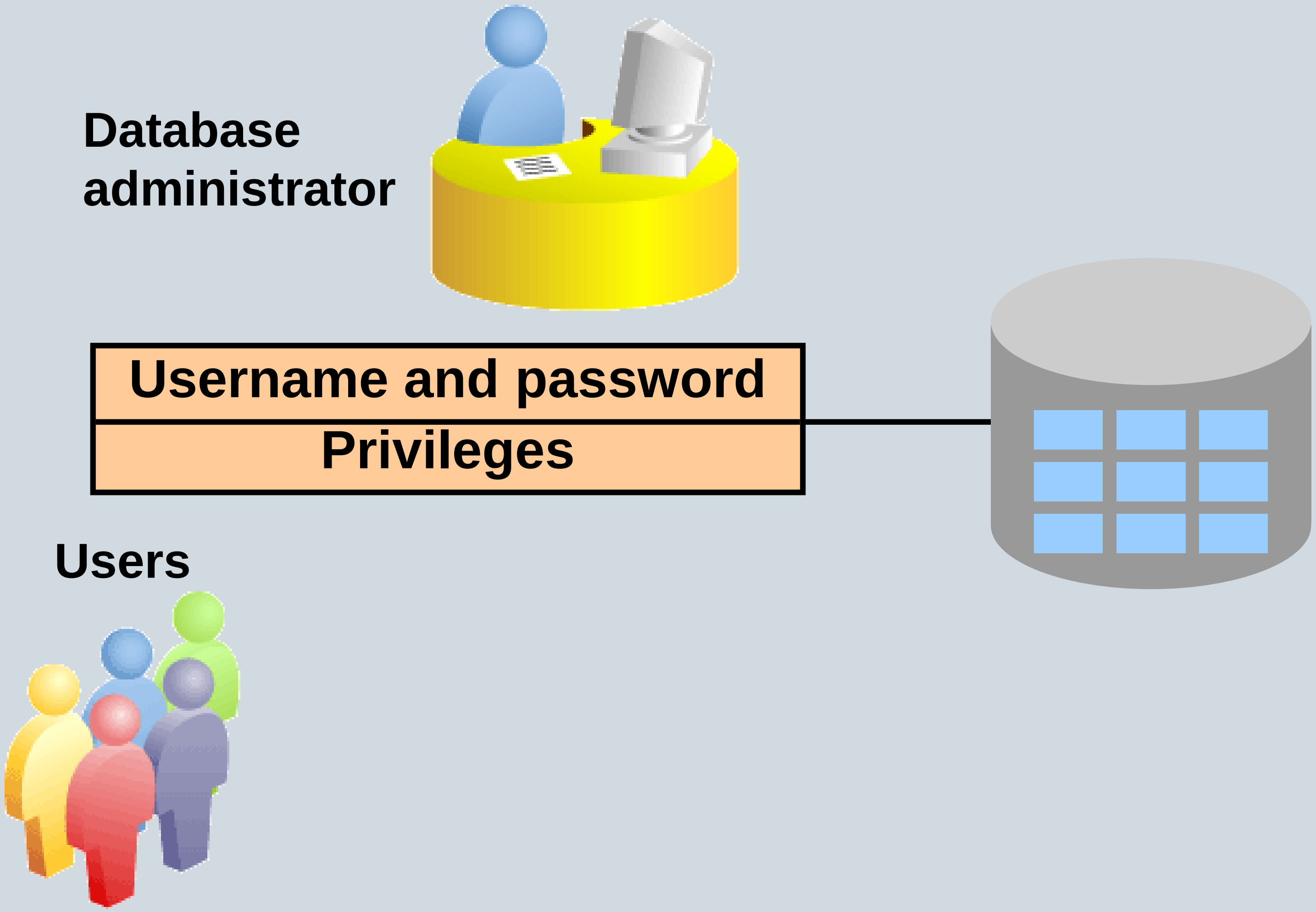
Controlling User Access

Controlling User Access



Database administrator

Username and password

Privileges

Users

In a multiple-user environment, you want to maintain security of database access and use.

