



DACM

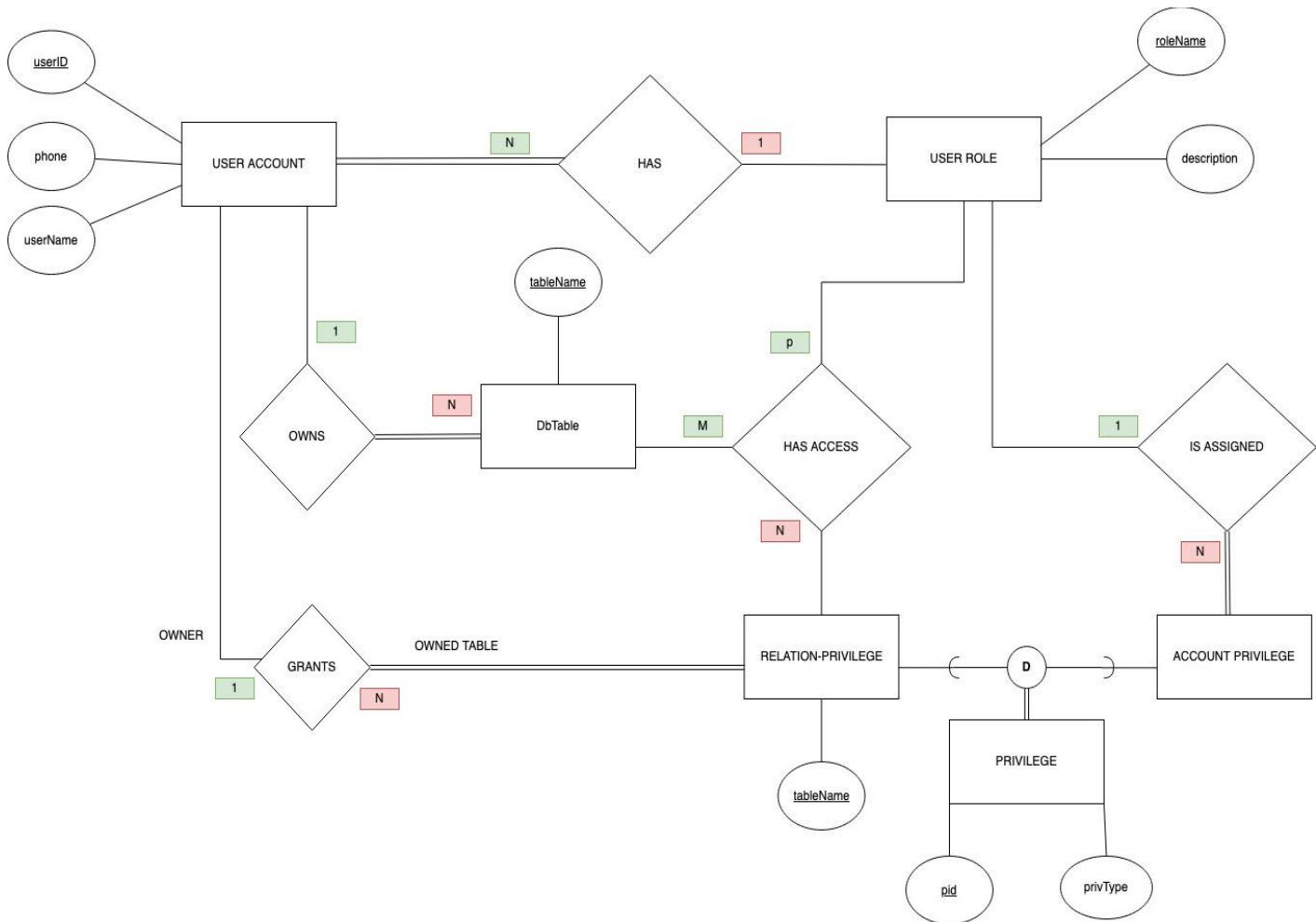
Discretionary Access Control

CSE5330 Project 2

Project Report

Nikhil Sujith | 1001768092
Abhishek Shinde | 1001754842

Part 1 – EER Diagram



Basic Assumptions

1. The user account has userID which is the primary key to identify the user account.
2. The user role has roleName as the primary key to uniquely identify each role in the system.
3. DbTable has tableName as the primary key to identify each unique table in the system.
4. Privilege is disjoint to 2 types.
5. User Account can only be assigned to 1 user role.
6. Multiple user accounts can be assigned to a role.
7. If a role is given a privilege, then all users assigned to that role will have that privilege.

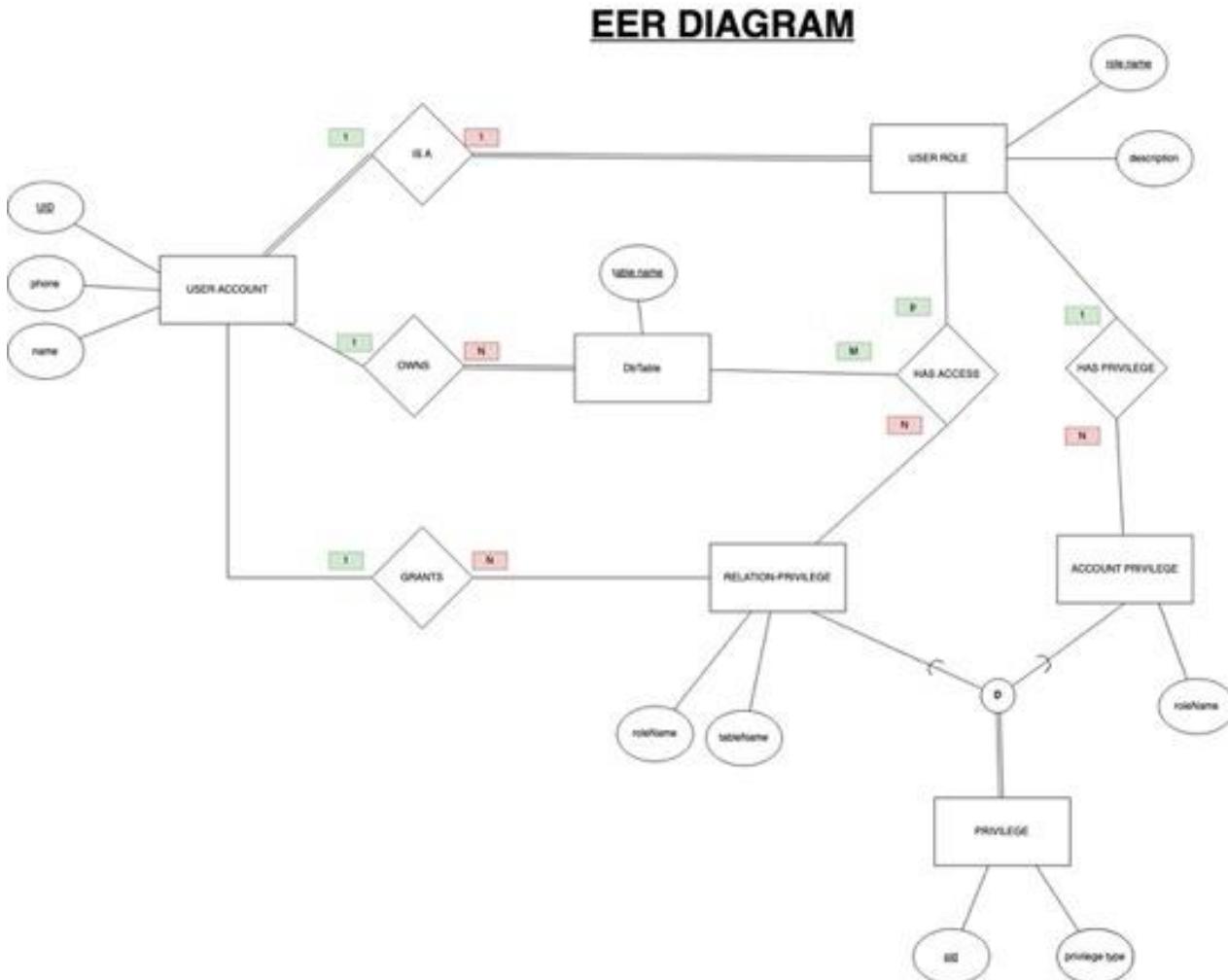
-
- 8. An user account can own multiple tables
 - 9. A table can only be owned by a single user account
 - 10. An user role can be assigned multiple account privileges
 - 11. An user role can have multiple relation privileges on multiple tables
 - 12. A table can be granted multiple privileges on multiple roles
 - 13. A relation privilege on a table can be granted to multiple roles.

