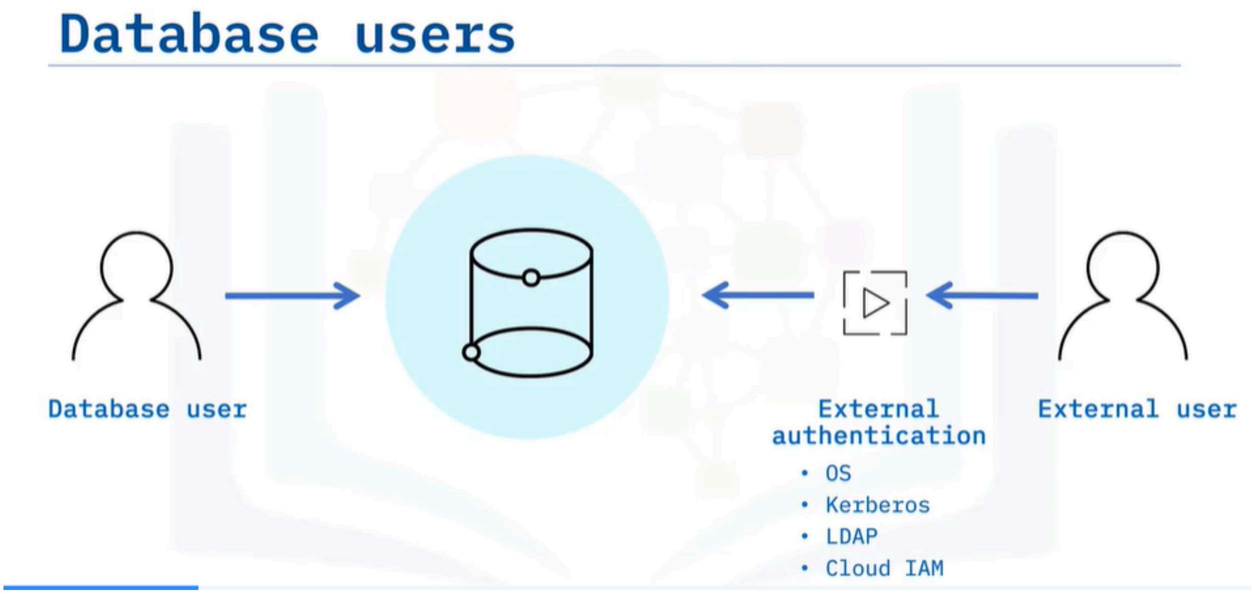
Users, Groups, and Roles



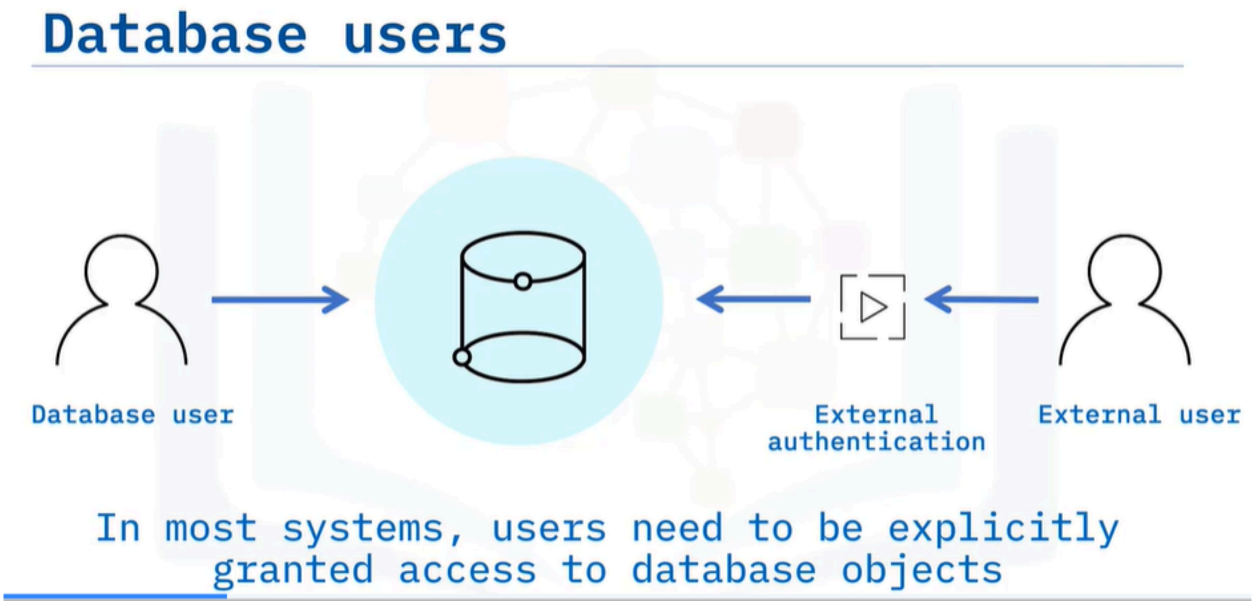
Database users

Database user

External authentication

- OS
- Kerberos
- LDAP
- Cloud IAM

External user



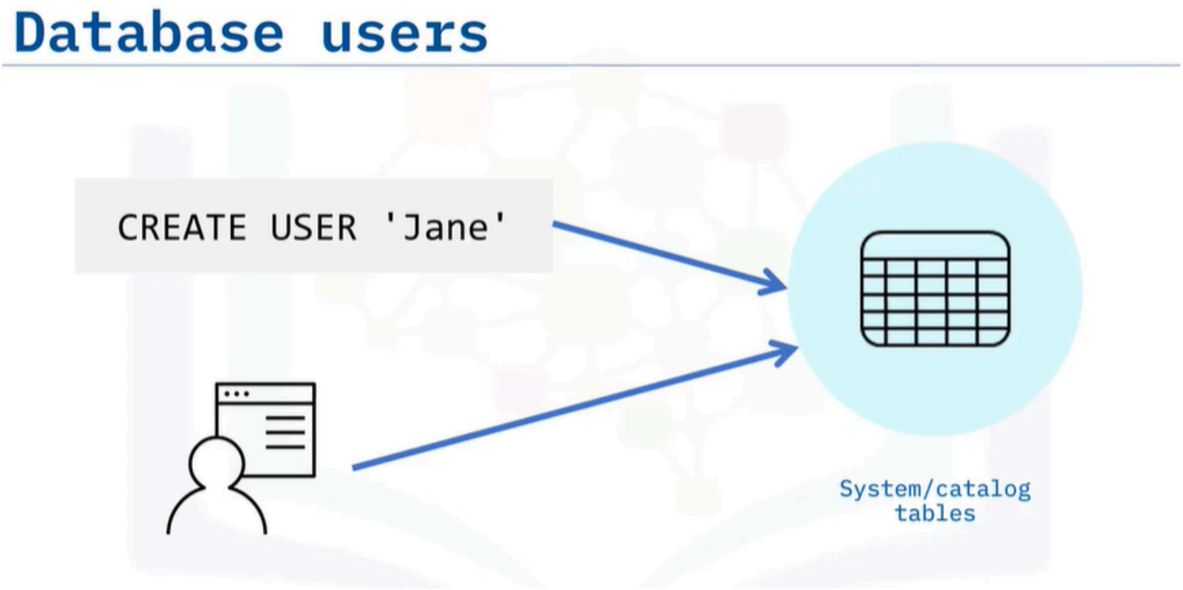
Database users

Database user

External authentication

External user

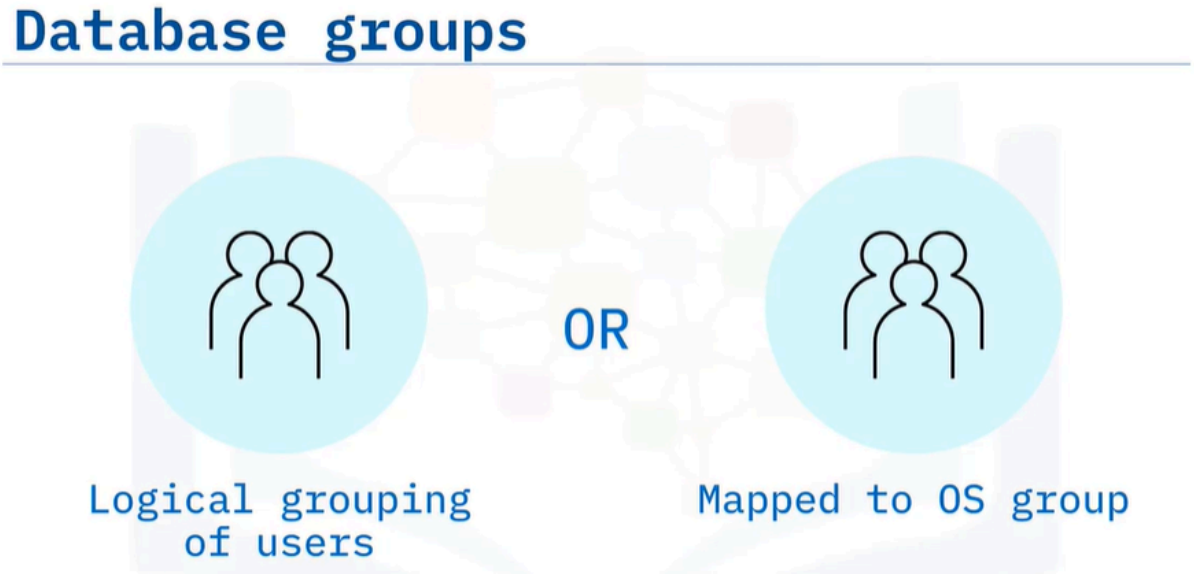
In most systems, users need to be explicitly granted access to database objects



Database users

```
CREATE USER 'Jane'
```

- A database user is a user account that is allowed to access specified database objects. Depending on the DBMS and your security policies, a user might be explicitly created and authenticated within the database system, or may be created externally and authenticated using external authentication such as the operating system or external identity management services like Kerberos, LDAP, and Cloud IAM.
- When you first create a user, they will generally have few, if any, permissions to actually interact with the database objects unless they are the creator of the database. User names are stored in system tables or catalog tables which you should not try to edit directly. All RDBMSs will provide SQL commands and/or user interface tools that you can use to manage your users.