With Oracle Server database security, you can do the following:

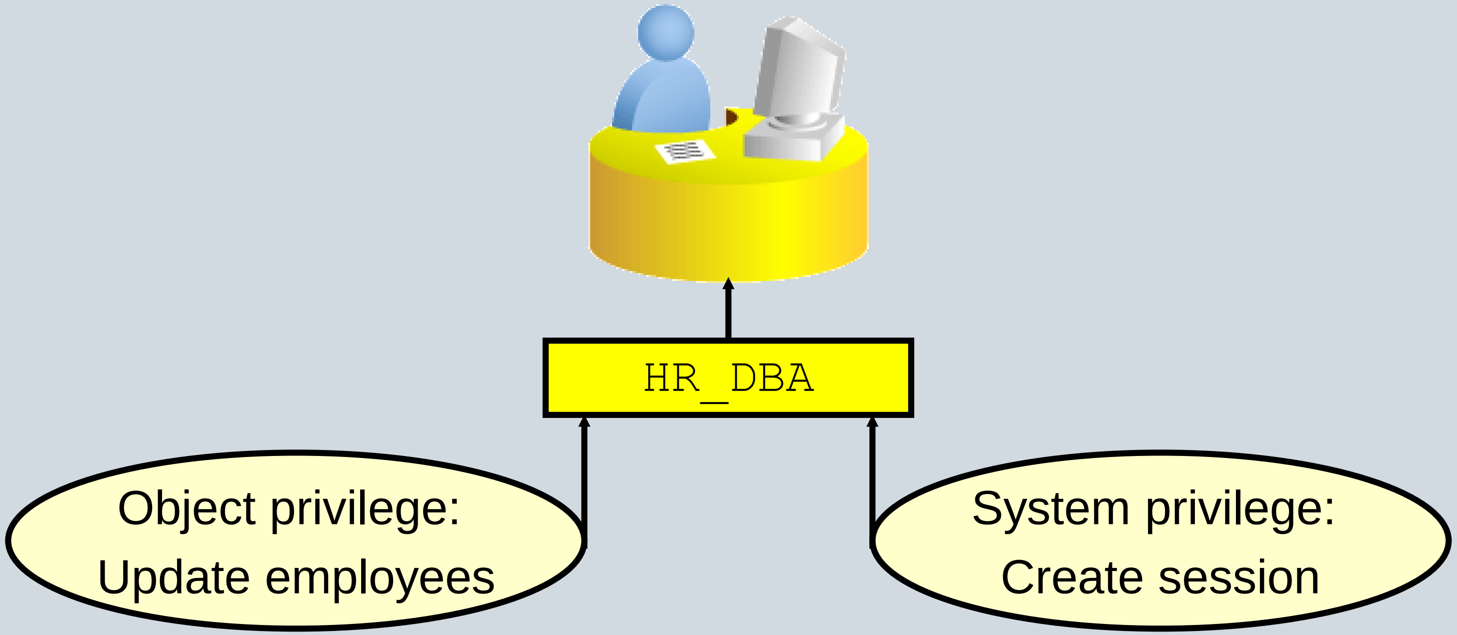
- Control database access.
- Give access to specific objects in the database.
- Confirm given and received privileges with the Oracle data dictionary.

Database security can be classified into two categories: system security and data security.
System security covers access and use of the database at the system level, such as the username and password, the disk space allocated to users, and the system operations that users can perform.
Database security covers access and use of the database objects and the actions that those users can perform on the objects.

Privileges

There are two types of user privileges:

- System: Enables users to perform particular actions in the database (170 distinct system privileges)
- Object: Enables users to access and manipulate a specific object (Without specific permission, users can access only their own objects. Object privileges can be granted by the owner of an object, by the administrator, or by someone who has been explicitly given permission to grant privileges on the object.)



HR_DBA

Object privilege:
Update employees

System privilege:
Create session

Privileges

- Database security:
  - System security
  - Data security
- System privileges: Performing a particular action within the database
- Object privileges: Manipulating the content of the database objects
- Schemas: Collection of objects such as tables, views, and sequences

A privilege is the right to execute particular SQL statements. The database administrator (DBA) is a high-level user with the ability to create users and grant users access to the database and its objects. Users require system privileges to gain access to the database and object privileges to manipulate the content of the objects in the database. Users can also be given the privilege to grant additional privileges to other users or to roles, which are named groups of related privileges.

