


INTRODUCCIÓN ADMINISTRACION BASES DE DATOS

ORACLE





Ingeniería de sistemas ocupa el segundo lugar de las carreras que ofrecen más trabajo en el país, con 3.100 ofertas mensuales, aproximadamente.

Las otras ingenierías presentes en el listado cierran el ranking de las diez primeras: Ingeniería de telecomunicaciones (1.250 ofertas) e Ingeniería de redes y telecomunicaciones (1.050 ofertas).

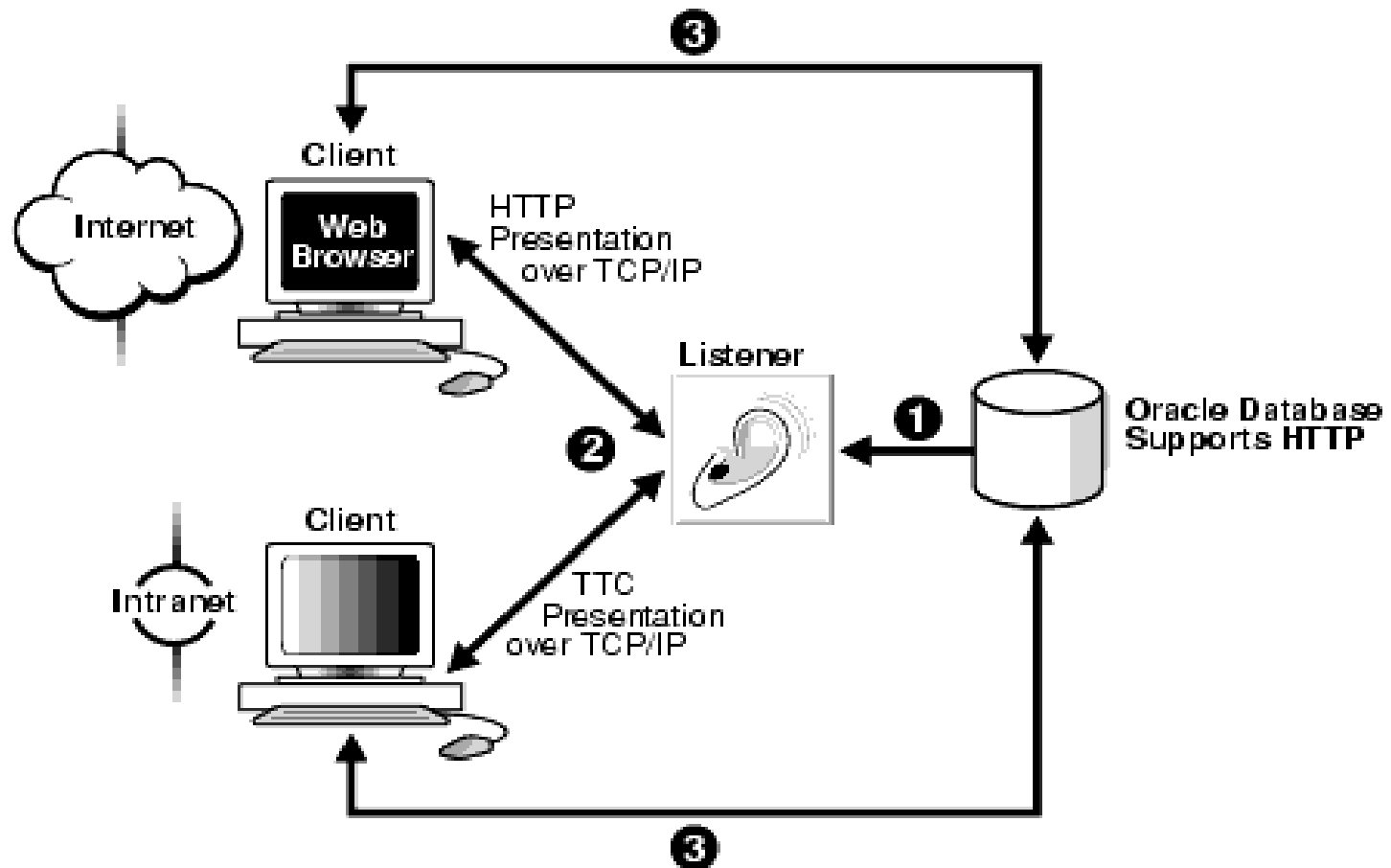
Que es una Base de Datos ?



