

TEMAS 6 Y 7:

--1. Crea la tabla PROVINCIAS con los siguientes campos

Cod_provi de tipo number(2) y es la clave primaria

Nombre de tipo varchar2(25), es obligatorio

Pais de tipo varchar2(25)

debe ser uno de los siguientes España, Portugal o Italia

Introduce la provincia 1 Ávila de España.

***/**

```
CREATE TABLE PROVINCIAS(  
  Cod_provi NUMBER(2) PRIMARY KEY,  
  Nombre VARCHAR2(25) NOT NULL,  
  Pais VARCHAR2(25),  
  CONSTRAINT ck_provincias_pais CHECK(Pais In('España', 'Portugal', 'Italia'))  
);  
INSERT INTO PROVINCIAS(Cod_provi, Nombre, Pais) VALUES(01, 'Ávila', 'España');
```

/*2. Crea la tabla EMPRESAS con los siguientes campos:

Cod_empre number(2) es la clave

Nombre varchar2(25) obligatorio por defecto será empresa1

Fecha_crea de tipo fecha por defecto será un día posterior a la fecha actual.

Introduce la empresa 20 llamada Alfa21 S.A.

***/**

```
CREATE TABLE EMPRESAS(  
  Cod_empre NUMBER(2) PRIMARY KEY,  
  Nombre VARCHAR2(25) DEFAULT 'empresa1' NOT NULL,  
  Fecha_crea DATE DEFAULT(SYSDATE + 1)  
);  
INSERT INTO EMPRESAS(Cod_empre, Nombre) VALUES(20, 'Alfa21 S.A.');
```

```
SELECT * FROM EMPRESAS;
```

3. Crea la tabla CONTINENTES con los siguientes campos

Cod_conti de tipo number y es la clave primaria

Nombre de tipo varchar2(20) el valor por defecto es EUROPA y es obligatorio */

```
CREATE TABLE CONTINENTES(  
  Cod_conti NUMBER(2) PRIMARY KEY,  
  Nombre VARCHAR2(20) DEFAULT 'EUROPA' NOT NULL  
);
```

4. Crea la tabla **ALUMNOS** con los siguientes campos

codigo number(3) y es la clave primaria **nombre** cadena de caracteres de longitud máxima 21, es obligatorio

apellido cadena de caracteres de longitud máxima 30, es obligatorio y ha de estar en mayúsculas.

Curso de tipo number y ha de ser 1, 2 o 3

Fecha_matri de tipo fecha y por defecto es la fecha actual. */

```
CREATE TABLE ALUMNOS(  
    codigo NUMBER(3) PRIMARY KEY,  
    nombre VARCHAR2(21) NOT NULL,  
    apellido VARCHAR2(30) NOT NULL CHECK (curso upper(apellido) = apellido),  
    Curso NUMBER(1) CHECK(curso IN(1, 2, 3)),  
    Fecha_matri DATE DEFAULT(SYSDATE)  
);
```

```
INSERT INTO ALUMNOS (codigo, nombre, apellido, Curso) VALUES(11, 'Pepe',  
'SANCHO', 2);  
SELECT * FROM ALUMNOS;
```

5. Crea la tabla **EMPLEADOS** con los siguientes campos

Cod_emple number(2) y es clave

Nombre cadena de caracteres de longitud máxima 20 y es obligatorio

Apellido cadena de caracteres de longitud máxima 25

Salario número de 7 cifras con dos decimales debe ser mayor que 0

Además tiene dos campos que son claves ajenas de las tablas provincias y empresas respectivamente.

Para la clave ajena de provincias indicaremos un borrado en cascada.

```
CREATE TABLE EMPLEADOS(  
    Cod_emple NUMBER(2) PRIMARY KEY,  
    Nombre VARCHAR2(20) NOT NULL,  
    Apellido VARCHAR2(25),  
    Salario NUMBER(7, 2) CHECK(Salario > 0),  
    Cod_provincia NUMBER(2),  
    Cod_empresa NUMBER(2),  
    CONSTRAINT CA_provincia_empleado FOREIGN KEY(Cod_provincia)  
    REFERENCES PROVINCIAS ON DELETE CASCADE,  
    CONSTRAINT CA_empresa_empleado FOREIGN KEY(Cod_empresa)  
    REFERENCES EMPRESAS  
);
```