Schemas

A schema is a collection of objects such as tables, views, and sequences. The schema is owned by a database user and has the same name as that user. A system privilege is the right to perform a particular action, or to perform an action on any schema objects of a particular type. An object privilege provides the user the ability to perform a particular action on a specific schema object.

System Privileges

- More than 200 privileges are available.
- The database administrator has high-level system privileges for tasks such as:
  - Creating new users
  - Removing users
  - Removing tables
  - Backing up tables

More than 200 distinct system privileges are available for users and roles. The table SYSTEM PRIVILEGE MAP contains all the system privileges available, based on the version release. This table is also used to map privilege type numbers to type names.

Creating Users

The DBA creates users with the `CREATE USER` statement.

```
CREATE USER user
IDENTIFIED BY   password;
```

```
CREATE USER  demo
IDENTIFIED BY demo;
```

User System Privileges

- After a user is created, the DBA can grant specific system privileges to that user.

```
GRANT privilege [, privilege...]
TO user [, user| role, PUBLIC...] [WITH ADMIN OPTION];
```

- An application developer, for example, may have the following system privileges:
  - CREATE SESSION
  - CREATE TABLE
  - CREATE SEQUENCE
  - CREATE VIEW
  - CREATE PROCEDURE

User System Privileges

Typical User Privileges

After the DBA creates a user, the DBA can assign privileges to that user.

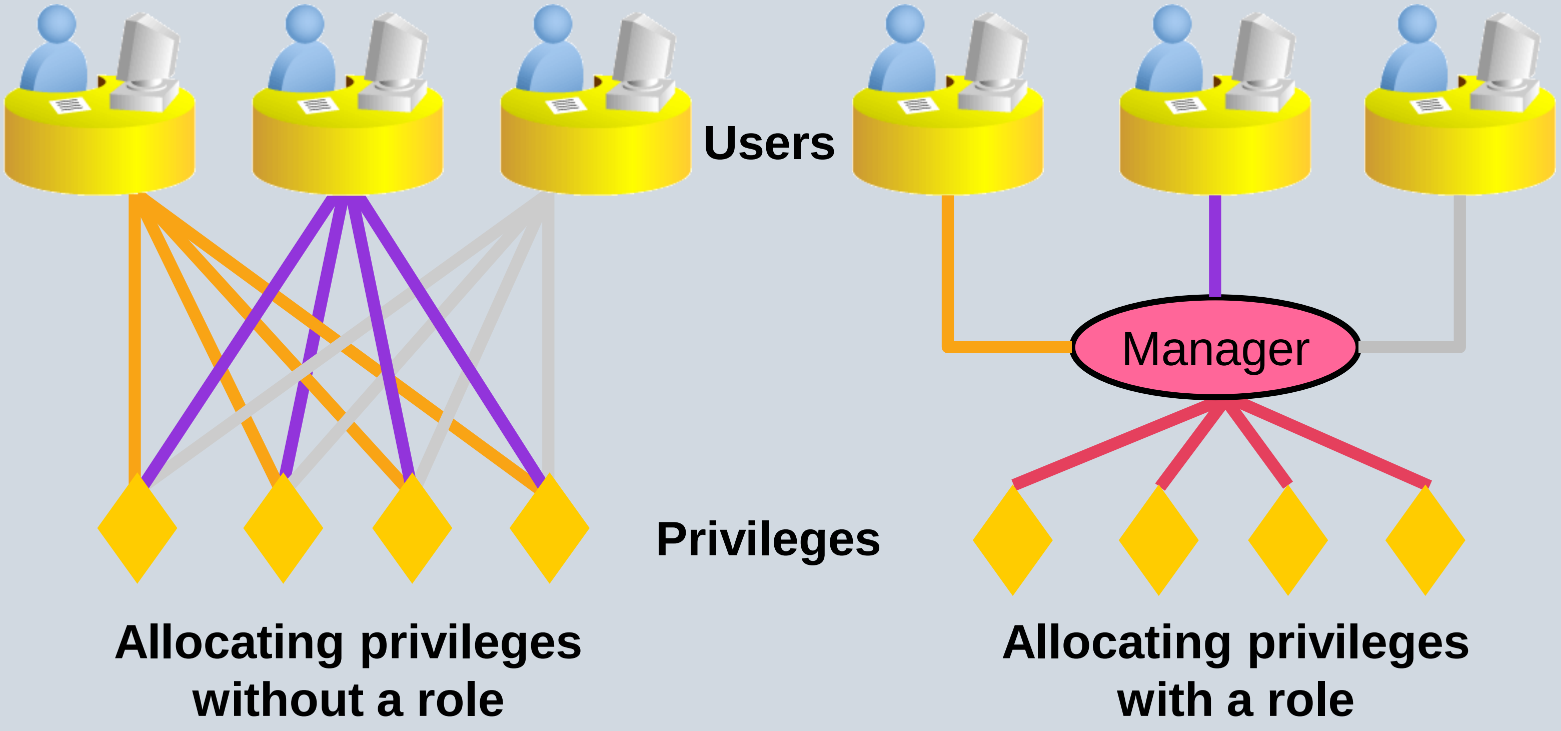
System Privilege	**Operations Authorized**
CREATE SESSION	Connect to the database.
CREATE TABLE	Create tables in the user’s schema.
CREATE SEQUENCE	Create a sequence in the user’s schema.
CREATE VIEW	Create a view in the user’s schema.
CREATE PROCEDURE	Create a stored procedure, function, or package in the user’s schema.

