



| Apellido y Nombre del Estudiante: | Cantidad de hojas a entregar: | NOTA |
|-----------------------------------|-------------------------------|------|
|                                   |                               |      |

**Tecnicatura Superior en Programación  
Diseño y Administración de Base de Datos**

**Primer Parcial**

- Resuelva los ejercicios en hoja separada y con lapicera.
- Como condición de aprobación se debe tener completado el 50% del ejercicio cada módulo para aprobar.
- Si la letra es ilegible se descontarán puntos.
- Ante cualquier duda en la interpretación consulte al profesor o escriba su interpretación para que sea tomada en cuenta.
- Escriba su nombre en TODAS LAS HOJAS y enumérelas. Cuando tenga el total de páginas anótelos en la parte superior de esta hoja.

**A. NORMALIZACION Y DER**

- 1) **Generar el diagrama de Entidad-Relación que nos permita cumplir con el siguiente enunciado:**

Aerolíneas UTN quiere comenzar a vender sus pasajes a través de internet. Aerolíneas UTN opera en varios mercados: ARGENTINA, BRASIL, COLOMBIA, PERÚ y ESPAÑA. Cada uno de estos mercados puede vender cualquier ruta que opere aerolíneas UTN pero las tarifas varían de mercado en mercado. Es decir, la ruta Buenos Aires - Córdoba tiene una tarifa para el mercado ARGENTINA y otro para el mercado BRASIL. Las tarifas se expresan en dólares. Los aviones de aerolíneas UTN, tienen cabina única, por lo tanto, no hay diferencias de tarifas entre Primera o Turista. La misma tarifa se aplica a cada pasajero.

Cada una de estas rutas tiene un detalle de vuelos, es decir cada ruta tendrá X vuelos que la cubrirán. Cada vuelo posee como datos la ruta que cubre, la fecha y hora del vuelo, el número de vuelo y el Avión que lo realizará, ya que debemos tener una lista de los aviones de la flota disponibles. Cada Avión tiene como datos su modelo, año y horas de vuelo. Los aviones de aerolíneas UTN son de color blanco. Cada uno de estos vuelos debe tener una tripulación asignada, por lo tanto debemos tener los datos de los tripulantes de cada vuelo (Nombre, Apellido, Pasaporte y Puesto). Los puestos pueden ser comandante, Piloto, Copiloto o Azafata.

El pasajero puede realizar una reserva para él y X pasajeros desde la página principal de cada uno de los mercados. Cada reserva genera un PNR que es número unívoco de reserva generado por un sistema externo. Se debe tener el detalle de cada pasajero (Nombre, Apellido, Pasaporte, Nacionalidad y Edad). La reserva tiene originalmente el estado de RESERVADA, debe pagarse en 48hs o se elimina del sistema. Esta puede pasar al estado CONFIRMADA una vez pagada. El pago puede hacerse por tarjeta de crédito, débito o efectivo. Como no se tienen estadísticas de cómo se pagan las reservas sería conveniente poseer esta información con el fin de generar promociones.

**Tener en cuenta:**

Especificar relaciones y cardinalidad, Identificar PK y FK e Identificar UNQ Constraints.



## B. SQL e INTEGRIDAD REFERENCIAL.

Dada la siguiente definición de base de datos :

```
CREATE TABLE RUBROS (ID_RUBRO INT AUTO_INCREMENT,  
                     NOMBRE VARCHAR(50),  
                     CONSTRAINT PK_RUBROS PRIMARY KEY (ID_RUBRO));
```

```
CREATE TABLE DEPOSITOS (ID_DEPOSITO INT AUTO_INCREMENT,  
                          NOMBRE VARCHAR(50) NOT NULL,  
                          CONSTRAINT PK_DEPOSITO PRIMARY KEY (ID_DEPOSITO));
```

```
CREATE TABLE PRODUCTOS ( ID_PRODUCTO int AUTO_INCREMENT  
                          NOMBRE varchar(50),  
                          PRECIO_UNITARIO float NOT NULL,  
                          ID_RUBRO int,  
                          CONSTRAINT PK_PRODUCTOS PRIMARY KEY (ID_PRODUCTO),  
                          CONSTRAINT FK_ID_RUBRO_RUBROS FOREIGN KEY (ID_RUBRO) REFERENCES  
                          RUBROS(ID_RUBRO) ON DELETE NO ACTION);
```

```
CREATE TABLE STOCK_PRODUCTOS (ID_PRODUCTO INT,  
                                ID_DEPOSITO INT,  
                                STOCK INT,  
                                STOCK_MINIMO INT,  
                                CONSTRAINT PK_STOCK_PRODUTO PRIMARY KEY (ID_PRODUCTO, ID_DEPOSITO),  
                                CONSTRAINT FK_ID_PRODUCTO_PRODUCTOS FOREIGN KEY (ID_PRODUCTO)  
                                REFERENCES PRODUCTOS (ID_PRODUCTO) ON DELETE  
                                CASCADE,  
                                CONSTRAINT FK_ID_DEPOSITO_DEPOSITOS FOREIGN KEY (ID_PRODUCTO)  
                                REFERENCES  
                                DEPOSITOS(ID_DEPOSITO) ON DELETE CASCADE,  
                                CONSTRAINT CHK_STOCK CHECK (STOCK > 0));
```

```
CREATE TABLE VENTAS (ID_VENTA INT AUTO_INCREMENT,  
                      MES INT ,  
                      ANIO INT,  
                      DIA INT,  
                      RAZON_SOCIAL VARCHAR(50),  
                      CONSTRAINT PK_VENTAS PRIMARY KEY (ID_VENTA));
```

```
CREATE TABLE ITEMS_VENTAS (ID_ITEM_VENTA INT AUTO_INCREMENT,  
                             ID_PRODUCTO INT,  
                             CANTIDAD INT NOT NULL,  
                             PRECIO_UNITARIO FLOAT NOT NULL,  
                             ID_VENTA INT,  
                             CONSTRAINT PK_ITEM_VENTA PRIMARY KEY (ID_ITEM_VENTA),  
                             CONSTRAINT FK_ID_PRODUCTO_PRODUCTOS_ITEM_VENTAS FOREIGN KEY  
                             (ID_PRODUCTO) REFERENCES PRODUCTOS(ID_PRODUCTO),  
                             CONSTRAINT FK_ID_VENTA_VENTAS FOREIGN KEY (ID_VENTA) REFERENCES  
                             VENTAS(ID_VENTA),  
                             CONSTRAINT UNQ_ID_PRODUCTO_ID_VENTA UNIQUE (ID_PRODUCTO, ID_VENTA));
```



