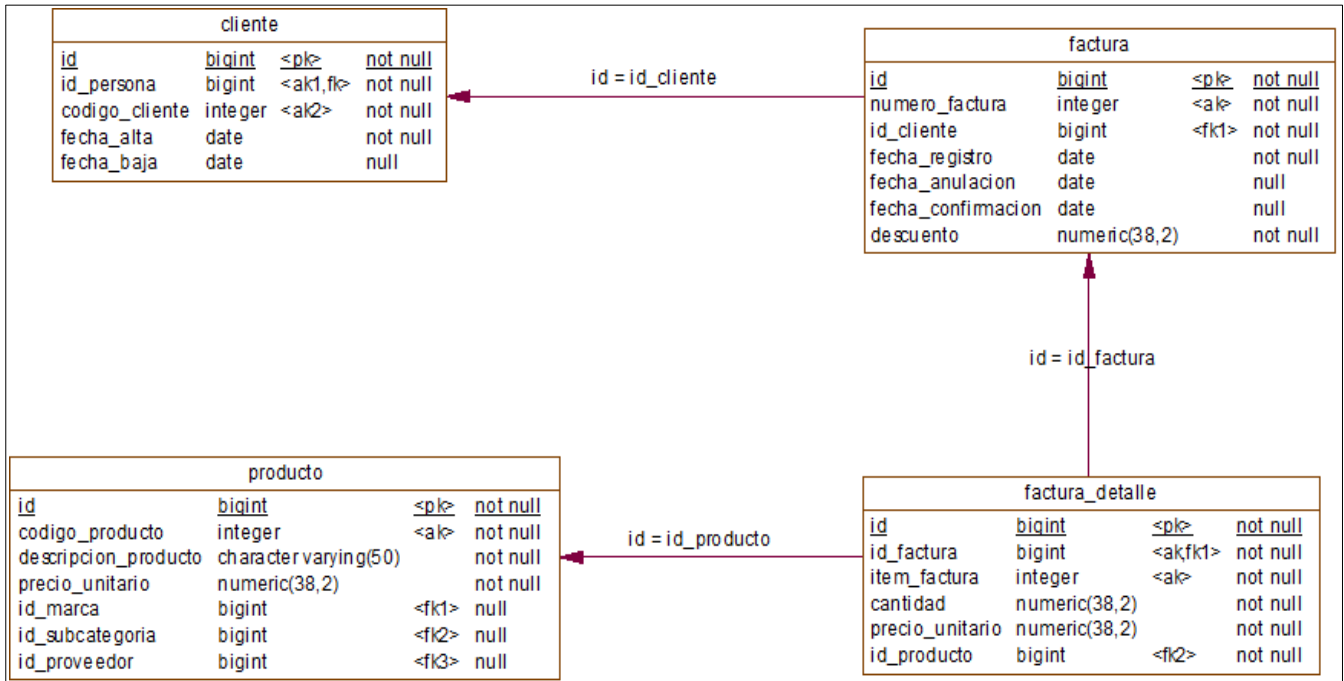
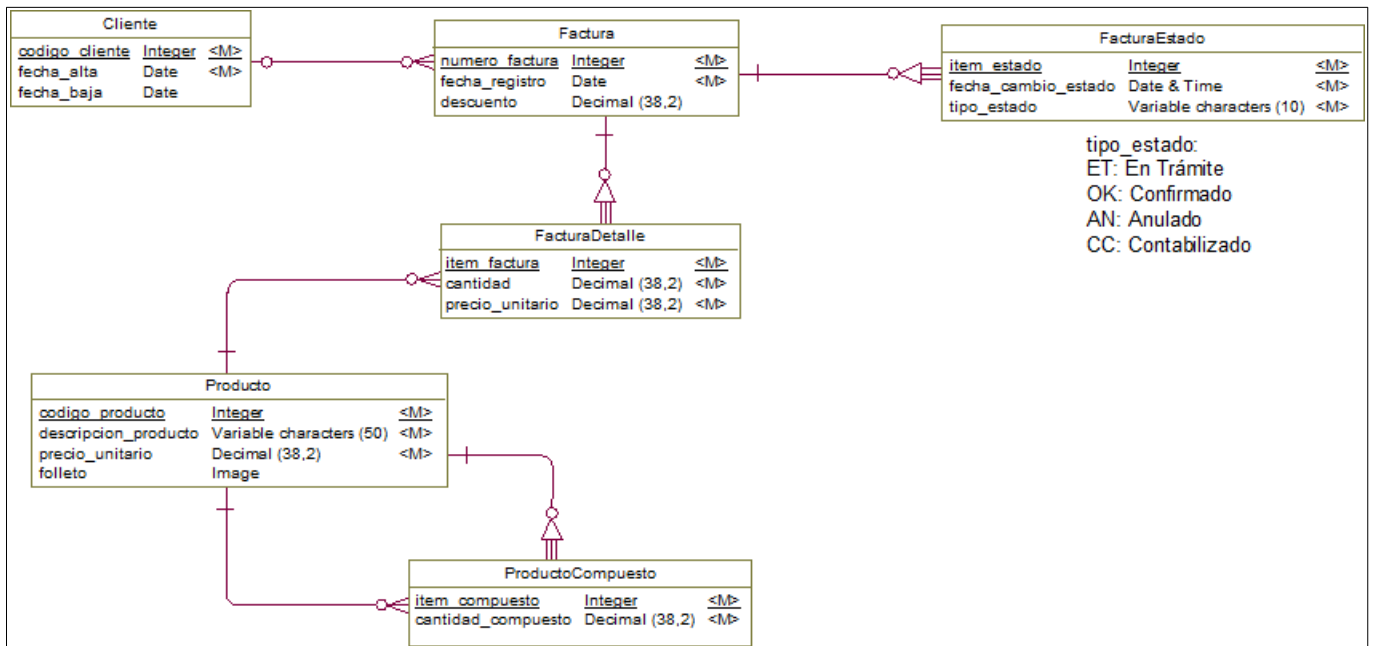


