

Ejercicio SQL (SQL PLUS – CONSOLA)

1. Crea un usuario llamado “empleados” con clave “empleados” y concédele los permisos “connect” y “resource”. Conéctate desde ese usuario.

- ✓ **sqlplus sys as SYSDBA**
(y pulsar ENTER directamente cuando salga para poner el password)
`alter session set "_ORACLE_SCRIPT"=true;`
- ✓ **CREATE USER empleados IDENTIFIED BY oracle;**
si da error, escribir: `alter session set "_ORACLE_SCRIPT"=true;`
- ✓ **GRANT connect, resource TO empleados;**
- ✓ **CONNECT empleados;**
(y poner la password)

• Crea una tabla llamada “empleados_planta_valencia” con los siguientes campos:

- ✓ **CREATE TABLE EMPLEADOS_PLANTA_VALENCIA** (
 num_id number(5) **PRIMARY KEY**,
 nif varchar2(9) **UNIQUE NOT NULL**,
 nombre varchar2(10),
 apellido1 varchar2(10),
 apellido2 varchar2(10),
 fecha_nac date,
 salario number(5),
 direccion varchar2(30),
 telefono number(9),
 email varchar2(20) **UNIQUE**,
 iban number(20),
 millonario number(1)
);

• Crea el sinónimo “t_empval” para la tabla anterior.

- ✓ **GRANT CREATE SYNONYM TO empleados;** (hacerlo desde SYSDBA y volver)
- ✓ **CREATE SYNONYM t_empval FOR EMPLEADOS_PLANTA_VALENCIA;**

- Realiza un insert y un select de todos los datos utilizando el sinónimo

✓ **INSERT INTO** *t_empval*

VALUES (1, '10203040', 'Pepe', 'Lopez', 'Gomez', '05012021', 900, 'Avenida De Las Ciencias 55 4B', 954404875, 'sergiodiaz@gmail.com', 2030607090, 1);

✓ **SELECT * FROM** *t_empval*;

- Crea el índice “nombrecompleto” que contenga las columnas nombre, apellido1 y apellido2. ¿En qué casos se utilizaría?

✓ **CREATE INDEX** *nombrecompleto* **ON** *t_empval* (nombre, apellido1, apellido2);

- Crea el índice del campo email. Si no se puede, indica el motivo.

- ¿Sería apropiado crear un índice de la columna “millonario”?

Sería recomendable, ya que si hay pocos millonarios, podríamos tener una tabla con los pocos que son millonarios, y así les encontraríamos mas rápido.

- Crea una secuencia llamada “numero_id_planta_valencia” con las siguientes características:

- Que incremente de 1 en 1
- Que empiece por 10000
- Que termine en 99999
- Que no sea cíclico

✓ **CREATE SEQUENCE** *numero_id_planta_valencia*

INCREMENT BY 1

START WITH 10000

MAXVALUE 99999

NOCYCLE;

- Crea una secuencia que se llame “num_iban_planta_valencia” con las siguientes características:

- Que incremente de 5 en 5
- Que empiece por “20000000000000000000”
- Que termine en “99999999999999999999”
- Que no sea cíclica

✓ **CREATE SEQUENCE** *num_iban_planta_valencia*
INCREMENT BY 5
START WITH 20000000000000000000
MAXVALUE 9999999999999999999
NOCYCLE;

- Modificar la secuencia anterior para que el incremento sea de 10 en 10.

✓ **ALTER SEQUENCE** *num_iban_planta_valencia*
INCREMENT BY 10;

- Obtén el valor que se asignará cuando se ejecute la sentencia **NEXTVAL** en esta secuencia.

✓ **SELECT** *num_iban_planta_valencia.NEXTVAL* **FROM DUAL**;

Nota: cuando aún no se ha asignado ningún valor, el CURRVAL será NULL y el NEXTVAL será el 1

- Crea una vista sobre la tabla que hemos trabajado, que se llame “v_empval” y sea el resultado de seleccionar los siguientes campos de la tabla:

- nif
- nombre
- apellido1
- apellido2
- teléfono

✓ **CREATE VIEW** *v_empval* **AS SELECT** nif, nombre, apellido1, apellido2, telefono **FROM** *t_empval*;

Nota: también podría usarse el CREATE OR REPLACE VIEW para crear otra vista diferente pero con el mismo nombre de otra

- Realiza una selección sobre esa vista.

✓ **SELECT *** **FROM** *v_empval*;

- ¿Qué utilidades encontrarías a la creación de tablas en este caso?

La privacidad y seguridad (además del rendimiento en la búsqueda) en los datos de algunos campos de especial carácter sensible de cara a la manipulación por otra persona que no sea de la confianza del jefe del dpto. de informática de la empresa