Introduce en la tabla EMPLEADOS, la empleada 300 de nombre Veronica Lopez y salario 3000 ¿Qué ocurre?¿Por qué?

```
INSERT INTO EMPLEADOS(Cod_emple, Nombre, Apellido, Salario )  
VALUES(300,'Veronica', 'Lopez', 3000);
```

/*

**Introduce los mismos datos que antes pero el código del empleado será 30.
Introduce en la tabla EMPLEADOS el empleado 10 llamado Sergio Perez que trabaja para la empresa de código 10 en la provincia 1 ¿Qué ocurre?¿Porqué?**

*/

```
INSERT INTO EMPLEADOS(Cod_emple, Nombre, Apellido, Salario )  
VALUES(30,'Veronica', 'Lopez', 3000);
```

/*

Introduce en la tabla EMPLEADOS el empleado 10 llamado Sergio Perez que trabaja para la empresa Alfa21 y vive en Ávila.

*/

```
INSERT INTO EMPLEADOS(Cod_emple, Nombre, Apellido, Cod_provincia,Cod_empresa)  
VALUES(10,'Sergio', 'Perez', 1, 20);
```

6. Crea la tabla PROVINCIAS y PERSONAS con la estructura que se muestra a continuación, en negrita la clave principal y codprovin referencia a cod_provincia. Además pondremos la opción de BORRADO EN CASCADA.

```
CREATE TABLE PROVINCIAS(  
    cod_provincia VARCHAR2(2) PRIMARY KEY,  
    nom_provincia VARCHAR2(15),  
    poblacion VARCHAR2(15)  
);
```

```
CREATE TABLE PERSONAS(  
    dni VARCHAR2(10) PRIMARY KEY,  
    nombre VARCHAR2(30),  
    direccion VARCHAR2(30),  
    codprovin VARCHAR2(2),  
    CONSTRAINT CA_provincias FOREIGN KEY (codprovin)  
    REFERENCES PROVINCIAS ON DELETE CASCADE  
);
```

**7. Crear la tabla EJEMPLO1 y asignar a la columna fecha la fecha del sistema
DNI VARCHAR2(10)**

NOMBRE VARCHAR2(30)

EDAD NUMBER(2)

FECHA DATE

Insertar una fila con los valores siguientes, 1234, PEPA, 21 sin la columna fecha, comprobando después que efectivamente añade la fecha actual

```
CREATE TABLE EJEMPLO1(  
  dni VARCHAR2(10) PRIMARY KEY,  
  nombre VARCHAR2(30),  
  edad NUMBER(2),  
  fecha DATE DEFAULT(SYSDATE)  
);  
INSERT INTO EJEMPLO1(dni, Nombre, edad) VALUES(1234, 'PEPA', 21);  
SELECT * FROM EJEMPLO1;
```

-- 13) ¿Cuáles son los nombres de los clientes que viven en la provincia de MADRID?

```
SELECT NOMBRE, PROVINCIA  
FROM CLIENTES  
WHERE PROVINCIA='MADRID';
```

-- 14) Mostrar el nombre y apellido de los clientes que (una consulta para cada punto):

-- viven en la provincia de zMADRIDz

```
SELECT NOMBRE, APELLIDOS  
FROM CLIENTES  
WHERE PROVINCIA='MADRID';
```

-- viven en Belmontez

```
SELECT NOMBRE, APELLIDOS  
FROM CLIENTES  
WHERE CIUDAD='Belmonte';
```

-- tienen como código postalz 45915

```
SELECT NOMBRE, APELLIDOS  
FROM CLIENTES  
WHERE COD_POSTAL='45915';
```

-- que hicieron pedidos en Mayo

```
SELECT CLIENTES.NOMBRE, CLIENTES.APELLIDOS
```

```
FROM CLIENTES, PEDIDOS
```

```
WHERE PEDIDOS.FECHA_PEDIDO > to_date('30/04/08', 'DD/MM/YY')  
AND PEDIDOS.FECHA_PEDIDO < to_date('1/06/08', 'DD/MM/YY')  
AND CLIENTES.ID_CLIENTE=PEDIDOS.ID_CLIENTE;
```

-- 15) ¿Hay pedidos en los que la fecha de envío fue puesta accidentalmente anterior a la fecha -- de pedido?

```
SELECT *
```

FROM PEDIDOS

WHERE FECHA_PEDIDO > FECHA_ENVIO;

-- 16) Mostrar el nombre y apellidos de los clientes cuyos apellidos empiezan por 'Pe'.

