

BASES DE DATOS – GUÍA DE REPASO

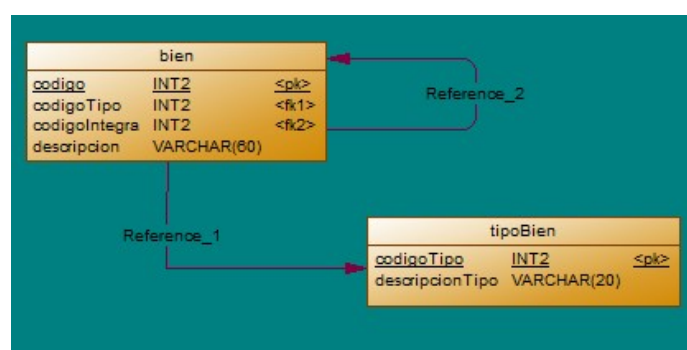
1. Presentar un escenario práctico donde se implemente algún mecanismo de seguridad, propio de los Sistemas de Gestión de Bases Datos (DBMS).
2. ¿Cómo se denomina al subconjunto de comandos SQL que permiten crear, modificar y destruir objetos de una base de datos? ¿Cuáles son estos comandos?
3. Y cómo se denomina al subconjunto de comandos SQL que sirven para recuperar, actualizar y eliminar datos? ¿Cuáles son estos comandos?
4. ¿Qué es el catálogo o diccionario de datos?
5. Nombrar qué comando debo escribir si necesito:
 - a. Agregar una columna a una tabla existente
 - b. Actualizar todas las filas de una tabla
 - c. Eliminar algunas filas de una tabla
 - d. Insertar filas de una tabla tomado datos de otra
 - e. Cambiar los valores permitidos de una *check constraint*.
6. Suponga la siguiente tabla, existente en una base de datos:



- a. ¿Qué comandos debería ejecutar si se desea cambiar el *check constraint* y el tipo de dato del atributo *tipoBien*, pasando a ser un *smallint* cuyos valores posibles sean:

Tipo	Descripción
1	Producto
2	Servicio
3	Adicionales

- b. Qué comandos serían necesarios para adaptar la estructura a este nuevo modelo:



7. Cómo resolvería el problema planteado en el punto anterior si la tabla *bien* tuviera valores cargados?

Por ejemplo:

Código	Descripción	Tipo
1	Bicicleta	Producto
2	Cubierta	Producto
3	Cámara	Producto
4	Lavado y engrase	Servicio
5	Reparación cubierta	Servicio

Plantear que comandos debería escribir y en qué orden, para mantener tanto la estructura como los datos preexistentes.

7. Dado el siguiente script de creación de tablas:

```
CREATE TABLE provincia (  
    id bigint NOT NULL,  
    codigo integer NOT NULL,  
    descripcion character varying(50) NOT NULL,  
    CONSTRAINT provincia_pk PRIMARY KEY (id),  
    CONSTRAINT provincia_codigo_uk UNIQUE (codigo)  
);  
  
CREATE TABLE localidad (  
    id bigint NOT NULL,  
    id_provincia bigint NOT NULL,  
    codigo integer NOT NULL,  
    descripcion character varying(50) NOT NULL,  
    codigo_postal integer NULL,  
    CONSTRAINT localidad_pk PRIMARY KEY (id),  
    CONSTRAINT localidad_codigo_uk UNIQUE (id_provincia, codigo),  
    constraint fk_localidad_provincia foreign key (id_provincia) references  
provincia(id)  
);  
  
CREATE TABLE sucursal (  
    id bigint NOT NULL,  
    codigo integer NOT NULL,  
    descripcion character varying(50) NOT NULL,  
    domicilio varchar(250) null,  
    id_localidad bigint null,  
    CONSTRAINT sucursal_pk PRIMARY KEY (id),  
    CONSTRAINT sucursal_codigo_uk UNIQUE (codigo),  
    constraint fk_sucursal_localidad foreign key (id_localidad) references  
localidad(id)  
);
```

Nota: en una publicación anterior de este ejercicio, faltaron incluir por error las *foreign keys*.

- Dibujar el modelo físico de datos correspondiente.
- Convertir el modelo físico de datos planteado con claves subrogadas (id) a una implementación con claves de negocio.