Granting System Privileges

The DBA can grant specific system privileges to a user.

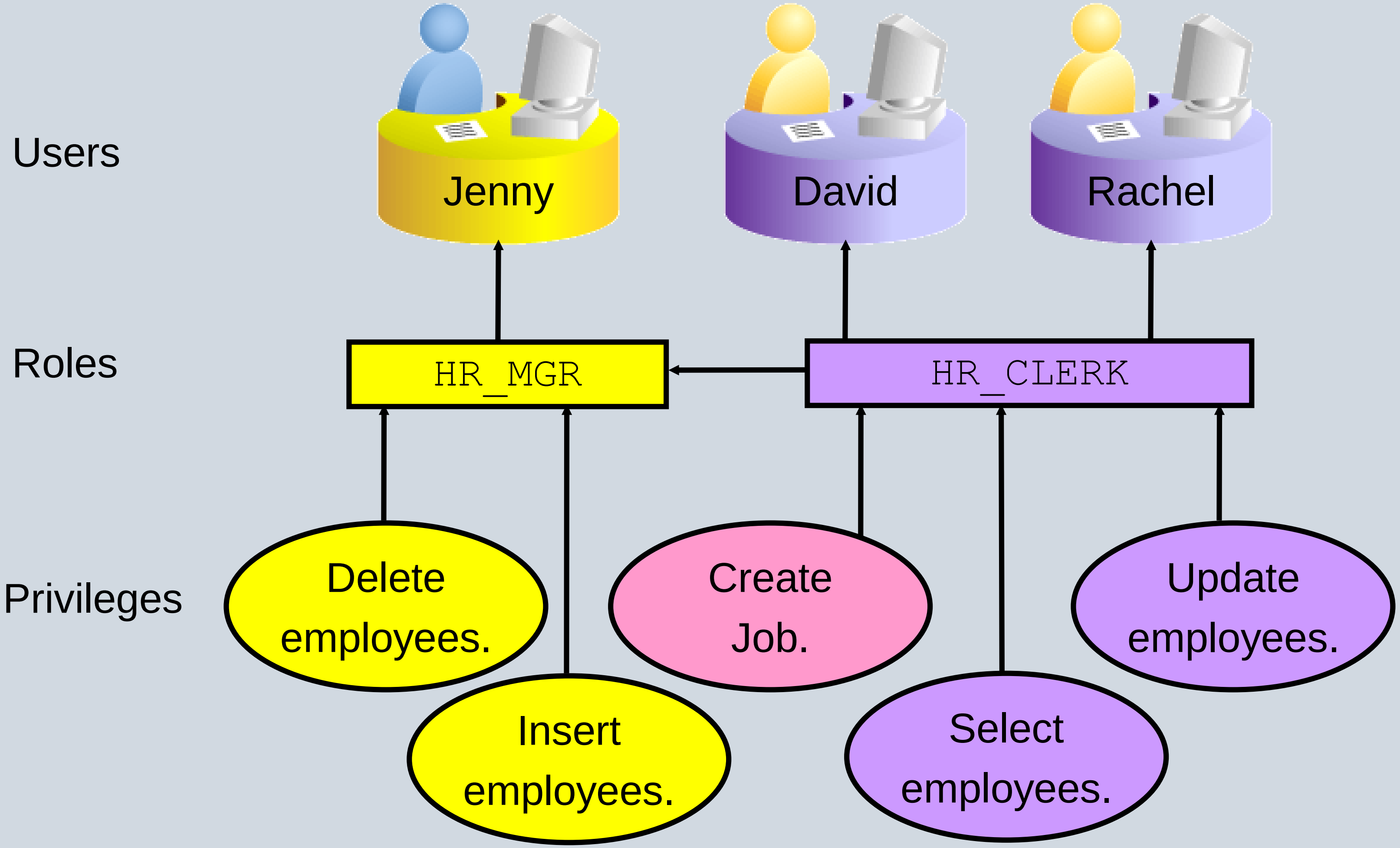
```
GRANT  create session, create table,
       create sequence, create view
TO     demo;
```