Que es una Base de Datos Relacional?

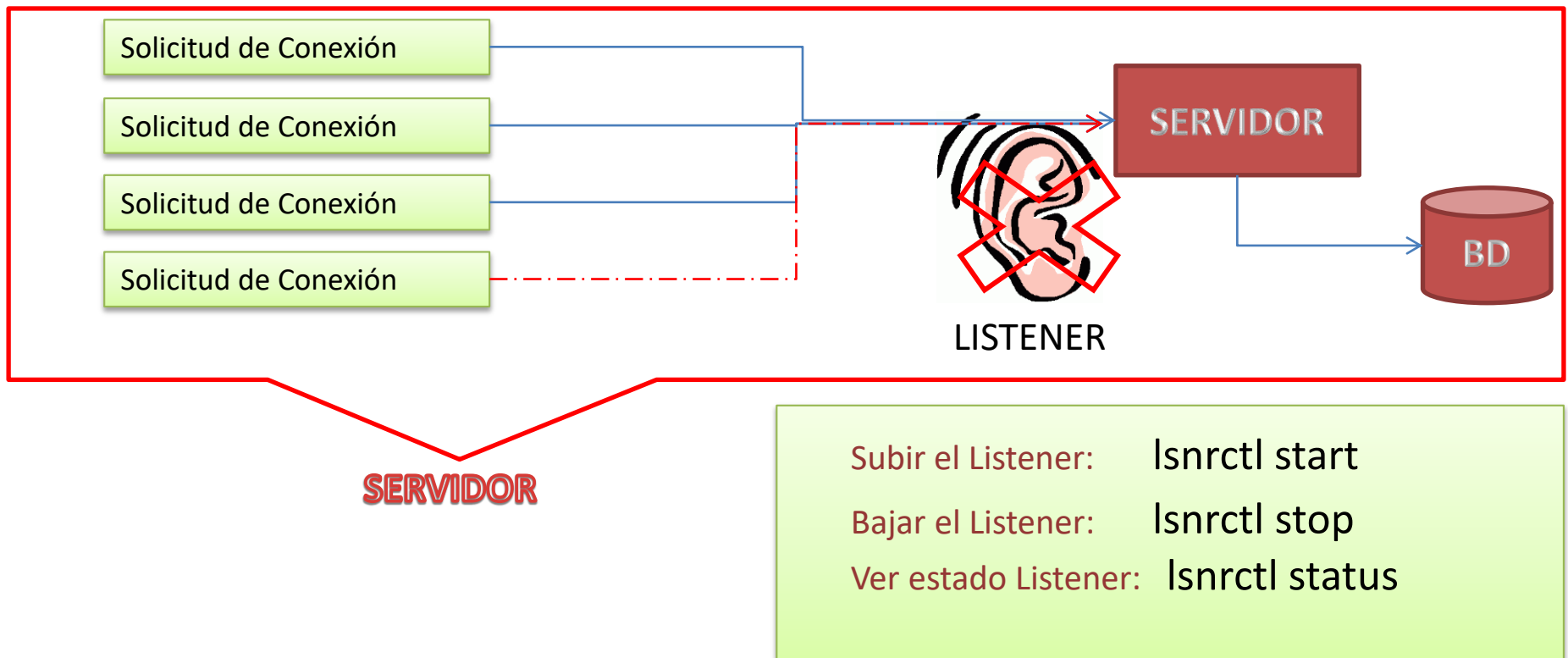
Que es un Sistema de Gestión de Base de Datos Objeto-Relacional ?