PRÁCTICA

Dado el siguiente modelo físico de datos:



Se desea adaptar el modelo físico a los siguientes cambios realizados en el modelo conceptual:



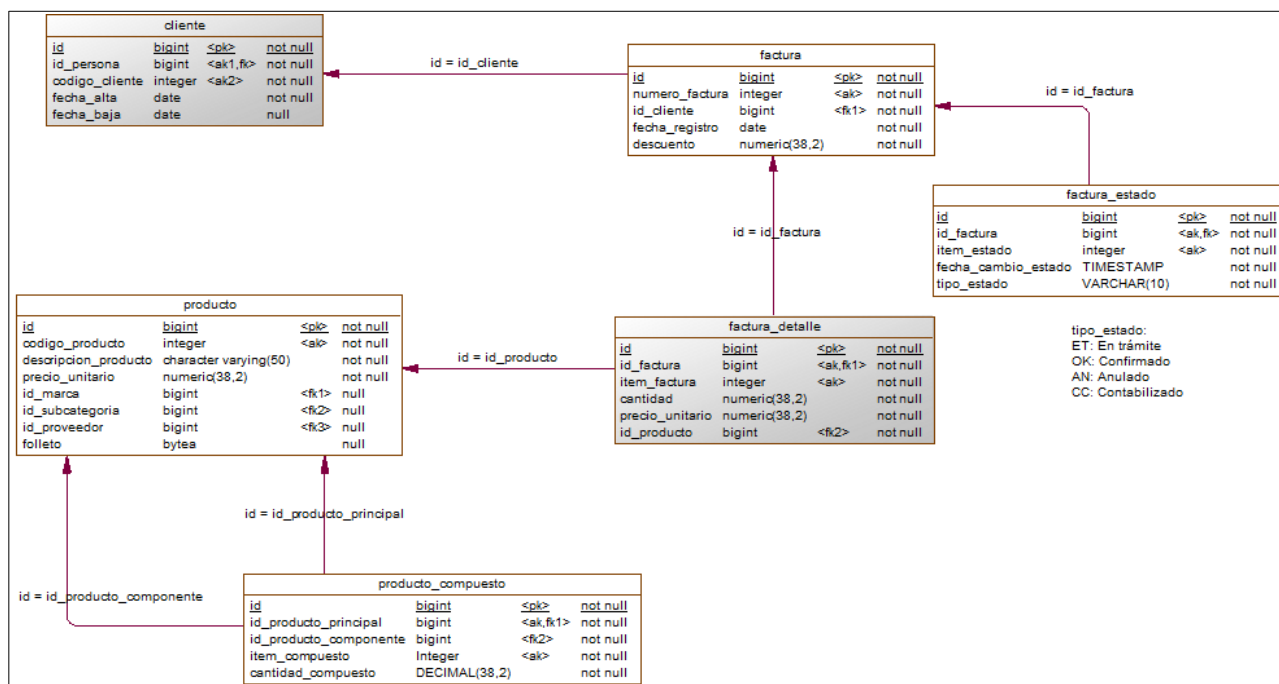
1. Completar el modelo físico con las nuevas tablas y relaciones derivadas del MCD.
2. Codificar los scripts necesarios para:
 - a. Crear la tabla derivada de la entidad *FacturaEstado*.
 - b. Modificar la tabla *factura*.
 - c. Crear la tabla derivada de la entidad *ProductoCompuesto*.
 - d. Modificar la tabla *producto*.
3. Para optimizar el rendimiento y la organización de los datos, se ha decidido dividir la tabla *producto* en dos tablas separadas: una para el uso frecuente del sistema de gestión y otra para la información de folletería. Se debe crear una nueva tabla llamada *producto_folleto* para almacenar exclusivamente la información de folletería y lograr una mejor organización de los datos. Codificar:
 - a. Sentencia "CREATE TABLE" de la tabla *producto_folleto* para contener las imágenes de los productos.
 - b. Comando que permita trasladar los datos hacia la nueva tabla. Si no recuerda el comando, describa la acción a realizar.
 - c. Sentencia "ALTER TABLE" para eliminar de la tabla *producto* la columna *folleto*.