SELECT NOMBRE, APELLIDOS

FROM CLIENTES

WHERE APELLIDOS LIKE 'Pe%';

17) Mostrar el nombre y dirección de los proveedores cuya dirección incluya la cadena 'del'.

SELECT NOMBRE, DIRECCION

FROM PROVEEDORES

WHERE DIRECCION LIKE '%del%';

-- 18) Listar el nombre y apellidos de los clientes que viven en Somosierra y cuyo apellido

-- empieza con la letra 'S'.

SELECT NOMBRE, APELLIDOS

FROM CLIENTES

WHERE CIUDAD='Somosierra' AND APELLIDOS LIKE 'S%';

-- 19) Listar el nombre y apellidos de los clientes que viven en Somosierra o en la provincia de ORENSE.

SELECT NOMBRE, APELLIDOS

FROM CLIENTES

WHERE CIUDAD='Somosierra' OR provincia='ORENSE';

-- 20) Mostrar una lista de los nombres y número de teléfono de los proveedores de las-- provincias de MADRID y CACERES.

SELECT NOMBRE, TELEFONO

FROM PROVEEDORES

WHERE PROVINCIA='MADRID' OR PROVINCIA='CACERES';

-- 21) Mostrar los datos de los pedidos del cliente 1001 en los que las fechas de pedido y envío coincidan.

SELECT *

FROM PEDIDOS

WHERE ID_CLIENTE=1001 AND FECHA_PEDIDO = FECHA_ENVIO;

-- 22) Mostrar los datos de los pedidos del cliente 1001 o que cumplan que la fecha de envío es 4 días posterior a la fecha de pedido.

SELECT *

FROM PEDIDOS

WHERE ID_CLIENTE=1001 AND FECHA_PEDIDO+4 = FECHA_ENVIO;

-- 23) Mostrar nombre, apellidos, provincia y código postal de los clientes que se apelliden

Pelayo en la provincia de CACERES o de los clientes cuyo código postal termine en 9.

SELECT NOMBRE, APELLIDOS, PROVINCIA, COD_POSTAL

FROM CLIENTES

WHERE APELLIDOS='Pelayo' AND PROVINCIA='CACERES' OR COD_POSTAL LIKE '%9';

-- 24) Mostrar nombre, apellidos, provincia y código postal de los clientes que se apelliden -- Pelayo y de cualquier otro cliente que viva en la provincia de CACERES o tenga un código--postal que termine en 9.

SELECT NOMBRE, APELLIDOS, PROVINCIA, COD_POSTAL
FROM CLIENTES

WHERE APELLIDOS='Pelayo' OR PROVINCIA='CACERES' OR COD_POSTAL LIKE '%9';

-- 25) Mostrar los datos de los proveedores que tienen su sede en CACERES, ORENSE o MADRID.

select *

from proveedores

where provincia IN ('CACERES', 'ORENSE', 'MADRID');

-- 26) Listar los clientes cuyo apellido empieza por H.

select *

from clientes

where apellidos like 'H%';

-- 27) Listar los clientes que no viven en Robledo ni en Somosierra.

select *

from clientes

where CIUDAD NOT IN ('Robledo', 'Somosierra');

-- 28) Mostrar el número de pedido, el id_cliente y la fecha de pedido de todos los pedidos que-- ha realizado el cliente 1001 ordenado por fecha de pedido.

select id_cliente, numero_pedido, FECHA_PEDIDO

from pedidos

where id_cliente=1001 order by fecha_pedido;

-- 29) Mostrar un listado en orden alfabético de los nombres de productos que empiezan por-- 'Dog'.

select nombre

from productos

where nombre like 'Dog%' order by nombre;

-- 30) Listar los nombres de todos los proveedores con sede en Batres, Belmonte o Robledo.

select nombre

from proveedores

where ciudad in ('Batres', 'Belmonte', 'Robledo');

-- 31) Mostrar en orden alfabético la lista de los nombres de productos cuyo precio de venta sea-- igual o superior a 125.00 euros.

select nombre

from productos

where precio_venta >= 125.00;

-- 32) Listar en orden alfabético los nombres de los proveedores que no tienen página web.

select nombre

from proveedores

where pag_web is null order by nombre;