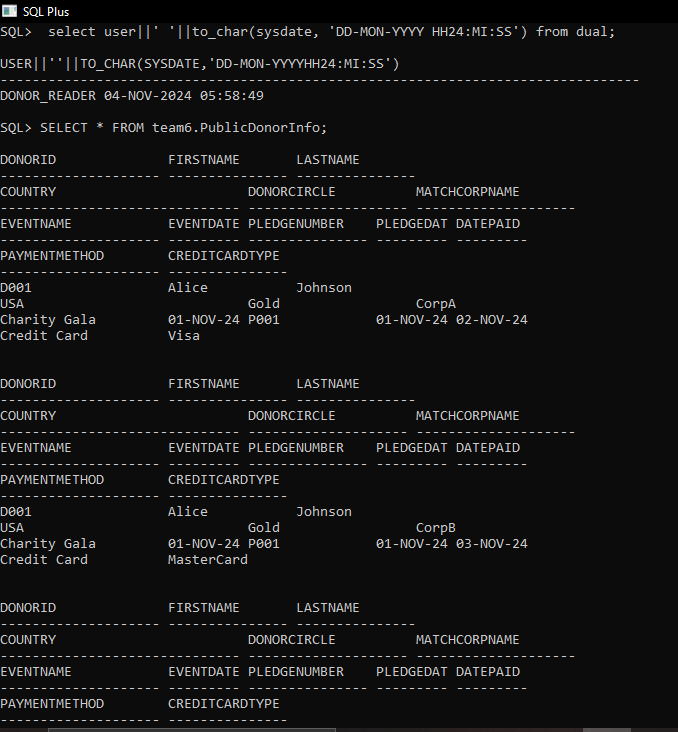
**Answer:**

CREATE USER donor\_reader IDENTIFIED BY donor\_reader123;

GRANT SELECT ON PublicDonorInfo TO donor\_reader;









**Step 8.3 – Create and authorize four other users to access/modify different parts of the database. For each new user, connect to the database and demonstrate (provide screenshots) that each new user can access/modify the different parts of the database that you granted. Using Microsoft Visio, update the authorization graph showing the privileges given.**

Note: Hand-written authorization graphs are not acceptable submissions.

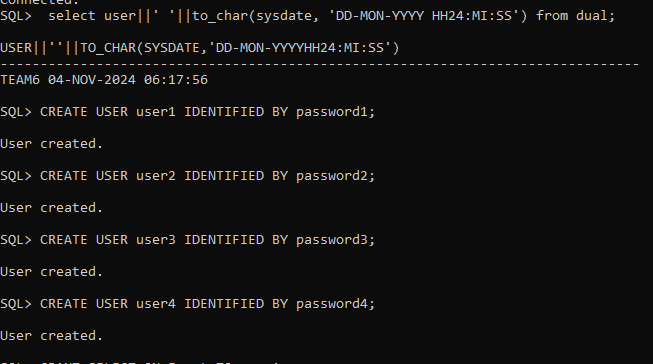
**Answer:**

CREATE USER user1 IDENTIFIED BY password1;

CREATE USER user2 IDENTIFIED BY password2;

CREATE USER user3 IDENTIFIED BY password3;

CREATE USER user4 IDENTIFIED BY password4;

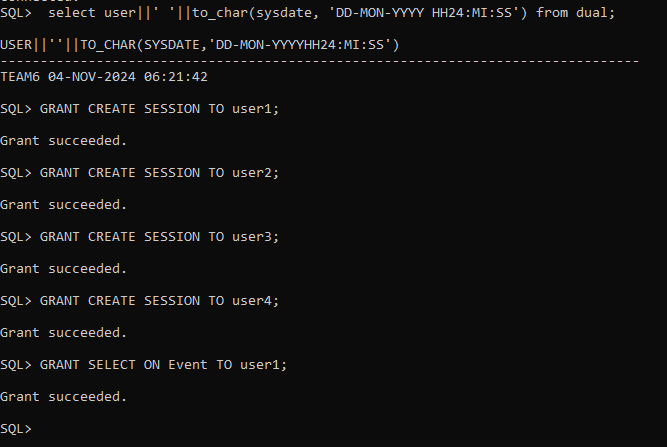


GRANT CREATE SESSION TO user1;