TEORÍA

1. ¿Qué es el manejador de bases de datos? ¿Cuáles son sus principales tareas?
2. Definir las restricciones de “Primary key” y “Unique”. Describir similitudes y diferencias.
3. ¿Para qué usa el sistema de gestión de bases de datos el diccionario de datos al ejecutar una sentencia de manejo de datos (DML)?

SOLUCIÓN PRÁCTICA

Ejercicio 1



Ejercicio 2

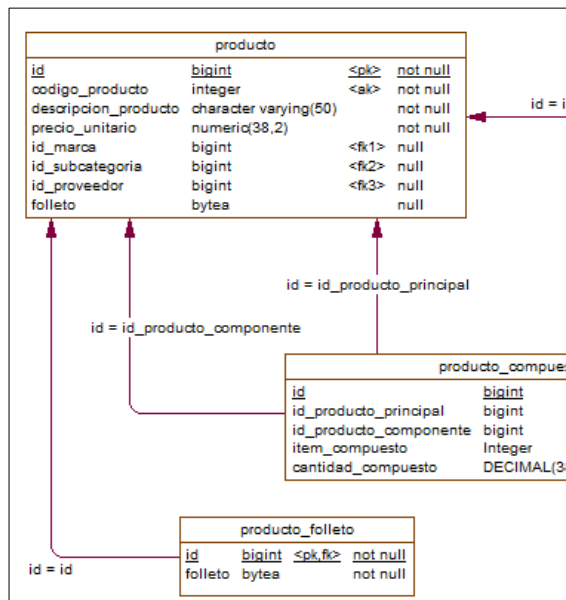
```
-- Tabla nueva: factura_estado
create table factura_estado (
    id                bigint                not null,
    id_factura        bigint                not null,
    item_estado       integer              not null,
    fecha_cambio_estado  timestamp         not null,
    tipo_estado       varchar(10)          not null,
    constraint PK_FACTURA_ESTADO primary key (id),
    constraint UK_FACTURA_ESTADO unique (id_factura, item_estado),
    constraint CKC_TIPO_ESTADO_FACTURA check (tipo_estado in ('ET','OK','AN','CC')),
    constraint FK_FACTURA_ESTADO_FACTURA foreign key (id_factura)
        references factura (id)
);

-- Columnas de factura que no se usan mas
alter table factura drop column fecha_anulacion;
alter table factura drop column fecha_confirmacion;

-- Tabla nueva: producto_compuesto
create table producto_compuesto (
    id                bigint                not null,
    id_producto_principal bigint          not null,
    id_producto_componente bigint        not null,
    item_compuesto     integer              not null,
    cantidad_compuesto decimal(38,2)      not null,
    constraint PK_PRODUCTO_COMPUESTO primary key (id),
    constraint UK_PRODUCTO_COMPUESTO unique (id_producto_principal, item_compuesto)
);

alter table producto_compuesto
    add constraint FK_PRODUCTO_COMP_PRODUCTO_PPAL foreign key (id_producto_principal)
        references producto (id);
alter table producto_compuesto
    add constraint FK_PRODUCTO_COMP_PRODUCTO_COMP foreign key (id_producto_componente)
        references producto (id);
-- Agrego columna a producto
alter table producto add column folleto bytea;
```