CONEXIÓN – ARQUITECTURA LISTENER

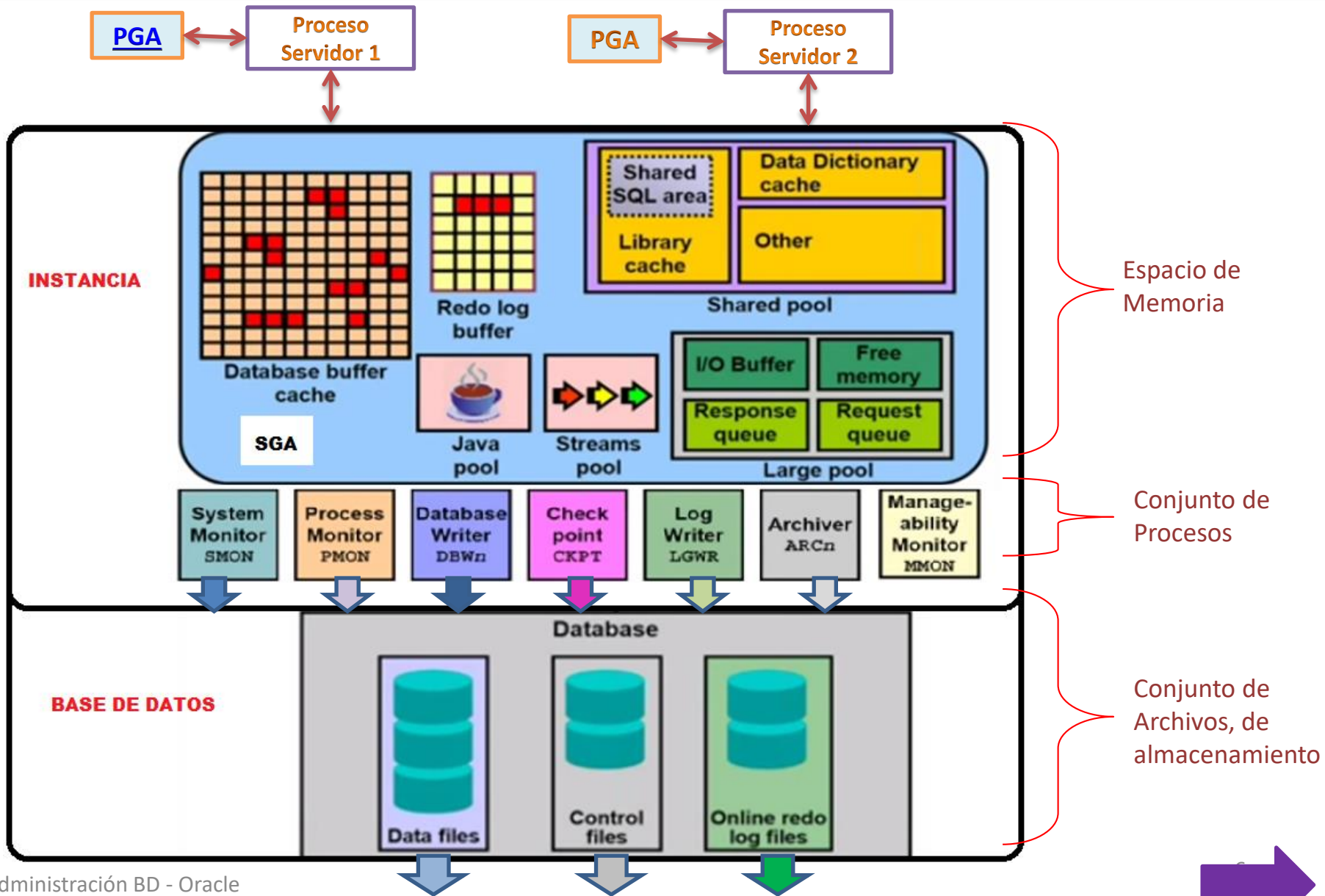


CONEXIÓN-ORACLE - LISTENER

Proceso servidor que provee la conectividad de red con la base de datos Oracle:



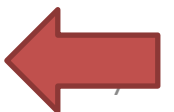
INSTANCIA Y BASE DE DATOS ORACLE



INSTANCIA – PGA (AREA GLOBAL DEL PROGRAMA)

Area de memoria utilizada por un proceso Oracle.