GRANT CREATE SESSION TO user2;

GRANT CREATE SESSION TO user3;

GRANT CREATE SESSION TO user4;

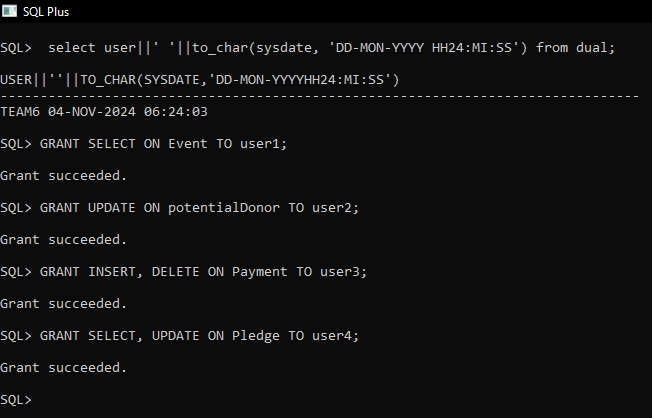


GRANT SELECT ON Event TO user1;

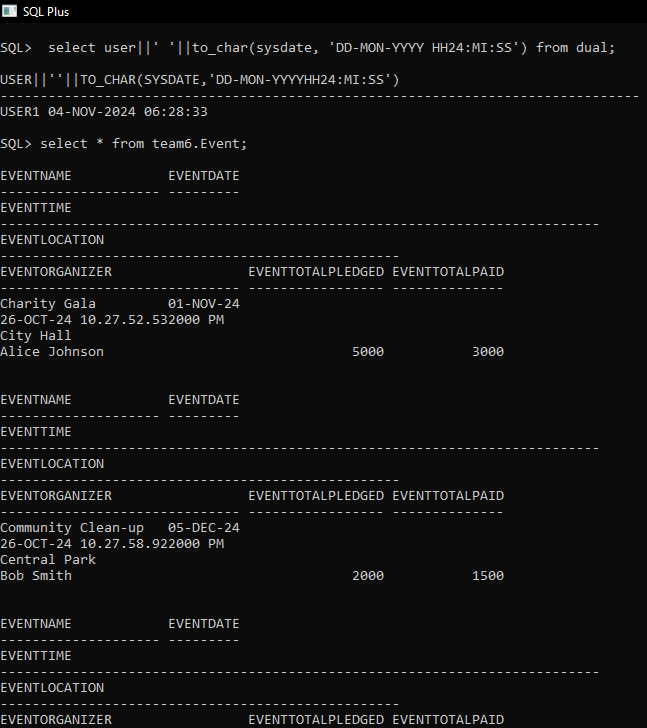
GRANT SELECT, INSERT, DELETE, UPDATE ON team6.PotentialDonor TO user2;

GRANT SELECT, INSERT, DELETE, UPDATE ON team6.Payment TO user3;

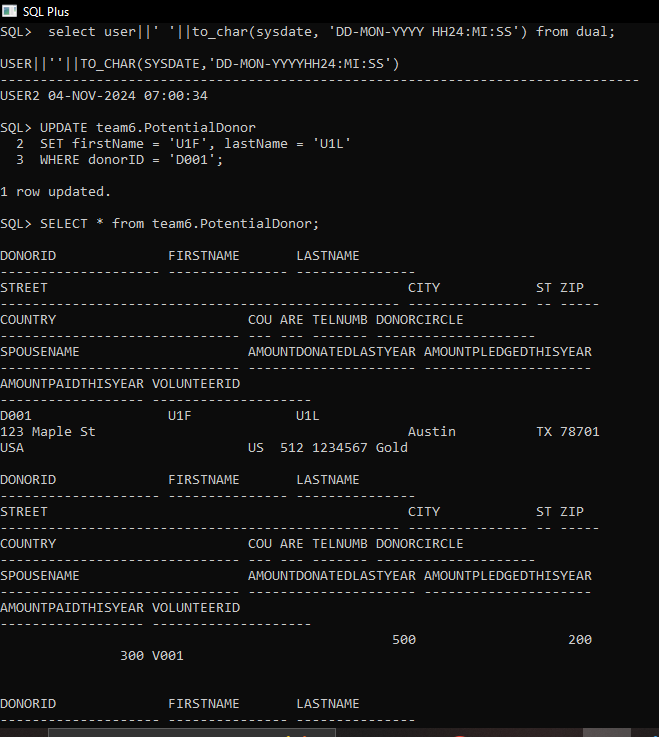
GRANT SELECT, INSERT, DELETE, UPDATE ON team6.Pledge TO user4;