Ejercicio 3



```

create table producto_folleto (
    id                bigint                not null,
    folleto           bytea                not null,
    constraint PK_PRODUCTO_FOLLETO primary key (id),
    constraint FK_PRODUCTO_FOLLETO_PRODUCTO foreign key (id)
        references producto (id)
);
    
```

/ Copiar los registros que tengan valor en la columna folleto desde la tabla producto, hacia la tabla producto_folleto */*

```

INSERT INTO producto_folleto
SELECT id, folleto
FROM producto
WHERE folleto IS NOT NULL;
    
```

```

ALTER TABLE producto DROP COLUMN folleto;
    
```

TEORÍA

1. ¿Qué es el manejador de bases de datos? ¿Cuáles son sus principales tareas?

Apunte Unidad I – Sistemas de manejo de Base de datos, Página 7.

El manejador de base de datos (también conocido como *Sistema de Gestión de Bases de Datos* - SGBD o DBMS en inglés-) es un programa (o conjunto de programas) que constituye la interfaz entre los datos de bajo nivel almacenados físicamente en la base de datos con los programas de aplicaciones y las consultas hechas al sistema. El manejador de base de datos es el responsable de las siguientes tareas:

- a. Interacción con el manejador de archivos.
- b. Implantación de la integridad.
- c. Puesta en práctica de la seguridad.
- d. Respaldo y recuperación.
- e. Control de concurrencia.

2. Definir las restricciones de “Primary key” y “Unique”. Describir similitudes y diferencias.

Presentaciones de clase Tema III – DDL, Integridad Declarativa. Unique Constraint. Primary key constraint.

"Primary Key":

La restricción Primary Key (clave primaria) identifica de manera única cada fila en una tabla de la base de datos. Las características principales de una clave primaria son:

- Unicidad: Los valores en la columna (o conjunto de columnas) definidas como clave primaria deben ser únicos para cada fila.
- No permite valores nulos
- Solo puede existir una clave primaria por tabla.
- Puede estar compuesta por varias columnas (clave compuesta).

"Unique":

La restricción Unique garantiza que los valores en una columna específica, o en un conjunto de columnas, sean únicos dentro de la tabla. Sus características son:

- Unicidad: Similar a la clave primaria
- Permite un valor nulo: A diferencia de la clave primaria, una columna UNIQUE puede contener un valor NULL, ya que SQL considera NULL como no comparable.
- Se pueden definir múltiples restricciones UNIQUE en una misma tabla.

Similitudes entre Primary Key y Unique:

Ambas restricciones garantizan la unicidad

Índice automático: generan índices automáticamente para garantizar la unicidad y optimizar consultas.

Diferencias:

PK no permite valores nulos; UK permite un único valor nulo

Cantidad en la tabla: solo un PK; puede tener varios UK.

3. ¿Para qué usa el sistema de gestión de bases de datos el diccionario de datos al ejecutar una sentencia de manejo de datos (DML)?

Apunte Unidad III. Página 1. Apunte Unidad I. Páginas 9 y 10. Charlado en clase

1. Validar la sintaxis/semántica: El motor revisa si la sentencia DML cumple con las reglas establecidas por la estructura de la base de datos. Por ejemplo, que la tabla referenciada exista, si se hace referencia específica a columnas que las mismas existan, si se usa "*" obtener las columnas de la tabla, que los tipos de datos utilizados sean correctos, etc.

2. Verificar restricciones y garantizar integridad: consulta el diccionario de datos para validar que se respeten restricciones de integridad, como claves primarias, foráneas, chequeos o restricciones de unicidad, cuando se modifican/insertan/borran datos.