

# **Database Administration Basics**

**Creating Databases, Managing Users & Roles**

By: Daniyal Haider (13298)

Anas Khalid (13465)

Ahmad Kamal (13)



# Overview

- Creating Database / Tablespace 03
- Managing Users 04
- Managing Roles 05
- Grant Command 06
- Revoke Command 07
- Conclusion 08



```
name" => null,
"Surname" => null,
"username" => "admin",
"gender" => null,
"email" => "info@necan.com",
"email_verified_at" => null,
"password" => "$2y$10$gX...",
"isActive" => 1,
"user_role" => "Admin",
"avatar" => "assets/images/default-user.png",
"remember_token" => null,
"created_at" => "2022-07-10T10:00:00.000Z",
"updated_at" => "2022-07-10T10:00:00.000Z"
```

# Creating Database / Tablespace



## Database

A database is a collection of information that is organized so that it can be easily accessed, managed, and updated. It could store customer records, product information, transactions, etc.



## Tablespace

A tablespace is a logical storage unit in a database that groups related logical structures together. It contains physical data files where the actual data resides.



## Important Commands:

`CREATE DATABASE database_name;`

`CREATE TABLESPACE tablespace_name  
DATAFILE 'path_to_file'; —`

# Managing Users

01



## What is a User?

A user in the database is an account that can connect to the database and perform operations depending on the permissions given.

02



## Create User Example:

```
CREATE USER username  
IDENTIFIED BY password;
```

Creates a new user with a secure password.

03



## Delete User Example:

```
DROP USER username CASCADE;
```

Deletes a user and all the objects they own.

04



## Why Important?

Managing users properly ensures that only authorized individuals have access to sensitive data, improving security.

# Managing Roles

## What is Role?

A role is a container for privileges. Instead of granting multiple permissions individually to each user, you can create a role, assign permissions to the role, and then assign the role to users.

## Create and Assign Roles:

- **CREATE ROLE role\_name;**
- **GRANT role\_name TO username;**

## Why Important

Using roles makes privilege management simpler, especially when many users need similar access.

# Grant Command



## What is GRANT?

The GRANT command is used to give users specific permissions on database objects (like tables, views, procedures).

## Example:

GRANT SELECT, INSERT ON employees TO john;  
Allows user john to view and insert data into the employees table.



- **WHAT IS REVOKE?**

THE REVOKE COMMAND IS USED TO TAKE BACK PERMISSIONS PREVIOUSLY GRANTED TO USERS OR ROLES.

- **EXAMPLE:**

REVOKE SELECT ON EMPLOYEES FROM JOHN;

USER JOHN WILL NO LONGER BE ABLE TO READ DATA FROM THE EMPLOYEES TABLE.

## Conclusion

- Good database administration ensures security, organization, and efficiency.
- Tablespaces help with physical storage management.
- Roles simplify user privilege management.
- Using GRANT and REVOKE commands properly maintains tight control over who can do what in your database.
- Always review and audit user privileges regularly to prevent potential security risks.





THANK TOU