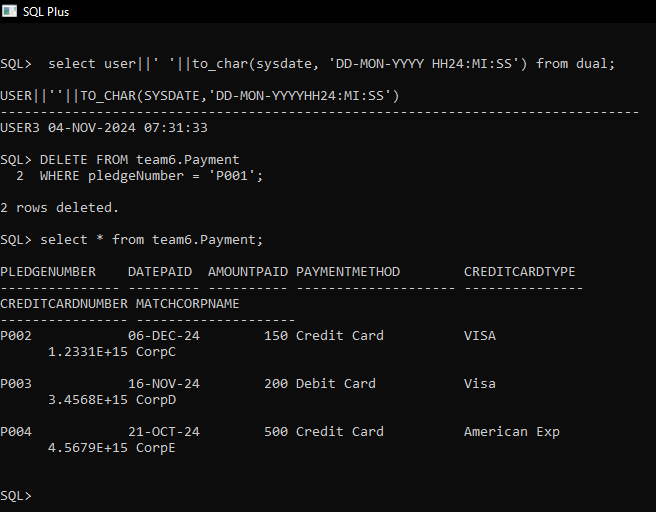
**User1:** GRANT SELECT ON Event TO user1;



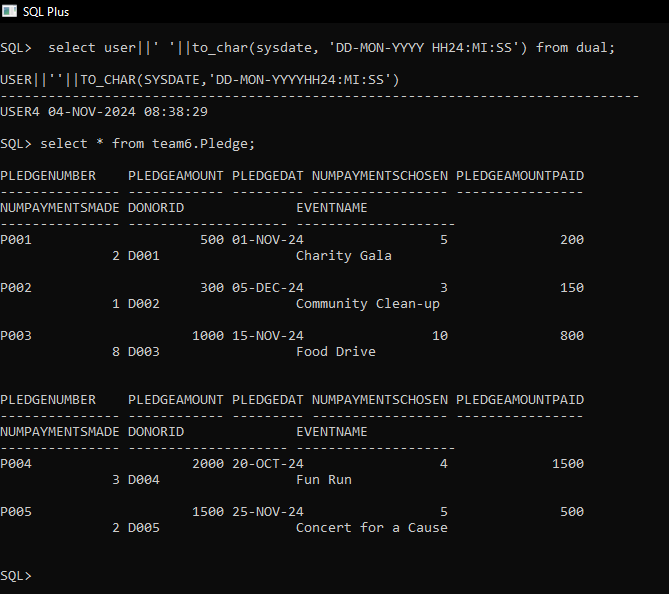
**user2:** GRANT SELECT, INSERT, DELETE, UPDATE ON potentialDonor TO user2;



**user3:** GRANT SELECT, INSERT, DELETE, UPDATE ON team6.Payment TO user3;



**user4:** GRANT SELECT, INSERT, DELETE, UPDATE ON team6.Pledge TO user4;





**Step 8.4 - Design an audit trail table and an audit trail trigger for updates to a sensitive/private item. Then create the audit trail table and audit trail trigger in the database.**

**Then design and execute SQL statements to demonstrate that the trigger is working as expected.** To demonstrate that the trigger is working as expected, provide a screenshot of the data before and after the trigger is executed.