What Is a Role?


Users
Manager
Privileges
Allocating privileges without a role
Allocating privileges with a role

A role is a named group of related privileges that can be granted to the user. This method makes it easier to revoke and maintain privileges. A user can have access to several roles, and several users can be assigned the same role. Roles are typically created for a database application.

Assigning Privileges to Roles and Assigning Roles to Users



Creating and Granting Privileges to a Role

- Create a role:

```
CREATE ROLE manager;
```

- Grant privileges to a role:

```
GRANT create table, create view
TO manager;
```

- Grant a role to users:

```
GRANT manager TO alice;
```

Changing Your Password

- The DBA creates your user account and initializes your password.
- You can change your password by using the `ALTER USER` statement.

```
ALTER USER demo
IDENTIFIED BY employ;
```

Object Privileges

Object privilege	Table	View	Sequence
ALTER	✓		✓
DELETE	✓	✓	
INDEX	✓		
INSERT	✓	✓	
REFERENCES	✓		
SELECT	✓	✓	✓
UPDATE	✓	✓	

Object Privileges

- Object privileges vary from object to object.
- An owner has all the privileges on the object.
- An owner can give specific privileges on that owner’s object.

```
GRANT       object_priv [(columns)]
ON          object
TO          {user|role|PUBLIC}
[WITH GRANT OPTION];
```

Granting Object Privileges

- Grant query privileges on the `EMPLOYEES` table:

```
GRANT  select
ON     employees
TO     demo;
```

- Grant privileges to update specific columns to users and roles:

```
GRANT  update (department_name, location_id)
ON     departments
TO     demo, manager;
```

Guidelines

- To grant privileges on an object, the object must be in your own schema, or you must have been granted the object privileges WITH GRANT OPTION.
- An object owner can grant any object privilege on the object to any other user or role of the database.
- The owner of an object automatically acquires all object privileges on that object.

Passing On Your Privileges

- Give a user authority to pass along privileges:

```
GRANT  select, insert
ON     departments
TO     demo
WITH   GRANT OPTION;
```

- Allow all users on the system to query data from Alice’s `DEPARTMENTS` table:

```
GRANT  select
ON    alice.departments
TO    PUBLIC;
```

Confirming Granted Privileges

Data Dictionary View	Description
ROLE_SYS_PRIVS	System privileges granted to roles
ROLE_TAB_PRIVS	Table privileges granted to roles
USER_ROLE_PRIVS	Roles accessible by the user
USER_SYS_PRIVS	System privileges granted to the user
USER_TAB_PRIVS_MADE	Object privileges granted on the user’s objects
USER_TAB_PRIVS_RECD	Object privileges granted to the user
USER_COL_PRIVS_MADE	Object privileges granted on the columns of the user’s objects
USER_COL_PRIVS_RECD	Object privileges granted to the user on specific columns

Revoking Object Privileges

- You use the REVOKE statement to revoke privileges granted to other users.
- Privileges granted to others through the WITH GRANT OPTION clause are also revoked.

```
REVOKE {privilege [, privilege...]|ALL}
ON    object
FROM   {user[, user...]|role|PUBLIC}
[CASCADE CONSTRAINTS];
```

You can remove privileges granted to other users by using the REVOKE statement. When you use the REVOKE statement, the privileges that you specify are revoked from the users you name and from any other users to whom those privileges were granted by the revoked user.