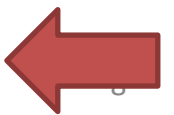
En esta memoria cada proceso almacena información que sólo es necesaria para su propio funcionamiento como por ejemplo sus variables globales, el estado actual de cada cursor (SQL) que se ejecuta... etc.



INSTANCIA – SMON (SYSTEM MONITOR)

Es el supervisor del sistema y se encarga de todas las recuperaciones que sean necesarias durante el arranque. Esto puede ser necesario si la BD se paró inesperadamente por fallo físico, lógico u otras causas.

Este proceso realiza la recuperación de la instancia de BD a partir de los ficheros redo log.



INSTANCIA – PMON (PROCESS MONITOR)



Este proceso restaura las transacciones no validadas de los procesos de usuario que abortan, liberando los bloqueos y los recursos de la SGA. Asume la identidad del usuario que ha fallado, liberando todos los recursos de la BD que estuviera utilizando, y anula la transacción cancelada.



INSTANCIA – DBWR (DATABASE WRITER)

Es el responsable de gestionar el contenido de los buffers de datos y del caché del diccionario. Él lee los bloques de los ficheros de datos y los almacena en la SGA. Luego escribe en los ficheros de datos los bloques cuyo contenido ha variado.

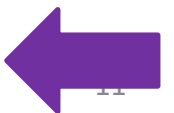
Es el único proceso que puede escribir en la BD.



INSTANCIA – CKPT (CHECKPOINT)



Este proceso escribe en los ficheros de control los checkpoints. Estos puntos de sincronización son referencias al estado coherente de todos los ficheros de la BD en un instante determinado, en un **punto de sincronización.**



INSTANCIA – LGWR (LOG WRITER)

Es el encargado de escribir los registros redo log en los ficheros redo log. Los registros redo log siempre contienen el estado más reciente de la BD, ya que puede que el DBWR deba esperar para escribir los bloques modificados desde el buffer de datos a los ficheros de datos.



INSTANCIA – ARCH (ARCHIVER)



El proceso archivador tiene que ver con los ficheros redo log. Por defecto, estos ficheros se reutilizan de manera cíclica de modo que se van perdiendo los registros redo log que tienen una cierta antigüedad. Cuando la BD se ejecuta en modo **ARCHIVELOG**, antes de reutilizar un fichero redo log realiza una copia del mismo.

- Un datafile es la representación física de un tablespace.
- Son los "ficheros de datos" donde se almacena la información físicamente.

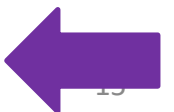


- Un fichero de control contiene entradas que especifican la estructura física de la BD.



- Se encuentra la siguiente información: nombre de la BD, nombre y localización de los ficheros de datos y ficheros de rehacer, fecha de creación de la base de datos.

- Se utiliza cada vez que se arranca la BD.



BD – REDO LOG FILES

- Archivos en un formato propietario que registra un historial de todos los cambios realizados en la base de datos.
- Protegen la base de datos de la pérdida de integridad en casos de fallos causados por suministro eléctrico, errores en discos duros.