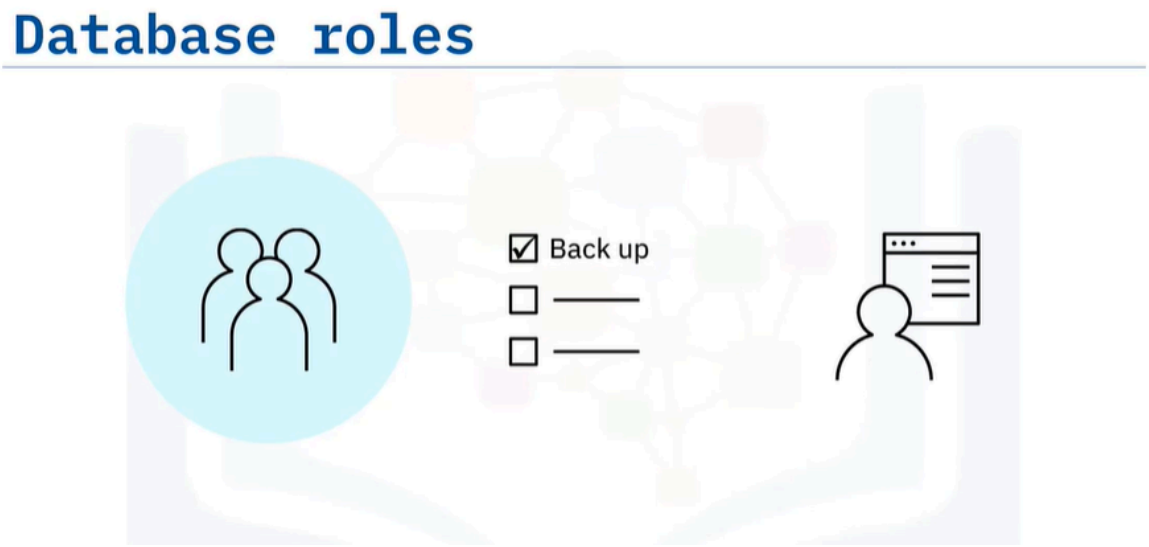
Database groups

Logical grouping of users

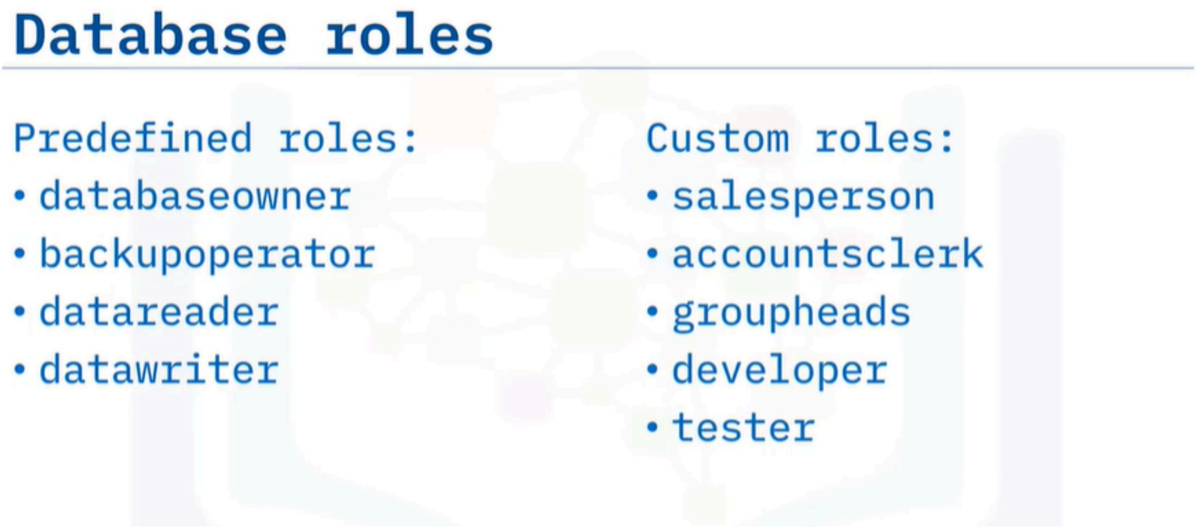
OR

Mapped to OS group

- Some RDBMSs support the concept of user groups. In some cases, such as Postgres, you define groups in the database and they are logical groupings of users to simplify user management. In other systems, such as SQL Server and Db2, you can map a database group to an administrative group in the underlying operating system. This is particularly useful when you are using that operating system to authenticate your users. Similarly, if you are running your database on Amazon Relational Database Service (RDS) for example, you can use virtual private cloud (VPC) security groups and database (DB) security groups to manage your user access.