Additional Assumptions

- 14. An user may own a table and that user will have all privileges on the table they own.
- 15. The owner of the table can then allow certain privileges on the table they own.
- 16. The owner of the table can then allow certain roles to access certain privileges on the table they own.
- 17. The owner of the table can revoke privileges that they might have assigned to a role on the owned table.
- 18. The owner of the table can grant privileges to roles on the owned table, if the role also has been allowed to perform those operations by the super-user.
- 19. Each role will be assigned certain privileges on them, and the role can only perform those operations as permitted by the privileges.
- 20. The system has been implemented to follow “Strict DAC” where there can only be a single level of access grants, where the owner can grant access on owned relation to another user / role but the other user / role will not be allowed any propagation of privileges.
- 21. A privilege can only be granted and revoked by the owner of the table.

Changes made to EER

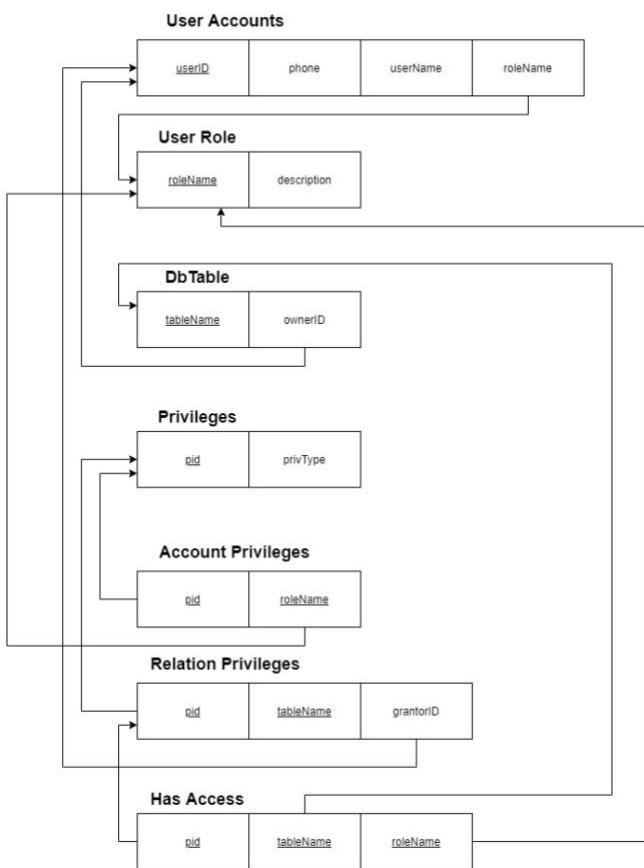
Initial EER



1. The mapping for user accounts and user role has been changed from 1:1 to N : 1. The reason for this change being, initially the assumption was that each role will have a single user. However, during the mapping stage we realized that
2. The participation for user accounts and relation privilege has been update to total participation on the relations privilege side. This is for the fact that relation privileges has to be given by an owner who is also represented as a user account in the database.
3. Some attributes have been adjusted to match the requirements and assumptions.
4. The participation for user the user role side of has relationship has been changes from total to partial. This was because, we found it more practical for the user role to be optionally assigned user accounts. This arises from a scenario where the system actor[person using the system] may decide to first only create role and later assign users as and when they see fit.

Part 2 – EER to Relational Mapping

Relational Schema



Mapping of regular (strong) entity types

In this step, we map the strong entities in our project, I.e. User Account, User Role, DBTable and Privilege , by creating a Relation with their simple attributes present in it, and select one of the attributes as a Primary Key.

Mapping of Binary Relationships

I) Mapping of 1:N Relationship:

1:N Relationship for User Account and DBTable:

We map this relationship by adding a Foreign Key in DBTable which references the Primary Key of User Accounts.

Here, the Foreign key is `ownerID` which references `userID` in User Accounts.

1:N Relationship for User Role and User Account :

We map this relationship by adding a Foreign Key in User Account which references the Primary Key of User Role.

Here, the Foreign Key is roleName which references roleName in User Role.

1:N Relationship for User Account and Relation-Privilege :

We map this relationship by adding a Foreign Key in Relation-Privilege which references the Primary Key of User Account.

Here, the Foreign Key is grantorID which references userID in User Role.

1:N Relationship for User Role and Account Privilege :

We map this relationship by adding a Foreign Key in Account -Privilege which references the Primary Key of User Role.

Here, the Foreign Key is roleName which references roleName in User Role.

Mapping of Ternary Relationships

For mapping a ternary relation where $n > 2$, we create a new relationship relation to represent the ternary relation. For our EER Mapping, we have a Ternary Relation between User Role, DBTable and Relation Privilege. To represent this relation in a Relational Schema, we create a new relationship relation “Has Access” which contains as Foreign Key, the primary keys of all the participating entities.