In the syntax:

CASCADE Is required to remove any referential integrity constraints made to the CONSTRAINTS object by means of the REFERENCES privilege

Revoking Object Privileges

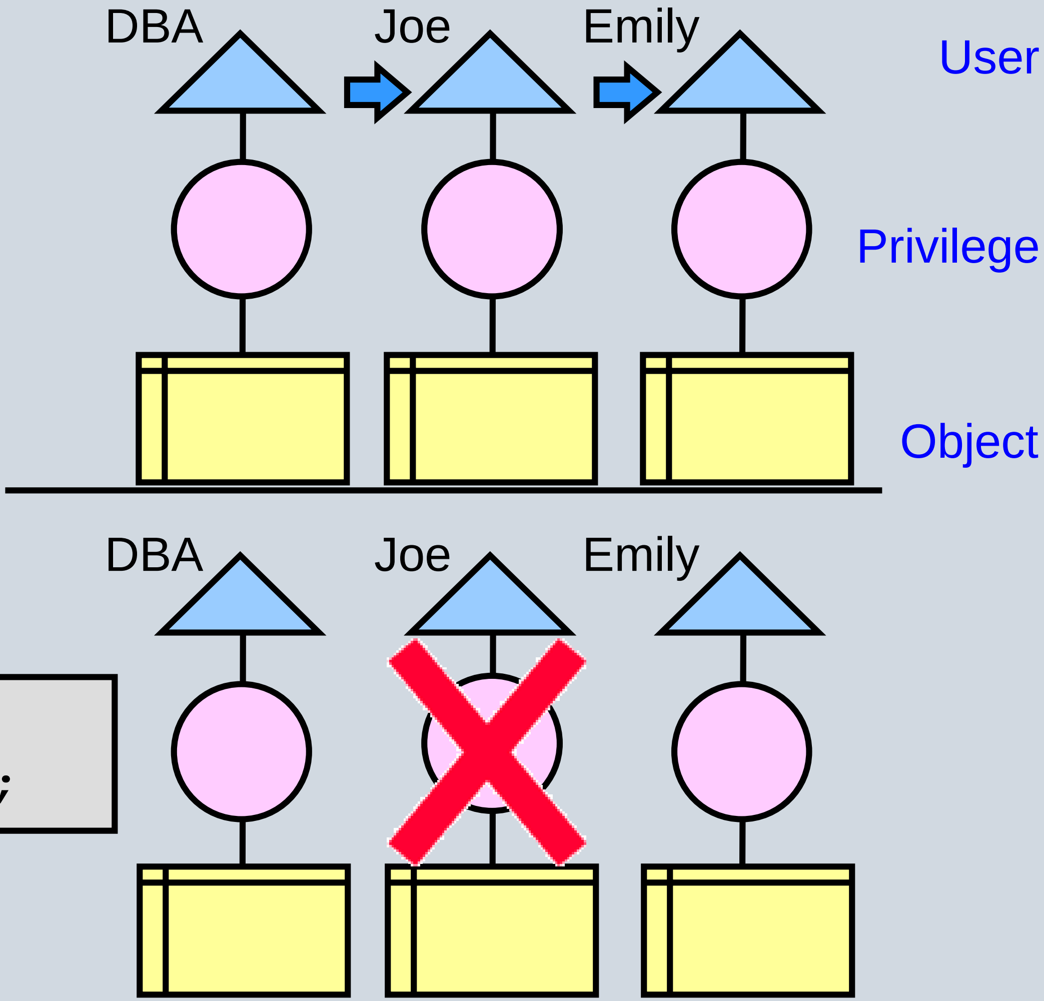
Revoke the `SELECT` and `INSERT` privileges given to the `demo` user on the `DEPARTMENTS` table.

```
REVOKE  select, insert
ON      departments
FROM    demo;
```