Database roles

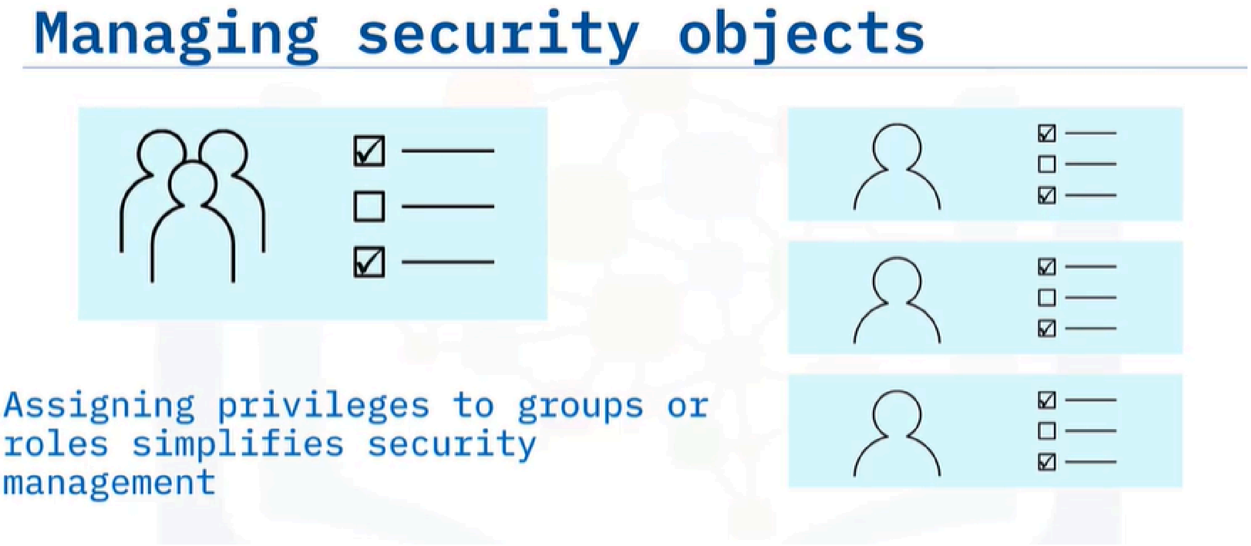


Database roles

Predefined roles:
- databaseowner
- backupoperator
- datareader
- datawriter

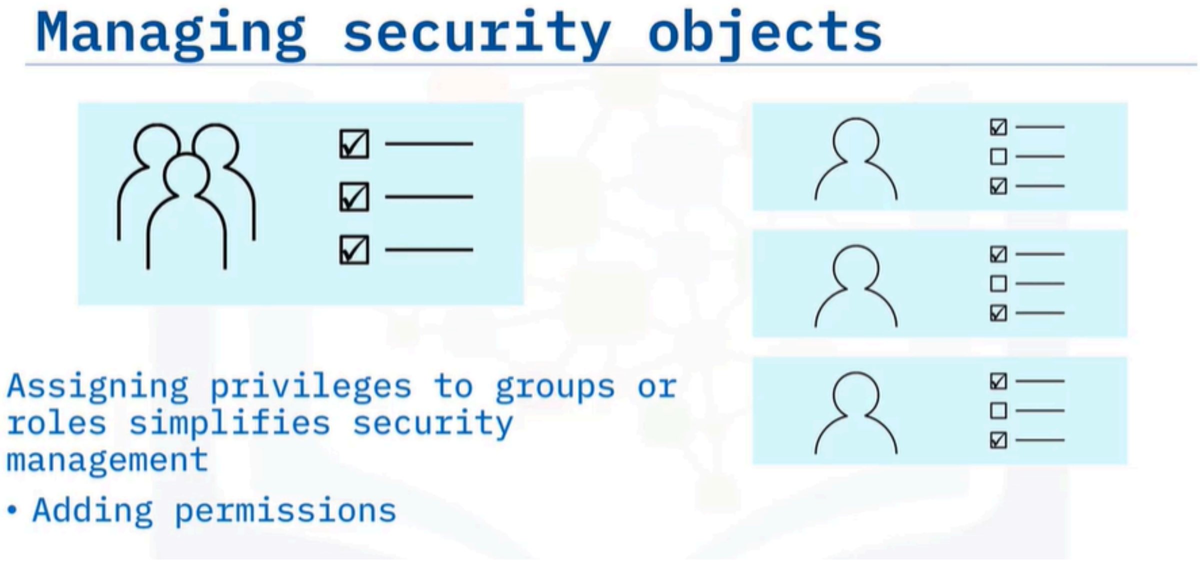
Custom roles:
- salesperson
- accountsclerk
- groupheads
- developer
- tester

- Database roles are similar to database groups in that they confer privileges and access rights to all users of that role. A database role defines a set of permissions needed to undertake a specific role in the database.
- For example, a backup operator role would have permissions to access a database and to perform backup functions. Some RDBMSs have a set of predefined roles that you can use for your users, such as a database owner or backup operator role. And most enable you to create your own roles, so you might create a salesperson role that has the permissions a salesperson will need on the relevant tables in the database.