ESTRUCTURAS DE ALMACENAMIENTO

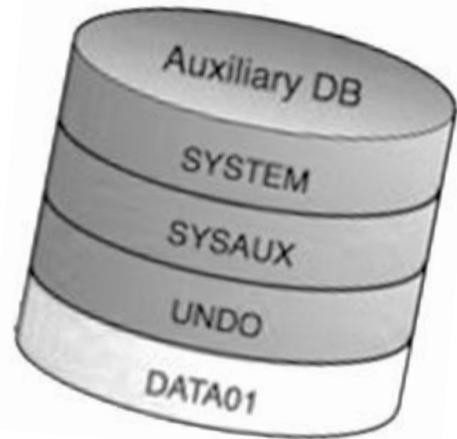
TERMINO	SIGNIFICADO
DataBase	Colección lógica de datos almacenados en TableSpaces.
File	Archivo de datos físicos ubicados en un solo TableSpace.
TableSpace	Repositorio lógico para agrupar datos físicamente.
Segment	Conjunto de uno o mas extents, que contiene todos los datos para una estructura específica contenida en un TableSpace.
Block	Bloque físico que localiza los datos existentes en un archivo (Es un componente de los archivos físicos, un archivo físico se compone de varios bloques).

Un tablespace es una unidad lógica de almacenamiento dentro de una bd oracle.

Es un puente entre el sistema de ficheros del sistema operativo y la base de datos.

Cada tablespace se compone de al menos, un datafile y un datafile solo puede pertenecer a un tablespace.





- **SYSTEM** = Diccionario de Datos.
- **UNDO** = Segmentos de Undo, almacena histórico de datos antiguos antes de modificar.
- **SYSAUX** = Datos auxiliares de las aplicaciones.
- **TEMP** = Complementa la PGA para aumentar la memoria.

BD – ADMINISTRACION TABLESPACES

Consultar tablespace existentes:

```
select tablespace_name, file_name, autoextensible  
from dba_data_files;
```

Como crear un tablespace:

```
Create tablespace tb_taller datafile 'c:\XE\dfiler_taller1.dbf' size 5M;
```

Aumentar el tamaño del Tablespace.

```
-Alter tablespace tb_taller add datafile 'c:\XE\dfiler_taller2.dbf' size 5M;  
-Alter database datafile 'c:\XE\dfiler_taller1.dbf' resize 20M;
```

Borrar un Tablespace

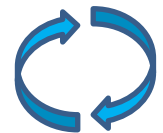
```
Drop tablespace tb_taller;
```

BD – ADMINISTRACION TABLESPACES

Ejercicio. Crear una tabla copia de la tabla cliente sobre el tablespace creado en el punto anterior, y poblarla hasta llenar los tablespace.

```
create table PERSONA_AUX  
(  
  numdoc          number(10),  
  tipodoc         char(2),  
  tipopersona     char(2),  
  Apellido3       varchar2(20),  
  nombre3         varchar2(40),  
  genero3         char(1)  
)  
tablespace TB_TALLER;
```

```
INSERT INTO PERSONA_AUX  
SELECT * FROM PERSONA;  
Commit;  
  
INSERT INTO PERSONA_AUX  
SELECT * FROM PERSONA_AUX;  
Commit;
```



```
-Alter database datafile 'c:\XE\dfiler_taller1.dbf' autoextend on next 5M;
```

ADMINISTRACION DE ESQUEMAS



ADMIN ESQUEMA - USUARIOS

CreSintaxis:

```
CREATE USER nomusuario IDENTIFIED BY claveusuario DEFAULT TABLESPACE  
defaultbespace TEMPORARY TABLESPACE temp;
```

Ejemplo:

```
CREATE USER user_consulta IDENTIFIED BY clave_consulta DEFAULT TABLESPACE  
tb_taller TEMPORARY TABLESPACE temp;
```

Asignación de Privilegios

Sintaxis GRANT:

```
GRANT CREATE privilegio TO usuario;  
GRANT ALTER privilegio TO usuario;  
GRANT DROP privilegio TO usuario;
```

Sintaxis REVOKE:

```
REVOKE CREATE privilegio FROM usuario;  
REVOKE ALTER privilegio FROM usuario;  
REVOKE DROP privilegio FROM usuario;
```

Ejemplo:

```
GRANT CREATE SESSION TO user_consulta ;
```

Algunos privilegios : CREATE session, CREATE table, CREATE view, CREATE procedure, CREATE synonym, ALTER table, ALTER view, ALTER procedure, ALTER synonym, DROP table, DROP view, DROP procedure, DROP synonym;

```
select * from system_privilege_map;
```