REVOKE <*system_privilege*> FROM <*grantee clause*>

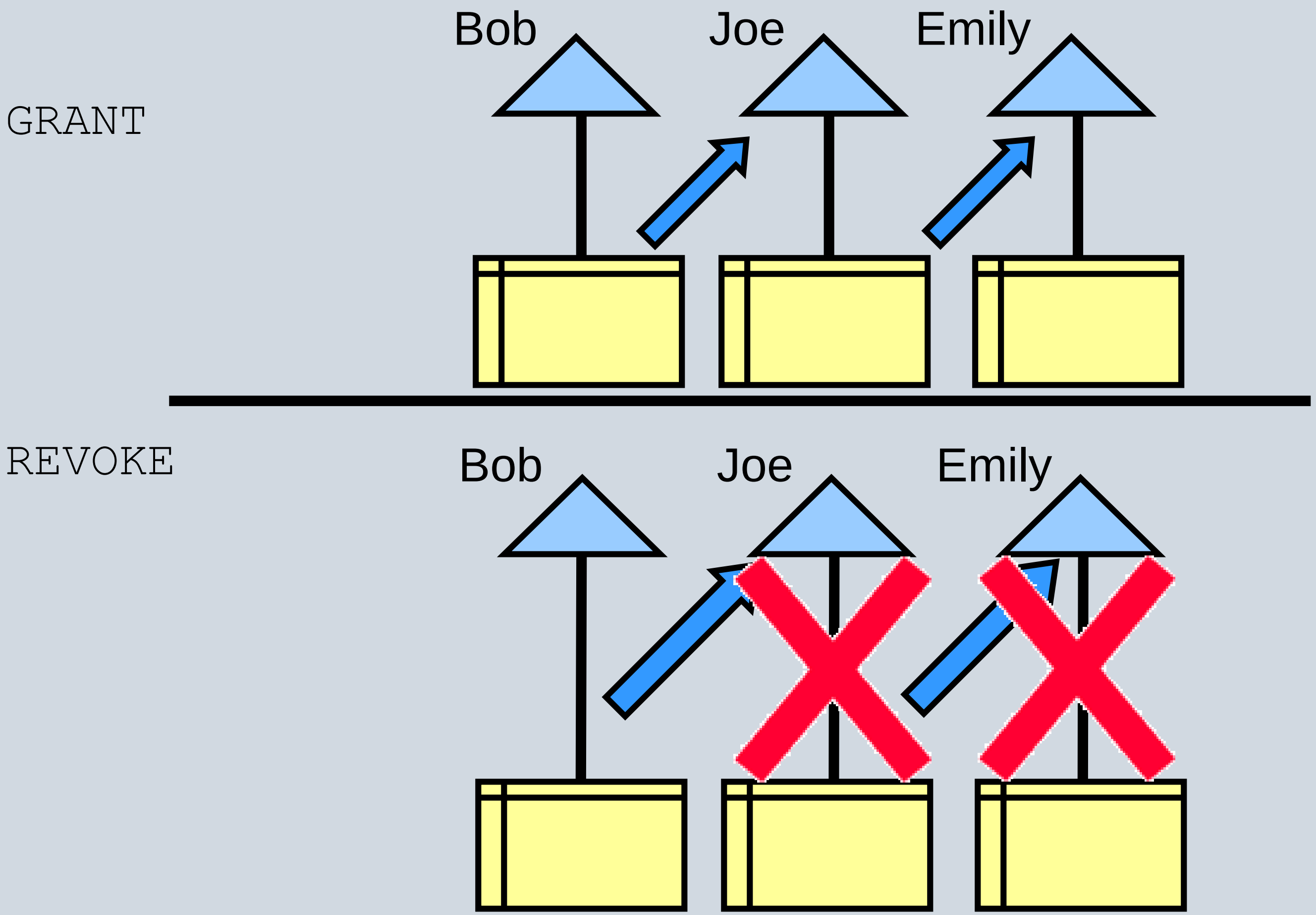
GRANT

REVOKE



```
REVOKE CREATE
TABLE FROM joe;
```

Revoking Object Privileges with GRANT OPTION


GRANT
Bob
Joe
Emily
REVOKE
Bob
Joe
Emily