Managing security objects

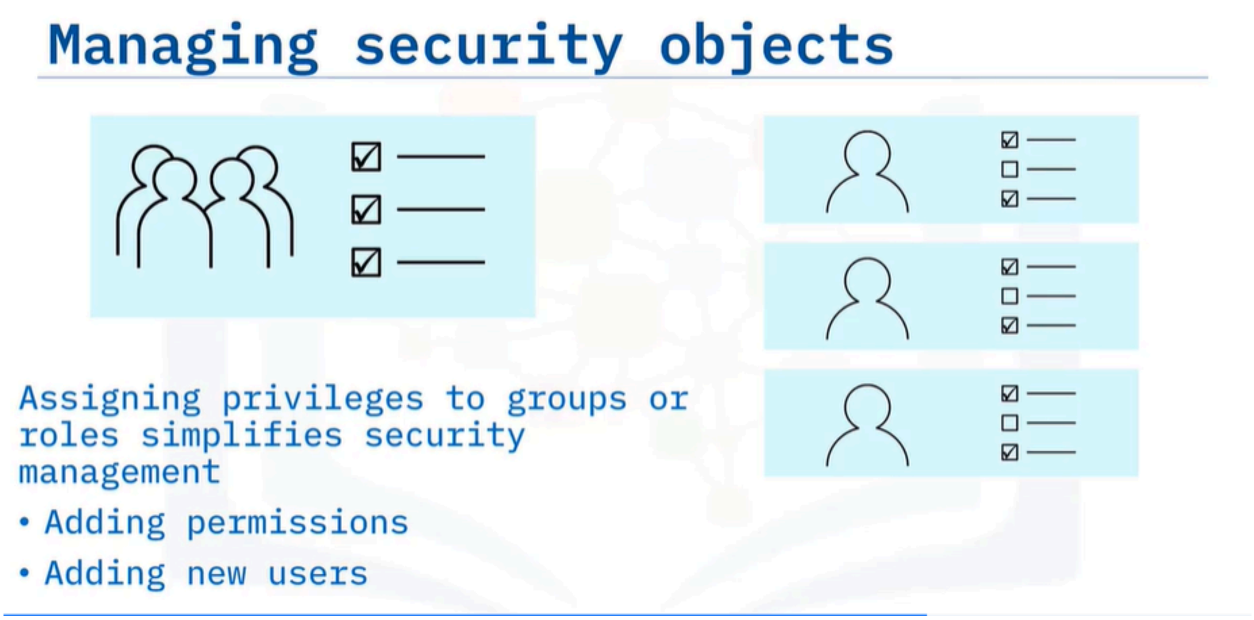
Assigning privileges to groups or roles simplifies security management