That is, Has Access has foreign key :

1. tableName – which references the primary key of DBTable
2. roleName – which references the primary key of User Role
3. pid – which references the primary key of Relation Privilege

Mapping of Specialization of Classes

For mapping of class specializations, we create a new relation for the Super Class with its simple attributes and its primary key. For the subclasses, we also create a new relation where the subclasses have the primary of their parent class. For our EER Diagram, Privilege has a specialization where Relation Privilege and Account Privilege are its sub classes. We Create a relation Privilege with pid as the primary key, and we then create relation Account Privilege and Relation Privilege which have as foreign key, pid, which refers to the primary key of parent class.

DDL

```

CREATE DATABASE SECURITY7;

USE SECURITY7;

CREATE TABLE user_accounts(
    userID INT NOT NULL auto_increment,
    phone varchar(12) NOT NULL DEFAULT 'NULL',
    userName varchar(100) NOT NULL UNIQUE ,
#    userRole varchar(100) NOT NULL DEFAULT 'staff',
    CONSTRAINT UAPK PRIMARY KEY (userID)
)ENGINE=InnoDB;

CREATE TABLE DbTable(
    tableName varchar(100) NOT NULL UNIQUE ,
    ownerID int NOT NULL,
    CONSTRAINT tpk PRIMARY KEY (tableName),
    CONSTRAINT tfk FOREIGN KEY (ownerID) REFERENCES user_accounts(userID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
)ENGINE=InnoDB;

CREATE TABLE privileges(
    pid int NOT NULL ,
    privType varchar(100) NOT NULL,
    CONSTRAINT ppk PRIMARY KEY (pid),
    UNIQUE (privType)
)ENGINE=InnoDB;

CREATE TABLE user_role(
    roleName varchar(100) NOT NULL,
    description varchar(100),
    userID int NOT NULL DEFAULT 12,
    CONSTRAINT urpk PRIMARY KEY (roleName, userID),
    CONSTRAINT urfk FOREIGN KEY (userID) REFERENCES user_accounts(userID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
)ENGINE=InnoDB;

CREATE TABLE account_privileges(
    pid int NOT NULL ,
    roleName varchar(100) NOT NULL ,
    CONSTRAINT appk PRIMARY KEY (pid, roleName),
    CONSTRAINT apfk1 FOREIGN KEY (pid) REFERENCES privileges(pid)
    ON DELETE CASCADE
    ON UPDATE CASCADE ,
    CONSTRAINT apfk2 FOREIGN KEY (roleName) REFERENCES user_role(roleName)
)

```

```
ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=InnoDB;

CREATE TABLE relation_privileges (
    pid int NOT NULL ,
    tableName varchar(100) NOT NULL ,
    grantorID int NOT NULL ,
    CONSTRAINT rppk PRIMARY KEY (pid,tableName),
    CONSTRAINT rpfk1 FOREIGN KEY (pid) REFERENCES privileges (pid)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT rpfk2 FOREIGN KEY (grantorID) REFERENCES
user_accounts(userID)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT rpfk3 FOREIGN KEY (tableName) REFERENCES DbTable(tableName)
    ON DELETE CASCADE
    ON UPDATE CASCADE
) ENGINE=InnoDB;

CREATE TABLE has_access (
    pid int NOT NULL,
    tableName varchar(100) NOT NULL ,
    roleName varchar(100) NOT NULL ,
    grantorID int NOT NULL ,
    CONSTRAINT hspk PRIMARY KEY (pid, roleName, tableName),
    CONSTRAINT hsfk1 FOREIGN KEY (pid) REFERENCES relation_privileges
(pid)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT hsfk2 FOREIGN KEY (tableName) REFERENCES
relation_privileges (tableName)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT hsfk3 FOREIGN KEY (grantorID) REFERENCES
relation_privileges (grantorID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
) ENGINE=InnoDB;
```

Part3 Question2 Answers

```
mysql> show tables;
+-----+
| Tables_in_SECURITY7 |
+-----+
| DbTable
| account_privileges
| all_user_priv
| has_access
| privileges
| relation_privileges
| role_priv
| user_accounts
| user_priv_table
| user_role
+-----+
10 rows in set (0.09 sec)
```

```
mysql> #Account Privileges
```

```
mysql> SELECT pid AS PRIVILEGE_ID, roleName AS ROLE_NAME
-> FROM SECURITY.account_privileges;
+-----+-----+
| PRIVILEGE_ID | ROLE_NAME |
+-----+-----+
| 100 | HR
| 101 | HR
| 102 | HR
| 103 | HR
| 103 | manager
| 103 | staff
| 104 | HR
| 104 | manager
| 105 | HR
| 105 | manager
| 106 | HR
| 106 | manager
| 106 | staff
+-----+-----+
13 rows in set (0.02 sec)
```

```
mysql>
```

```
mysql> #Tables
```

```
mysql> SELECT tableName AS TABLE_NAME, ownerID AS TABLE_OWNER_ID
-> FROM SECURITY.DbTable;
+-----+-----+
| TABLE_NAME | TABLE_OWNER_ID |
+-----+-----+
| hrTable | 1 |
| managerTable | 2 |
| staffTable | 3 |
+-----+-----+
3 rows in set (0.02 sec)
```

```
mysql>
```

```

mysql> #Has Access
mysql> SELECT pid AS PRIVILEGE_ID, roleName AS ROLE_NAME, grantorID AS
GRANTOR_ID, tableName AS TABLE_NAME
-> FROM SECURITY7.has_access;
+-----+-----+-----+-----+
| PRIVILEGE_ID | ROLE_NAME | GRANTOR_ID | TABLE_NAME |
+-----+-----+-----+-----+
|          106 | hr        |          1 | hrTable    |
+-----+-----+-----+-----+
1 row in set (0.02 sec)

mysql>
mysql> #Privileges
mysql> SELECT pid AS PRIVILEGE_ID, privType AS PRIVILEGE_NAME
-> FROM SECURITY.privileges;
+-----+-----+
| PRIVILEGE_ID | PRIVILEGE_NAME |
+-----+-----+
|          100 | create      |
|          101 | drop        |
|          102 | alter       |
|          103 | insert      |
|          104 | delete      |
|          105 | update      |
|          106 | select      |
+-----+-----+
7 rows in set (0.02 sec)

mysql>
mysql>
mysql> #Relation Privileges
mysql> SELECT pid as Privilege_ID, tableName as Table_Name, grantorID as
Owner_ID FROM SECURITY7.relation_privileges;
+-----+-----+
| Privilege_ID | Table_Name | Owner_ID |
+-----+-----+
|          106 | hrTable   |          1 |
|          106 | managerTable |          2 |
|          103 | staffTable |          3 |
|          106 | staffTable |          3 |
+-----+-----+
4 rows in set (0.04 sec)

mysql>
mysql> #User Accounts
mysql> SELECT userID as User_ID, phone as Phone_Number, userName as
User_Name FROM SECURITY7.user_accounts;
+-----+-----+-----+
| User_ID | Phone_Number | User_Name |
+-----+-----+-----+
|          1 | 201-932-1111 | J.Jane    |
|          2 | 201-922-1234 | J.Doe     |
|          3 | 201-922-4321 | A.Jack    |
|          4 | 901-123-1234 | B.Wong    |
+-----+-----+-----+