**1) ¿Qué sucede al ejecutar las siguientes sentencias?:**

```
INSERT INTO RUBROS(NOMBRE) VALUES ('Rubro 1')
INSERT INTO PRODUCTOS(ID_RUBRO,NOMBRE,PRECIO_UNITARIO)
VALUES (1,'Producto 1',12.12);
INSERT INTO VENTAS(FECHA,RAZON_SOCIAL) values (GETDATE(),'Cliente para el parcial')
INSERT INTO ITEMS_VENTAS(ID_PRODUCTO,ID_VENTA,CANTIDAD,PRECIO_UNITARIO)
VALUES (1,1,10,1);
INSERT INTO ITEMS_VENTAS(ID_PRODUCTO,ID_VENTA,CANTIDAD,PRECIO_UNITARIO)
VALUES (1,1,33,1.5);
```

- a) Se inserta un registro correctamente en la tabla RUBRO, PRODUCTOS y VENTAS.
- b) Se inserta un registro en la tabla VENTAS.
- c) Se inserta el primer registro en la tabla ITEM\_VENTAS, pero no se inserta el segundo registro por una restricción FOREIGN KEY.
- d) Se inserta el primer registro en la tabla ITEM\_VENTAS, pero no se inserta el segundo registro por una restricción UNIQUE.
- e) Ninguna de las anteriores
- f) Son correctas las opciones a,b y d.

**2) Si se elimina un producto con la sentencia DELETE FROM PRODUCTOS WHERE ID = 5**

- a) Se elimina el registro de la tabla productos con el campo id = 5.
- b) Se elimina a) y además se eliminan todos los registros de la tabla STOCK\_PRODUCTOS.
- c) Se eliminan el registro de la tabla RUBROS al que pertenece el producto
- d) Ninguna de las anteriores
- e) Son correctas las opciones a y b.

**3) ¿Qué sucede al ejecutar la sentencia UPDATE STOCK\_X\_PRODUCTOS SET STOCK =0 WHERE ID\_DEPOSITO = 5?**

- a) Se actualiza el campo STOCK de todos los productos con ID\_DEPOSITO = 5.
- b) La tabla STOCK\_X\_PRODUCTOS no permite que el campo STOCK sea 0.
- c) Se eliminan todos los registros de la tabla DEPOSITOS con ID=5
- d) Ninguna de las anteriores.

**4) Escribir una sentencia MySQL para obtener la razón social y la cantidad de ventas que se emitieron a esa empresa en el mes de marzo.**

**5) Escribir una sentencia MySQL para mostrar el Nombre del producto, el nombre del rubro al que pertenece y el stock que tiene.**

**6) Escribir una sentencia MySQL para tomar los 5 Rubros que contienen más productos.**

**7) Escribir una sentencia MySQL para mostrar el las razones sociales cuyo nombre contiene en su nombre "S.A" y sus ventas están activas en el año corriente.**

| Ejercicio A | Item | Puntaje | Puntaje Obtenido |
|-------------|------|---------|------------------|
| A           | 1    | 45      |                  |
| B           | 1    | 5       |                  |
|             | 2    | 5       |                  |
|             | 3    | 5       |                  |
|             | 4    | 10      |                  |
|             | 5    | 10      |                  |
|             | 6    | 10      |                  |
|             | 7    | 10      |                  |
| TOTAL       |      | 100     |                  |



**UNIVERSIDAD TECNOLÓGICA NACIONAL**  
**FACULTAD REGIONAL MAR DEL PLATA**

Diseño y Administración de Base de Datos I

2do Año – 2do Cuatrimestre