Managing security objects

Assigning privileges to groups or roles simplifies security management

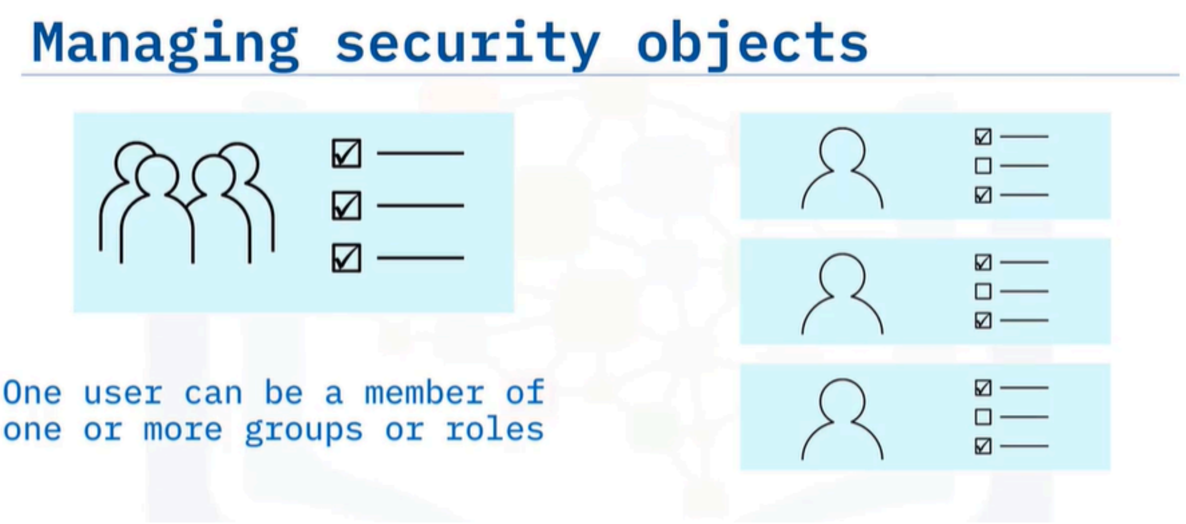
- Adding permissions



Managing security objects

Assigning privileges to groups or roles simplifies security management

- Adding permissions
- Adding new users

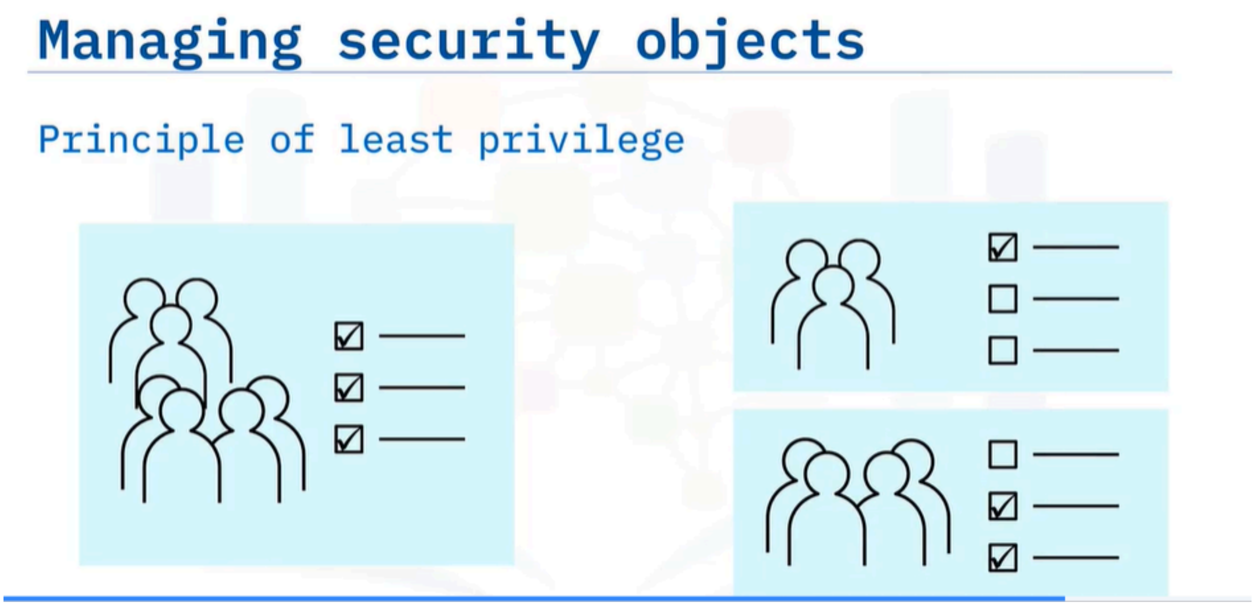


Managing security objects

One user can be a member of one or more groups or roles

- Assigning privileges to groups or roles, rather than individual users, greatly simplifies your security management tasks. If you know that a set of users all need access to the same functionality to fulfill their job role, you can put those users in one group and assign the relevant permissions to the group.
- If that job role changes in the future, it is quicker, easier, and less prone to mistakes to just add the new permissions to the group rather than individual users.