```

```

|      5 | 901-321-4321 | R.Ravi
|      6 | 842-879-0987 | Q.David
|      7 | 842-849-4562 | B.Wayne
|      8 | 456-224-1957 | H.Park
|      9 | 682-926-1502 | O.Oscar
|     10 | 272-592-0184 | W.Man
|     11 | 837-238-0281 | S.Guy
|     12 | XXX-XXX-XXXX | Default-User-For-Role
+-----+-----+-----+
12 rows in set (0.03 sec)

```

```

mysql>
mysql> #User Roles
mysql> SELECT roleName as Role_Name, description as Description_Of_Role,
userID as User_ID FROM SECURITY7.user_role;
+-----+-----+-----+
| Role_Name | Description_Of_Role | User_ID |
+-----+-----+-----+
| clerk     | clerk role           |      3 |
| clerk     | clerk role           |      5 |
| hr        | human resource       |      1 |
| manager   | manager              |      2 |
| manager   | manager              |      6 |
| manager   | manager              |      7 |
| staff     | staffing             |      4 |
| staff     | staffing             |      8 |
| staff     | staffing             |      9 |
| staff     | staffing             |     10 |
| staff     | staffing             |     11 |
+-----+-----+-----+
11 rows in set (0.02 sec)

```

```
mysql> notee
```

Screenshots for Part 3

CREATE NEW USER

The screenshot shows a web-based application titled "Discretionary Access Control Management". On the left, a sidebar menu lists various database management tasks under the "DADM" heading. The main area is titled "Create New User". Under "User Settings", there are two input fields: "Username" (containing "Name") and "Phone Number" (containing "XXX-XXX-XXXX"). A blue button labeled "Execute Query" is positioned below these fields. To the right, a table titled "Existing User Details" displays 10 rows of user information, each with columns for "User ID", "Phone", and "User Name". A green banner at the top of the page indicates "User Added!". The system status bar at the bottom right shows the time as 3:49 PM and the date as 11/22/2020.

This screenshot shows the same DADM application interface. In the "User Settings" section, the "Username" field contains "Name" and the "Phone Number" field contains "XXX-XXX-XXXX". When the "Execute Query" button is clicked, a red banner at the top displays the error message "User Exists". The rest of the interface and data table remain the same as the previous screenshot, showing the existing user list and the system status bar at the bottom right.

CREATE NEW ROLE

The screenshot shows a web browser window titled "Create New User" with the URL "localhost/dbms/SecurityDB/Forms/Code/html/new_role.php". The main content area is titled "Discretionary Access Control Management" and contains a "Create New Role" form. On the left, there is a sidebar with various DACM-related menu items. The "User Settings" section of the form includes fields for "Role Name" (set to "newrole") and "Description" (set to "This is a new role"). To the right, a table titled "Existing Roles" lists several pre-existing roles with their descriptions. The status bar at the bottom indicates the time as 3:50 PM and the date as 11/22/2020.

Role Name	Description
clerk	clerk role
clerk	clerk role
hr	human resource
manager	manager
manager	manager
manager	manager
staff	staffing

Role Added

Role Name	Description
clerk	clerk role
clerk	clerk role
hr	human resource
manager	manager
manager	manager
manager	manager
newrole	This is a new role.
staff	staffing
staff	staffing
staff	staffing

S Create New User + localhost/dbms/SecurityDB/Forms/Code/html/new_role.php

Apps (90) YouTube Project AcgnX Torre... Just a moment... Mail - abhishek.sh... StarRez Portal - Stu... TimePro® Employee... Introduction to Py... Solve Programming...

DACM Discretionary Access Control Management

Create New Role

User Settings

Role Name: newrole
Description: This is a new role.

Existing Roles

Role Name	Description
clerk	clerk role
clerk	clerk role
hr	human resource
manager	manager
manager	manager
manager	manager
newrole	This is a new role.
staff	staffing

Execute Query

3:53 PM 11/22/2020

S Create New User + localhost/dbms/SecurityDB/Forms/Code/html/new_role.php

Apps (90) YouTube Project AcgnX Torre... Just a moment... Mail - abhishek.sh... StarRez Portal - Stu... TimePro® Employee... Introduction to Py... Solve Programming...

DACM Discretionary Access Control Management

Create New Role

Role Already Exists

User Settings

Role Name: newrole
Description: This is a new role.

Existing Roles

Role Name	Description
clerk	clerk role
clerk	clerk role
hr	human resource
manager	manager
manager	manager
manager	manager
newrole	This is a new role.
staff	staffing
staff	staffing
staff	staffing

Execute Query

3:53 PM 11/22/2020

CREATE NEW TABLE

Create New Table

localhost/dbms/SecurityDB/Forms/Code/html/create_new_table.php

Discretionary Access Control Management

User Settings

Table Name: newTable

Owner ID: 2

Existing Table Details

Table Name	Owner ID
clerkTable	1
hrTable	1
managerTable	2
staffTable	3
employeeTable	4
projectTable	5
locationTable	6
supervisorTable	7
departmentTable	8
dependentTable	9

Execute Query

Create New Table

localhost/dbms/SecurityDB/Forms/Code/html/create_new_table.php

Discretionary Access Control Management

Table Added!

User Settings

Table Name: newTable

Owner ID: 2

Existing Table Details

Table Name	Owner ID
clerkTable	1
hrTable	1
managerTable	2
newTable	2
staffTable	3
employeeTable	4
projectTable	5
locationTable	6
supervisorTable	7
departmentTable	8

Execute Query

Create New Table

localhost/dbms/SecurityDB/Forms/Code/html/create_new_table.php

DACM

Discretionary Access Control Management

Create New Table

User Settings

Table Name: newTable
Owner ID: 3

Existing Table Details

Table Name	Owner ID
clerkTable	1
hrTable	1
managerTable	2
newTable	2
staffTable	3
employeeTable	4
projectTable	5
locationTable	6
supervisorTable	7
departmentTable	8
dependentTable	9

Execute Query

3:54 PM 11/22/2020

Create New Table

localhost/dbms/SecurityDB/Forms/Code/html/create_new_table.php

DACM

Discretionary Access Control Management

Create New Table

Table Exists

User Settings

Table Name:
Owner ID:

Existing Table Details

Table Name	Owner ID
clerkTable	1
hrTable	1
managerTable	2
newTable	2
staffTable	3
employeeTable	4
projectTable	5
locationTable	6
supervisorTable	7
departmentTable	8

Execute Query

3:54 PM 11/22/2020

ADD NEW PRIVILEGE

The screenshot shows a web browser window titled "Create New User" on the address bar, with the URL localhost/dbms/SecurityDB/Forms/Code/html/add_privilege.php. The main content area is titled "Discretionary Access Control Management". On the left, there is a sidebar with various navigation links. The main form is titled "Add A New Privilege". It has two sections: "User Settings" and "Existing Privileges". In "User Settings", the "Privilege ID" field contains "107" and the "Privilege Type" field contains "reference". Below these fields is a blue "Execute Query" button. To the right is a table titled "Existing Privileges" with columns "Privilege ID" and "Privilege Type". The table lists entries from 100 to 106, each with its corresponding privilege type. At the bottom of the page, a green banner displays the message "Privilege Added!".