**Answer:**

CREATE TABLE AuditTrail (

AuditID NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

TableName VARCHAR2(50),

OperationType VARCHAR2(10),

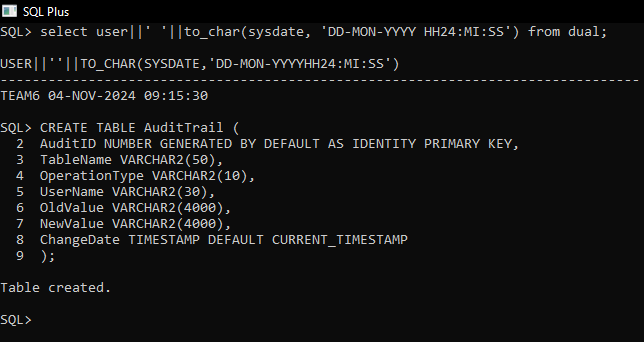
UserName VARCHAR2(30),

OldValue VARCHAR2(4000),

NewValue VARCHAR2(4000),

ChangeDate TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);



CREATE OR REPLACE TRIGGER trg\_Audit\_PotentialDonor

AFTER UPDATE ON team6.PotentialDonor

FOR EACH ROW

BEGIN

INSERT INTO AuditTrail (TableName, OperationType, UserName, OldValue, NewValue)

VALUES (

'PotentialDonor',

'UPDATE',

USER,

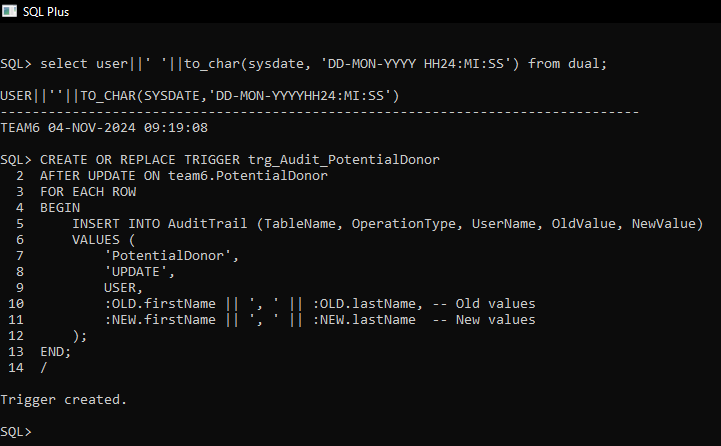
:OLD.firstName || ', ' || :OLD.lastName, -- Old values

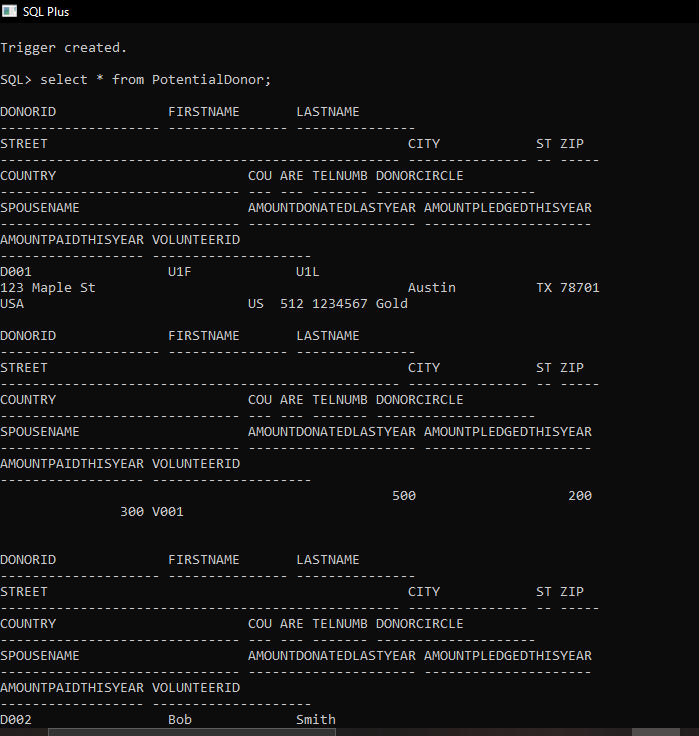
:NEW.firstName || ', ' || :NEW.lastName -- New values

);

END;

/





UPDATE team6.PotentialDonor

SET firstName = 'UpdatedName',

lastName = 'UpdatedLastName'

WHERE donorID = 'D001';