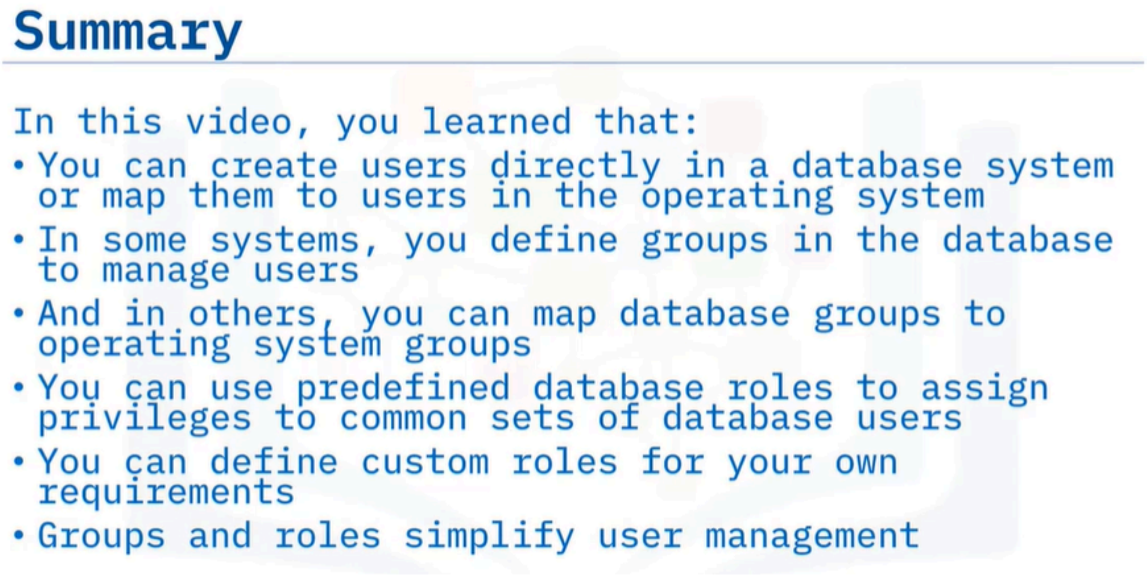
- Similarly, if a new member of staff joins the team, you can simply add them to the role or group, rather than having to assign all the separate permissions to them. One user can be a member of one or more roles. So the head of the sales group could be a member of the group heads role and the salesperson role.



Managing security objects

Principle of least privilege

- When implementing role or group membership, remember to use the principle of least privilege: only adding users to groups they need to be a member of, and ensuring that your roles don’t include any permissions that the majority of the users in them will not need. In the latter scenario, you should reassess your role permissions and create multiple, more granular, roles or groups.



Summary

In this video, you learned that:

- You can create users directly in a database system or map them to users in the operating system
- In some systems, you define groups in the database to manage users
- And in others, you can map database groups to operating system groups
- You can use predefined database roles to assign privileges to common sets of database users
- You can define custom roles for your own requirements
- Groups and roles simplify user management

Summary

In simple terms, a **database user** is like a person who has a key to enter a building (the database). This key allows them to access certain rooms (database objects) inside. When you create a user, they usually start with very limited access, just like a new tenant who might not have access to all areas of the building until they get the right permissions.

Now, think of **groups** as teams of people who share similar responsibilities. For example, if you have a group of salespeople, you can give them all the same access to the sales data without having to give each person individual permissions. This makes it easier to manage who can see what. Similarly, **roles** are like job titles that come with specific responsibilities. For instance, a "backup operator" role would have the necessary permissions to perform backups of the database. By assigning roles to users, you can streamline access and ensure everyone has what they need to do their job effectively.