Privilege ID	Privilege Type
100	create
101	drop
102	alter
103	insert
104	delete
105	update
106	select

Screenshot of a web application titled "Discretionary Access Control Management". The page shows a form to "Add A New Privilege" and a table of existing privileges.

User Settings

- Privilege ID: 107
- Privilege Type: reference

Existing Privileges

Privilege ID	Privilege Type
100	create
101	drop
102	alter
103	insert
104	delete
105	update
106	select
107	reference

Screenshot of the same web application showing an error message: "Privilege Type Already Exists".

User Settings

- Privilege ID: (empty)
- Privilege Type: (empty)

Existing Privileges

Privilege ID	Privilege Type
100	create
101	drop
102	alter
103	insert
104	delete
105	update
106	select
107	reference

RELATE USER ACCOUNT TO ROLE

Screenshot of a web application titled "Discretionary Access Control Management". The application is running on a local host at port 80.

The main interface shows a sidebar with various administrative tasks:

- Create New User
- Create New Role
- Create New Table
- Insert New Privilege
- Relate User Account to Role
- Relate Acc Priv to Role
- Add Relation Privilege
- Relate Relation Priv to Role and Table
- Revoke Privilege of User From Table
- Retrieve Role Privileges
- Retrieve User Privileges
- Check Privilege for User
- Access Table as User

The current page is "Assign Role to a User". It has two sections: "User Settings" and "Existing Role and Assigned Users".

In "User Settings", the "Role Name" field contains "hr" and the "User ID" field contains "13". A blue button labeled "Execute Query" is present.

The "Existing Role and Assigned Users" section displays a table:

Role Name	Description	User Account ID
clerk	clerk role	3
clerk	clerk role	5
hr	human resource	1
manager	manager	2
manager	manager	6
manager	manager	7
newrole	This is a new role.	12
staff	staffing	4
staff	staffing	8
staff	staffing	9
staff	staffing	10

Below the table, a green message box says "User Has Been Assigned a Role".

The taskbar at the bottom shows several open applications, including a browser window for "Assign Roles to User Account" and other system icons like File Explorer, Task View, and Start.

S Assign Roles to User Account +

localhost/dbms/SecurityDB/Forms/Code/html/relate_account_to_role.php

Apps (90) YouTube Project AcgnX Torre... Just a moment... Mail - abhishekshi... StarRez Portal - Stu... TimePro® Employee... Introduction to Py... Solve Programming...

DACM Discretionary Access Control Management

Assign Role to a User

User Settings

Role Name: hr

User ID: 13

Existing Role and Assigned Users

Role Name	Description	User Account ID
clerk	clerk role	3
clerk	clerk role	5
hr	human resource	1
hr	human resource	13
manager	manager	2
manager	manager	6
manager	manager	7
newrole	This is a new role.	12
staff	staffing	4
staff	staffing	8
staff	staffing	9

Execute Query

3:56 PM 11/22/2020

S Assign Roles to User Account +

localhost/dbms/SecurityDB/Forms/Code/html/relate_account_to_role.php

Apps (90) YouTube Project AcgnX Torre... Just a moment... Mail - abhishekshi... StarRez Portal - Stu... TimePro® Employee... Introduction to Py... Solve Programming...

DACM Discretionary Access Control Management

User Does Not Exist / User Already Assigned

Assign Role to a User

User Settings

Role Name:

User ID:

Existing Role and Assigned Users

Role Name	Description	User Account ID
clerk	clerk role	3
clerk	clerk role	5
hr	human resource	1
hr	human resource	13
manager	manager	2
manager	manager	6
manager	manager	7
newrole	This is a new role.	12
staff	staffing	4
staff	staffing	8

Execute Query

3:56 PM 11/22/2020

RELATE ACCOUNT PRIVILEGE TO ROLE

Screenshot of a web application titled "Discretionary Access Control Management". The application is used for managing account privileges. It shows a sidebar with various administrative tasks and two main sections: "User Settings" and "Existing Role and Account Privileges".

User Settings:

- Privilege ID: 106
- Role Name: newrole
- Buttons: Execute Query

Existing Role and Account Privileges:

Role Name	Privilege Type	Privilege ID
hr	create	100
hr	drop	101
hr	alter	102
hr	insert	103
hr	delete	104
hr	update	105
hr	select	106
manager	alter	102
manager	insert	103
manager	delete	104
manager	update	105

The second screenshot shows the same interface after a privilege has been granted. A green header bar indicates "Privilege Granted to Role".

Screenshot of a web application window titled "Assign Privileges to Role". The URL is "localhost/dbms/SecurityDB/Forms/Code/html/relate_acc_priv_to_role.php".

The left sidebar contains a list of database management tasks:

- Create New Note
- Create New Table
- Insert New Privilege
- Relate User Account to Role
- Relate Acc Priv to Role
- Add Relation Privilege
- Relate Relation Priv to Role and Table
- Revoke Privilege of User From Table
- Retrieve Role Privileges
- Retrieve User Privileges
- Check Privilege for User
- Access Table as User

The main area has two sections: "User Settings" and "Existing Role and Account Privileges".

"User Settings" section:

- Privilege ID:
- Role Name:
-

"Existing Role and Account Privileges" table:

Role Name	Privilege Type	Privilege ID
hr	create	100
hr	drop	101
hr	alter	102
hr	insert	103
hr	delete	104
hr	update	105
hr	select	106
manager	alter	102
manager	insert	103
manager	delete	104
manager	update	105
newrole	select	106
staff	insert	103
staff	select	106

Screenshot of the same web application window, showing the results of an executed query.

The "User Settings" section shows the input values from the previous screenshot:

- Privilege ID:
- Role Name:
-

The "Existing Role and Account Privileges" table remains the same as in the first screenshot.

System status bar at the bottom right: 3:56 PM, ENG, 11/22/2020.

