



Universidad  
**Inca Garcilaso de la Vega**

**Nuevos Tiempos. Nuevas Ideas**

Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones

---

## Usuarios, Roles y Privilegios

**Asignatura: Gestión de Base de Datos**

**Semana 15**

**Docente: Mg. Christian Almóguer Martínez**

**Mail: [almoguer@uigv.edu.pe](mailto:almoguer@uigv.edu.pe)**

# AGENDA

- Usuarios
- Roles
- Privilegios



# Usuario

- En un marco técnico representa la descripción de permisos disponibles durante nuestra conexión.
- Estos permisos se definen sobre un nombre de usuario.
- Por tanto, un usuario no es más que un conjunto de permisos que se aplican a una conexión de base de datos.



## Funciones de usuarios

- Ser el propietario de ciertos objetos.
- Asignacion del *tablespace* para los objetos de un usuario.
- Copias de seguridad.
- Cuotas de almacenamiento.



# Creación de Usuario

```
CREATE USER nombre IDENTIFIED [BY clave| EXTERNALLY]
{DEFAULT TABLESPACE nombre_defecto}
{TEMPORARY TABLESPACE tablespace_temporal}
```

- **IDENTIFIED BY**, permite indicar la autorización
- **DEFAULT TABLESPACE**, será el tablespace por defecto en la creación de objetos del usuario que estamos creando. Se toma como defecto **SYSTEM**.
- **TEMPORARY TABLESPACE**, indica el tablespace que se utilizará para la creación de objetos temporales, Se toma por defecto el **TEMP**.

# Ejemplo

- Ejecutar previamente el siguiente comando;  
`alter session set "_ORACLE_SCRIPT"=true;`
- Create user **gestor** identified by **password**  
default tablespace system  
temporary tablespace temp;
- Create user **apellido** identified by **nombre**  
default tablespace system  
temporary tablespace temp;

Nota: apellido y nombre colocar sus datos

# Roles

- La creación de roles permite asignar permisos a usuarios, y poder modificar estos permisos sin tener que ir modificando uno por uno.

```
CREATE ROLE nombre_rol
```

```
{ [not identified | identified [by clave] }
```

# Ejemplo de Roles

Crear los siguientes Roles:

- `CREATE ROLE control_total;`
- `CREATE ROLE area_sistemas;`



# Privilegio

- Un privilegio representa un(os) permiso(s) dado(s) a un usuario para que realice cierta operación.

*Estos pueden ser de dos tipos:*

- **Privilegio del Sistema:** necesita el permiso de sistema correspondiente.
- **Privilegio sobre Objeto:** necesita el permiso sobre el objeto en cuestión.

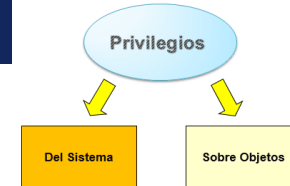
```
graph TD; A([Privilegios]) --> B[Del Sistema]; A --> C[Sobre Objetos]
```

**Privilegios**

**Del Sistema**

**Sobre Objetos**

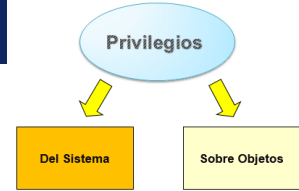
# Privilegios del Sistema



Rol predefinido	Descripción
CONNECT	Todos los permisos necesarios para iniciar una sesión en Oracle
RESOURCE	Todos los permisos necesarios para tener recursos para la creación de objetos
DBA	Todos los permisos para un administrador de base de datos (DBA)
EXP_FULL_DATABASE	Permisos para poder exportar toda la base de datos.
IMP_FULL_DATABASE	Permisos para poder importar toda la base de datos.

**Existen otros permisos ...** create user, create ..... o drop user

# Privilegios del Sistema



- Estos privilegios son permisos para realizar ciertas operaciones en la base de datos.
- El modo de asignar un privilegio es a través de la instrucción **GRANT** y de quitarlo a través de **REVOKE**.

# GRANT sobre Sistema

**GRANT [privilegio\_sistema | roles] TO**  
**[usuarios | roles | public]**

## Ejemplos:

- ✓ GRANT connect, resource TO apellido, area\_sistemas;
- ✓ GRANT create user TO apellido ;
- ✓ GRANT drop user TO apellido;
- ✓ GRANT dba TO control\_total;
- ✓ GRANT control\_total TO gestor;

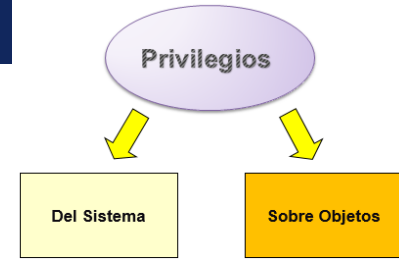
# REVOKE sobre Sistema

```
REVOKE [privilegio_sistema | roles]  
FROM [usuarios | roles |
```

## Ejemplos:

- REVOKE dba FROM control\_total;
- REVOKE drop user FROM apellido;
- REVOKE connect, resource FROM apellido;

# Privilegio Sobre Objetos



- Estos privilegios permiten que ciertos objetos (creados por un usuario) puedan ser tratado o accedidos por otros usuarios.
- El nivel de acceso depende del permiso que le demos. Entre ellos tenemos SELECT, UPDATE, DELETE, INSERT o todos ellos.
- El modo de asignación es a través de la instrucción GRANT y de cancelar REVOKE.

# CASO PRÁCTICO

---



# Crear Tablas

create table profesor

(pro\_cod char(3) not null primary key,  
pro\_nom varchar(15),  
pro\_ape varchar(20),  
pro\_tlf char(9)  
)

create table alumno

(alu\_cod char(5) not null primary key,  
alu\_nom varchar(15),  
alu\_ape varchar(20),  
alu\_fin date  
)



# GRANT sobre Objetos

**GRANT** [all | select | insert | update | delete ] **ON**  
objeto **TO** [usuarios | rol | public]

## Ejemplos:

- GRANT select ON profesor TO public;
- GRANT select, update, insert ON alumno TO apellido
- GRANT All ON profesor TO area\_sistemas;

## REVOKE sobre Objetos

```
REVOKE [all | select | insert | update | delete ] ON  
objeto FROM [usuarios | rol | public]  
{ WITH ADMIN OPTION }
```

### Ejemplos:

- REVOKE ALL ON profesor FROM area\_sistemas;
- REVOKE select ON alumno FROM apellido;