ADMIN ESQUEMA - USUARIOS

Actualizar datos de usuario Sintaxis:

Clave:

```
ALTER USER user_consulta IDENTIFIED BY consulta;
```

Tablespace:

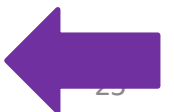
```
ALTER USER user_consulta DEFAULT TABLESPACE system  
TEMPORARY TABLESPACE temp;
```

Bloquear y desbloquear un Usuario:

```
ALTER USER user_consulta account lock;  
ALTER USER user_consulta account unlock;
```

Borrar Un usuario

```
DROP USER user_consulta CASCADE;
```



ADMIN ESQUEMA - ROLES

Un rol es una forma de agrupar permisos o privilegios para asignarlos luego a los usuarios.

Sintaxis:

```
CREATE ROLE nombrerol;
```

Ejemplos :

```
CREATE ROLE rol_consulta;
```

```
CREATE ROLE rol_modulo;
```

```
CREATE ROLE rol_total;
```

ADMIN ESQUEMA - ROLES

Asignación de ROL a usuarios

Sintaxis:

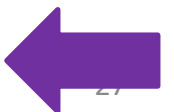
GRANT nombrerole **TO** {user, | role, | PUBLIC};

Ejemplos :

```
GRANT rol_consulta TO user_consulta;
```

```
GRANT rol_modulo TO PUBLIC;
```

```
GRANT rol_total TO ROL_XYZ;
```



ADMIN ESQUEMA - PERMISOS

Derecho a ejecutar un sentencia o a acceder a un objeto de otro usuario.

Los permisos se asignan a nivel de usuario o de rol.

Los permisos son fijos.

Sintaxis:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLA TO usuario/role/PUBLIC;
```

```
GRANT EXECUTE ON PROCEDIMIENTO/FUNCTION/PACKAGE TO usuario/role/PUBLIC;
```

Ejemplos :

```
GRANT SELECT ON PERSONA TO rol_consulta;
```

```
GRANT SELECT, INSERT, UPDATE ON PERSONA TO rol_modulo;
```

ADMIN ESQUEMA - PERMISOS

Se pueden otorgar permisos para que el que los reciba los pueda a su vez asignar a otros usuarios o roles. (Solo para asignación de usuarios directamente).

Sintaxis:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLA TO usuario WITH GRANT OPTION;  
GRANT EXECUTE ON PROCEDIMIENTO TO usuario WITH GRANT OPTION;
```

Ejemplos :

```
GRANT SELECT ON PERSONA TO user_consulta WITH GRANT OPTION;  
GRANT INSERT, UPDATE ON PROFESOR TO user_consulta WITH GRANT OPTION;
```

ADMIN ESQUEMA - PERMISOS

Cada tipo de objeto tiene su propio conjunto de permisos, los mas utilizados son:

Tables: select, insert, update, delete

Views: select, insert, update, delete

Sequence: select

Packages, Procedures, Functions (Java classes,...):
execute, debug

Directories: read, write



ADMIN ESQUEMA - SINONIMOS

Es una representación local o pública de un objeto perteneciente a un esquema.

Sirve para poder hacer referencia a aquel objeto sin tener que anteponer su esquema.

Un sinónimo público puede ser visto por todos los usuarios, pero uno privado, sólo por el usuario que lo creó.

Sintaxis:

```
CREATE SYNONYM nombresinonimo FOR propietario.objeto;
```

```
CREATE PUBLIC SYNONYM nombresinonimo FOR propietario.objeto;
```

Ejemplo:

```
CREATE SYNONYM PERSONA FOR USER_SQL.PERSONA;
```

```
CREATE PUBLIC SYNONYM PROFESOR FOR USER_SQL.PROFESOR;
```

Taller para desarrollar en clase:

- Desarrollarlo en grupos de dos personas.
- Entregar las sentencias en un solo archivo.
- En caso de evidencias enviar imágenes.