S Assign Privileges to Role +

localhost/dbms/SecurityDB/Forms/Code/html/relate_acc_priv_to_role.php

Apps (90) YouTube Project AcgnX Torre... Just a moment... Mail - abhishek.shi... StarRez Portal - Stu... TimePro® Employee... Introduction to Py... Solve Programming... ↗

DACM

Discretionary Access Control Management

Privilege Already Granted to Role / Wrong Privilege ID

Adding Account Privileges To A Role

User Settings Existing Role and Account Privileges

Role Name	Privilege Type	Privilege ID
hr	create	100
hr	drop	101
hr	alter	102
hr	insert	103
hr	delete	104
hr	update	105
hr	select	106
manager	alter	102
manager	insert	103
manager	delete	104

Privilege ID:
Role Name:

3:57 PM 11/22/2020

ADD RELATION PRIVILEGE FOR A TABLE

Screenshot of a web application titled "Discretionary Access Control Management" showing the "Add Relation Privilege for A Table" page.

The User Settings section contains:

- As Owner With ID: 1
- of Table: hrTable
- Allow Privilege with ID: 104

The Existing Privileges on Table section displays the following table:

Privilege ID	Table Name	Owner ID
106	hrTable	1
106	managerTable	2
103	staffTable	3
106	staffTable	3

The Execute Query button is present below the table.

The second screenshot shows the same interface after the privilege has been added. A green success message at the top states "New Privilege Added On Table".

The User Settings section now shows:

- As Owner With ID: 104
- of Table: hrTable
- Allow Privilege with ID: 106

The Existing Privileges on Table section displays the following table:

Privilege ID	Table Name	Owner ID
104	hrTable	1
106	hrTable	1
106	managerTable	2
103	staffTable	3
106	staffTable	3

The Execute Query button is present below the table.

S Allow Privilege on Table +

localhost/dbms/SecurityDB/Forms/Code/html/add-relation-privilege.php

Apps (90) YouTube Project AcgnX Torre... Just a moment... Mail - abhishek.sh... StarRez Portal - Stu... TimePro® Employee... Introduction to Py... Solve Programming... ▾

DACM Discretionary Access Control Management

Add Relation Privilege for A Table

User Settings Existing Privileges on Table

As Owner With ID
1

of Table
hrTable

Allow Privilege with ID
104

Execute Query

Privilege ID	Table Name	Owner ID
104	hrTable	1
106	hrTable	1
106	managerTable	2
103	staffTable	3
106	staffTable	3

3:58 PM 11/22/2020

S Allow Privilege on Table +

localhost/dbms/SecurityDB/Forms/Code/html/add-relation-privilege.php

Apps (90) YouTube Project AcgnX Torre... Just a moment... Mail - abhishek.sh... StarRez Portal - Stu... TimePro® Employee... Introduction to Py... Solve Programming... ▾

DACM Discretionary Access Control Management

Add Relation Privilege for A Table

User Settings Existing Privileges on Table

As Owner With ID

of Table

Allow Privilege with ID

Execute Query

Privilege ID	Table Name	Owner ID
104	hrTable	1
106	hrTable	1
106	managerTable	2
103	staffTable	3
106	staffTable	3

Privilege Already Allowed on Table / Wrong Privilege Entered

3:58 PM 11/22/2020

RELATE RELATION PRIVILEGE TO ROLE ON TABLE

The screenshot shows two instances of a web browser displaying the Discretionary Access Control Management (DACM) application. The URL is localhost/dbms/SecurityDB/Forms/Code/html/relate-ternary.php.

User Settings:

- As Owner With ID: 1
- of Table: hrTable
- Grant Privilege with ID: 106
- To Role: newrole

Existing Privileges on Table:

Grantor ID	Table Name	Privilege ID	Role Name
1	hrTable	106	hr

Privilege to Role Granted on Entered Table:

Grantor ID	Table Name	Privilege ID	Role Name
1	hrTable	106	hr
1	hrTable	106	newrole

Relate Ternary

localhost/dbms/SecurityDB/Forms/Code/html/relate-ternary.php

DACM

Discretionary Access Control Management

Relate Account Privilege to Role and Table

User Settings

Existing Privileges on Table

Grantor ID	Table Name	Privilege ID	Role Name
1	hrTable	106	hr
1	hrTable	106	newrole

As Owner With ID: 2

of Table: hrTable

Grant Privilege with ID: 106

To Role: newrole

3:59 PM 11/22/2020

Relate Ternary

localhost/dbms/SecurityDB/Forms/Code/html/relate-ternary.php

DACM

Discretionary Access Control Management

Access Denied: Entered Owner ID is not the owner of the table

Relate Account Privilege to Role and Table

User Settings

Existing Privileges on Table

Grantor ID	Table Name	Privilege ID	Role Name
1	hrTable	106	hr
1	hrTable	106	newrole

As Owner With ID:

of Table:

Grant Privilege with ID:

To Role:

3:59 PM 11/22/2020

Relate Ternary

localhost/dbms/SecurityDB/Forms/Code/html/relate-ternary.php

DACM

Discretionary Access Control Management

Relate Account Privilege to Role and Table

User Settings

Existing Privileges on Table

Grantor ID	Table Name	Privilege ID	Role Name
1	hrTable	106	hr
1	hrTable	106	newrole

As Owner With ID: 1

of Table: hrTable

Grant Privilege with ID: 103

To Role: newrole

Execute Query

3:59 PM 11/22/2020

Relate Ternary

localhost/dbms/SecurityDB/Forms/Code/html/relate-ternary.php

DACM

Discretionary Access Control Management

This Privilege Does Not Exist For Entered Table.

Relate Account Privilege to Role and Table

User Settings

Existing Privileges on Table

Grantor ID	Table Name	Privilege ID	Role Name
1	hrTable	106	hr
1	hrTable	106	newrole

As Owner With ID:

of Table:

Grant Privilege with ID:

To Role:

Execute Query

3:59 PM 11/22/2020

REVOKE PRIVILEGES FOR A USER ON TABLE

Screenshot of a web-based Discretionary Access Control Management (DACM) application showing the process of revoking privileges for a user on a table.

The application interface includes a sidebar with various DACM-related functions:

- Create New User
- Create New Role
- Create New Table
- Insert New Privilege
- Relate User Account to Role
- Relate Acc Priv to Role
- Add Relation Privilege
- Relate Relation Priv to Role and Table
- Revoke Privilege of User From Table
- Retrieve Role Privileges
- Retrieve User Privileges
- Check Privilege for User
- Access Table as User

The main page displays the "Revoke Privilege for a User on A Table" form and a table showing existing privileges.

