# TrabajoPráctico Nro2

1. Crear tablas de Nivel 0.
2. Crear tablas de Nivel 1.
3. Definir para todos los casos restricciones por Unique, Not NULL, Check, claves foráneas y claves primarias.

**Tablas de nivel 0**

CREATE TABLE PAISES

(

PAIS INT,

descripcion VARCHAR(100)NOTNULL,

CONSTRAINT PK\_PAIS PRIMARY KEY (PAIS),

CONSTRAINT UNIQUE\_DESC UNIQUE (descripcion)

)

CREATE TABLE PRODUCTOS

(

PRODUCTO INT,

descripcion VARCHAR(100)NOTNULL,

precio FLOATNOTNULL,

CONSTRAINT PK\_PRODUCTO PRIMARY KEY (PRODUCTO)

)

**Tablas de nivel 1**

CREATE TABLE STOCKS

(

FECHA DATETIME,

PRODUCTO INT,

cantidadINTNOTNULL,

CONSTRAINT PK\_FECHA\_PRODUCTO PRIMARYKEY (FECHA,PRODUCTO),

CONSTRAINT FK\_PRODUCTO FOREIGNKEY (PRODUCTO) REFERENCES PRODUCTOS(PRODUCTO)

)

CREATE TABLE PROVINCIAS

(

PROVINCIA INT,

descripcion VARCHAR(100)NOTNULL,

PAIS INT,

CONSTRAINT PK\_PROVINCIA PRIMARYKEY (PROVINCIA),

CONSTRAINT FK\_PAIS FOREIGNKEY (PAIS)REFERENCES PAISES(PAIS)

)

CREATE TABLE LOCALIDADES

(

LOCALIDAD INT,

descripcion VARCHAR(100)NOTNULL,

PROVINCIA INT,

CONSTRAINT PK\_LOCALIDAD PRIMARYKEY (LOCALIDAD),

CONSTRAINT FK\_PROVINCIA\_LOCALIDADES FOREIGNKEY (PROVINCIA) REFERENCES PROVINCIAS(PROVINCIA)

)

CREATE TABLE PROVEEDORES

(

PROVEEDOR INT,

descripcion VARCHAR(100)NOTNULL,

cuit VARCHAR(45)NOTNULL,

telefono VARCHAR(45),

LOCALIDAD INT,

direccion VARCHAR(100) NOT NULL,

fecha DATETIME NOT NULL,

CONSTRAINT PK\_PROVEEDOR PRIMARYKEY (PROVEEDOR),

CONSTRAINT UNIQUE\_cuit UNIQUE (cuit),

CONSTRAINT FK\_LOCALIDAD\_PROVEEDORES FOREIGNKEY (LOCALIDAD) REFERENCES LOCALIDADES(LOCALIDAD)

)

CREATE TABLE CLIENTES

(

CLIENTE INT,

descripcion VARCHAR(100)NOT NULL,

telefono VARCHAR(45),

nro\_documento INT NOT NULL,

TIPO\_DOCUMENTO INT,

direccion VARCHAR(100)NOT NULL,

LOCALIDAD INT,

C\_FISCAL INT,

CONSTRAINT PK\_CLIENTE PRIMARYKEY (CLIENTE),

CONSTRAINT FK\_TIPO\_DOCUMENTO\_CLIENTES FOREIGNKEY (TIPO\_DOCUMENTO) REFERENCES TIPOS\_DOCUMENTOS (TIPO\_DOCUMENTO),

CONSTRAINT FK\_LOCALIDAD\_CLIENTES FOREIGNKEY (LOCALIDAD)REFERENCES LOCALIDADES(LOCALIDAD),

CONSTRAINT FK\_C\_FISCAL\_CLIENTES FOREIGNKEY (C\_FISCAL) REFERENCES CONDICION\_FISCAL(C\_FISCAL)

)

CREATETABLE COMPRAS\_CABECERA

(

fecha DATETIME NOT NULL,

PROVEEDOR INT,

TIPO VARCHAR(3),

FACTURA INT,

CONSTRAINT PK\_TIPO\_FACTURA\_CC PRIMARYKEY (TIPO,FACTURA),

CONSTRAINT CK\_TIPO\_CC CHECK (TIPO='A' OR TIPO='B' OR TIPO='C'),

CONSTRAINT FK\_PROVEEDOR\_CC FOREIGNKEY (PROVEEDOR) REFERENCES PROVEEDORES (PROVEEDOR)

)

CREATETABLE COMPRAS\_DETALLE

(

PRODUCTO INT,

TIPO VARCHAR(3),

FACTURA INT,

cantidad INT NOT NULL,

precio FLOAT NOT NULL,

CONSTRAINT PK\_PTF\_CD PRIMARYKEY (PRODUCTO, TIPO, FACTURA),

CONSTRAINT FK\_PRODUCTO\_CD FOREIGNKEY (PRODUCTO) REFERENCES PRODUCTOS (PRODUCTO),

CONSTRAINT FK\_TIPO\_FACTURA\_CD FOREIGNKEY (TIPO, FACTURA) REFERENCES COMPRAS\_CABECERA (TIPO, FACTURA),

)

CREATETABLE VENTAS\_CABECERA

(

fecha DATETIME NOT NULL,

CLIENTE INT,

TIPO VARCHAR(3),

FACTURA INT,

CONSTRAINT PK\_TIPO\_FACTURA\_VC PRIMARYKEY (TIPO,FACTURA),

CONSTRAINT FK\_CLIENTE\_VC FOREIGNKEY (CLIENTE) REFERENCES CLIENTES(CLIENTE),

CONSTRAINT CK\_TIPO\_VC CHECK (TIPO='A' OR TIPO='B' OR TIPO='C')

)

CREATETABLE VENTAS\_DETALLE

(

PRODUCTO INT,

TIPO VARCHAR(3),

FACTURA INT,

cantidad INT NOT NULL,

precio FLOAT NOT NULL,

CONSTRAINT PK\_PRODUCTO\_TIPO\_FACTURA\_VD PRIMARYKEY (PRODUCTO, TIPO, FACTURA),

CONSTRAINT FK\_PRODUCTO\_VD FOREIGNKEY (PRODUCTO) REFERENCES PRODUCTOS (PRODUCTO),

CONSTRAINT FK\_TIPO\_FACTURA\_VD FOREIGNKEY (TIPO, FACTURA) REFERENCES VENTAS\_CABECERA (TIPO, FACTURA)

)

1. Definir **Trigger**. Estudiar y documentar las tablas que podrían utilizar este objeto, detallando la acción a seguir de acuerdo al evento.

Los triggers o disparadores son objetos de la base de datos que ejecutan **acciones** cuando se producen ciertos **eventos como**inserciones, modificaciones, borrados, creación de tablas, etc.)

**Trigger** para la tabla **VENTAS\_DETALLE**, para la acción INSERT, cuando se inserta un registro en VENTAS\_DETALLE debe hacer un UPDATE en la tabla STOCK, modificando la cantidad del PRODUCTO insertado en VENTAS\_DETALLE. (Cantidad insertada en VENTAS\_DETALLE debe ser en STOCKS: Cantidad actual – cantidad insertada en VENTAS\_DETALLE).

**Trigger** para la tabla **COMPRAS\_DETALLE**, para la acción INSERT, cuando se inserta un registro debe hacer un UPDATE en la tabla STOCK, modificando la cantidad del PRODUCTO insertado en COMPRAS\_DETALLE. (Cantidad insertada en COMPRAS\_DETALLE debe ser en STOCKS: Cantidad actual + cantidad insertada en COMPRAS\_DETALLE).