User Settings:

- As Owner With ID: 1
- of Table: hrTable
- Revoke Privilege with ID: 106
- To Role: newrole

Existing Privileges on Table:

Grantor ID	Table Name	Privilege ID	Role Name
1	hrTable	106	hr
1	hrTable	106	newrole

Success Message:

Privilege Successfully Revoked

System status bar at the bottom shows the date (11/22/2020), time (4:00 PM), and language (ENG).

S Revoke Privilege +

localhost/dbms/SecurityDB/Forms/Code/html/revoke.php

Apps (90) YouTube Project AcgnX Torre... Just a moment... Mail - abhishek.sh... StarRez Portal - Stu... TimePro® Employee... Introduction to Py... Solve Programming... ↗

DACM Discretionary Access Control Management

Revoke Privilege for a User on A Table

User Settings Existing Privileges on Table

Grantor ID	Table Name	Privilege ID	Role Name
1	hrTable	106	hr

As Owner With ID: 1
of Table: hrTable
Revoke Privilege with ID: 106
To Role: newrole

4:00 PM 11/22/2020

S Revoke Privilege +

localhost/dbms/SecurityDB/Forms/Code/html/revoke.php

Apps (90) YouTube Project AcgnX Torre... Just a moment... Mail - abhishek.sh... StarRez Portal - Stu... TimePro® Employee... Introduction to Py... Solve Programming... ↗

DACM Discretionary Access Control Management

Revoke Unsuccessful as Privilege / Role may be invalid

Revoke Privilege for a User on A Table

User Settings Existing Privileges on Table

Grantor ID	Table Name	Privilege ID	Role Name
1	hrTable	106	hr

As Owner With ID:
of Table:
Revoke Privilege with ID:
To Role:

4:00 PM 11/22/2020

RETRIEVING ROLE PRIVILEGES

The screenshot shows two instances of a web browser displaying the 'Retrieving Role Privileges' page of a Discretionary Access Control Management (DADM) application. The application is running on a local host at `localhost/dbms/SecurityDB/Forms/Code/php/retrieve_role-priv-mysql.php`.

The interface includes a sidebar with various administrative tasks such as Create New User, Create New Role, Create New Table, Insert New Privilege, etc. The main content area is titled 'Retrieving Role Privileges'.

In the 'User Settings' section, there is a 'Role Name' input field containing 'newrole' and a 'Execute Query' button.

The 'Roles and Privileges' section contains a table:

Role	Privilege
newrole	select

A green message box at the bottom left of the table area states: 'Privileges for newrole found!'

The system status bar at the bottom right shows the time as 4:00 PM and the date as 11/22/2020.

The screenshot shows a web application titled "Discretionary Access Control Management". The left sidebar contains a navigation menu with various options such as "Create New User", "Create New Role", "Create New Table", "Insert New Privilege", "Relate User Account to Role", "Relate Acc Priv to Role", "Add Relation Privilege", "Relate Relation Priv to Role and Table", "Revoke Privilege of User From Table", "Retrieve Role Privileges", "Retrieve User Privileges", "Check Privilege for User", and "Access Table as User". The main content area has a title "Retrieving Role Privileges" and two tabs: "User Settings" and "Roles and Privileges". The "User Settings" tab is active, showing a search input field with "hr" and a blue "Execute Query" button. The "Roles and Privileges" tab displays a table with columns "Role" and "Privilege", containing one row: "newrole" under "Role" and "select" under "Privilege". The browser's address bar shows the URL "localhost/dbms/SecurityDB/Forms/Code/php/retrieve_role-priv-mysql.php". The system tray at the bottom right shows the date and time as "11/22/2020 4:00 PM".

localhost/dbms/SecurityDB/Forms/Code/php/retrieve_role-priv-mysql.php

Create New User
Create New Role
Create New Table
Insert New Privilege
Relate User Account to Role
Relate Acc Priv to Role
Add Relation Privilege
Relate Relation Priv to Role and Table
Revoke Privilege of User From Table
Retrieve Role Privileges
Retrieve User Privileges
Check Privilege for User
Access Table as User

Discretionary Access Control Management

Retrieving Role Privileges

User Settings Roles and Privileges

Role	Privilege
hr	create
hr	drop
hr	alter
hr	insert
hr	delete
hr	update
hr	select

Name:

Privileges for hr found!

RETRIEVING PRIVILEGES FOR A USER

The screenshot shows a web application titled "Discretionary Access Control Management". The main page is titled "Retrieving Privileges for a User". On the left, there is a sidebar with various menu items under the heading "DADM". The "User Settings" section contains a "User ID" input field with the value "1" and a "Execute Query" button. The "Roles and Privileges" section contains a table with three columns: "User ID", "User Name", and "Privilege". The table shows the following data:

User ID	User Name	Privilege
1	J.Jane	create
1	J.Jane	drop
1	J.Jane	alter
1	J.Jane	insert
1	J.Jane	delete
1	J.Jane	update
1	J.Jane	select

This screenshot is identical to the one above, showing the "Discretionary Access Control Management" application. The "User Settings" section has "User ID" set to "ID" and the "Execute Query" button is visible. The "Roles and Privileges" section shows the same table of privileges for User ID 1. A green message box at the bottom left of the table area says "Privileges for user with ID 1 found!".

Screenshot of a web browser showing the "Retrieving Privileges for a User" page of the Discretionary Access Control Management (DADM) application.

The URL is `localhost/dbms/SecurityDB/Forms/Code/php/retrieve-user-priv.php`.

The sidebar on the left contains the following menu items:

- Create New User
- Create New Role
- Create New Table
- Insert New Privilege
- Relate User Account to Role
- Relate Acc Priv to Role
- Add Relation Privilege
- Relate Relation Priv to Role and Table
- Revoke Privilege of User From Table
- Retrieve Role Privileges
- Retrieve User Privileges
- Check Privilege for User
- Access Table as User

The main content area shows the "User Settings" section with a "User ID" input field containing "21" and a "Execute Query" button. To the right is a "Roles and Privileges" table:

User ID	User Name	Privilege
1	J.Jane	create
1	J.Jane	drop
1	J.Jane	alter
1	J.Jane	insert
1	J.Jane	delete
1	J.Jane	update
1	J.Jane	select

The status bar at the bottom shows the date and time as 4:01 PM 11/22/2020.

Screenshot of a web browser showing the same "Retrieving Privileges for a User" page, but with a different result.

The URL is `localhost/dbms/SecurityDB/Forms/Code/php/retrieve-user-priv.php`.

The sidebar on the left is identical to the first screenshot.

The main content area shows the "User Settings" section with a "User ID" input field containing "ID" and a "Execute Query" button. Below the table, a message box displays: "Privileges for user with ID 21 not found".

The status bar at the bottom shows the date and time as 4:01 PM 11/22/2020.

CHECKING PRIVILEGES FOR A USER

Discretionary Access Control Management

Checking Privilege for a User

User Settings

User ID: 1
Privilege Type: create

Roles and Privileges

User ID	Privilege Type
1	create

Execute Query

Discretionary Access Control Management

Checking Privilege for a User

User Settings

User ID:
Privilege Type:

Roles and Privileges

User ID	Privilege Type
JJane	create

Execute Query

User with ID 1 has create privilege

Retrieving Role Privileges

localhost/doms/SecurityDB/Forms/Code/php/check-priv-for-user.php

DACM

Discretionary Access Control Management

Checking Privilege for a User

User Settings Roles and Privileges

User ID	Privilege Type
JJane	create

User ID: 1
Privilege Type: reference

Create New User
Create New Role
Create New Table
Insert New Privilege
Relate User Account to Role
Relate Acc Priv to Role
Add Relation Privilege
Relate Relation Priv to Role and Table
Revoke Privilege of User From Table
Retrieve Role Privileges
Retrieve User Privileges
Check Privilege for User
Access Table as User

4:01 PM 11/22/2020

Retrieving Role Privileges

localhost/dbms/SecurityDB/Forms/Code/php/check-priv-for-user.php

Just a moment... Mail - abhishek.sh... StarRez Portal - St... TimePro® Employee... Introduction to Pyt... Solve Programming...

DACM

Discretionary Access Control Management

Checking Privilege for a User

User Settings Roles and Privileges

User ID	Privilege Type
<input type="text"/>	

User ID Privilege Type

Privilege Type

User with ID1 does not have reference privilege

Create New User Create New Role Create New Table Insert New Privilege Relate User Account to Role Relate Acc Priv to Role Add Relation Privilege Relate Relation Priv to Role and Table Revoke Privilege of User From Table Retrieve Role Privileges Retrieve User Privileges Check Privilege for User Access Table as User

ACCESS TABLE AS A USER

The screenshot shows two identical instances of a web browser window titled "Access Table as a User". The URL is `localhost/dbms/SecurityDB/Forms/Code/html/user-proof.php`. The page header is "Discretionary Access Control Management". On the left, there is a sidebar menu with the following items:

- Create New User
- Create New Role
- Create New Table
- Insert New Privilege
- Relate User Account to Role
- Relate Acc Priv to Role
- Add Relation Privilege
- Relate Relation Priv to Role and Table
- Revoke Privilege of User From Table
- Retrieve Role Privileges
- Retrieve User Privileges
- Check Privilege for User
- Access Table as User

The main content area is titled "Access Table as a User" and contains a "User Settings" section. It includes fields for "As User" (set to "JJane"), "Access Privilege" (set to "select"), and "on Table" (set to "hrTable"). Below these fields is a blue "Execute Query" button. In the second instance of the browser, a green message bar at the top states: "Access has already been granted! User can perform the entered operation on this table".

S Access Table as a User +

localhost/dbms/SecurityDB/Forms/Code/html/user-proof.php

Apps (90) YouTube Project AcgnX Torre... Just a moment... Mail - abhishek.sh... StarRez Portal - Stu... TimePro® Employee... Introduction to Py... Solve Programming... ↗

DACM Discretionary Access Control Management

Access Table as a User

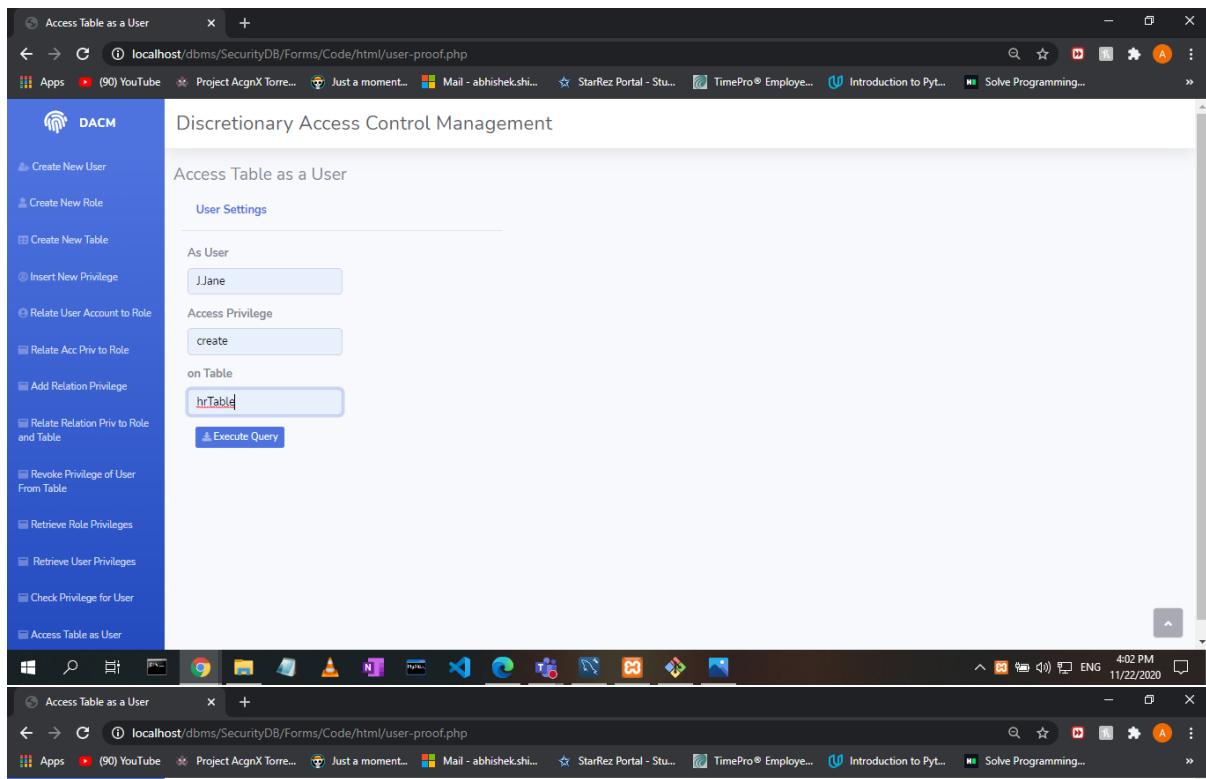
User Settings

As User: JJane

Access Privilege: create

on Table: hrTable

Execute Query



S Access Table as a User +

localhost/dbms/SecurityDB/Forms/Code/html/user-proof.php

Apps (90) YouTube Project AcgnX Torre... Just a moment... Mail - abhishek.sh... StarRez Portal - Stu... TimePro® Employee... Introduction to Py... Solve Programming... ↗

DACM Discretionary Access Control Management

Access Table as a User

Access denied. User does not have privilege on this table

User Settings

As User: (empty)

Access Privilege: (empty)

on Table: (empty)

Execute Query

