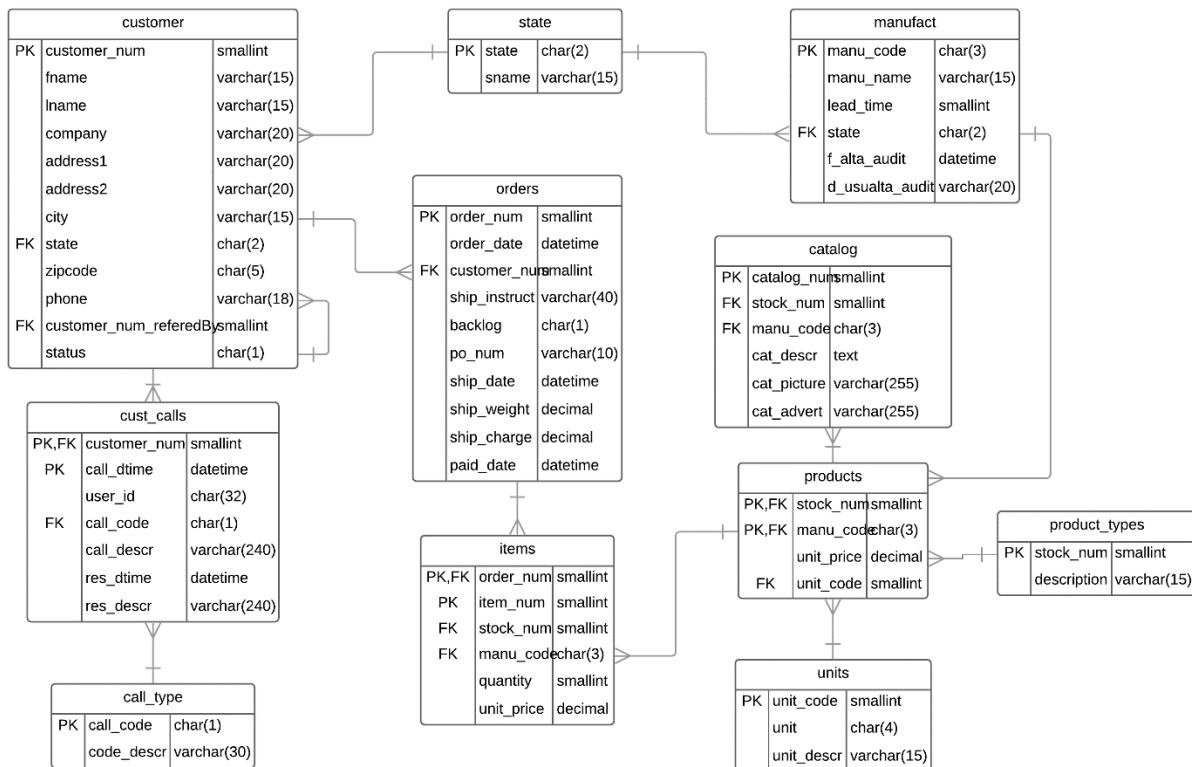


Legajo: ..... Apellido y Nombre:.....



1. Explique las diferencias existentes entre una función de usuario y un stored procedure.

2. Detalle por lo menos 3 objetos de bases de datos relacionados con la funcionalidad de integridad. Explique brevemente el uso de cada objeto para asegurar la integridad.

### 3. Query

Crear una consulta que muestre de las tres Estados que tengan la mayor cantidad de **VENTAS (no compras)**: Nombre del Estado, monto total vendido en ese Estado, nombre del fabricante y cantidad vendida total de ese fabricante en esa provincia.

Solo se deberán mostrar en la consulta los fabricantes cuyas ventas totales superen el 15% de las ventas de su provincia.

Ordenar el resultado por el monto total vendido del Estado de mayor a menor y por monto vendido del fabricante de manera descendente.

**Notas:** Se puede utilizar SOLO UN subquery. No usar Store procedures, ni funciones de usuarios, ni tablas temporales.

Ej.

Estado	MontoTotal	Código	Nombre	MontoFabricante
California	23878.80	ANZ	Anza	11081.80
California	23878.80	HSK	Husky	5998.00
California	23878.80	KAR	Karsten	3917.00
Colorado	9586.91	SHM	Shimara	6077.91
Colorado	9586.91	SMT	Smith	3509.00
Arizona	6380.00	NRG	Norge	3500.00
Arizona	6380.00	PRC	ProCycle	2130.00

#### 4. Store Procedure

Crear un procedimiento **ResumenMensualPR** que reciba una fecha como parámetro. Este Procedure deberá guardar en una tabla **VENTASxMES** el Monto total y las cantidades totales de unidades vendidas de **productos** para el Año y mes (yyyymm) de la fecha ingresada como parámetro.

Dependiendo del atributo **unit** correspondiente a la unidad del producto las **cantidades** deberán ser “ajustadas” según la siguiente tabla:

Box: Se multiplica la cantidad x 12  
 Case: Se multiplica la cantidad x 6  
 Pair: Se multiplica la cantidad x 2  
 Each: Las cantidades no se ajustan.

Tabla VENTASxMES

anioMes      varchar(6) PK  
 stock\_num   smallint      PK  
 manu\_code   char(3)      PK  
 Cantidad      int  
 Monto          decimal(10,2)

El procedimiento debe manejar TODO el proceso en una transacción y deshacer todas las operaciones en caso de error.

#### 5. Trigger

Se cuenta con una tabla PermisosxProducto que contiene por cada **customer\_num** los **productos** que este cliente puede comprar.

La estructura de la tabla es la siguiente:

(Customer\_num, Manu\_code, Stock\_num)

Se pide crear un trigger que ante la inserción de una o varias filas en la tabla **ítems**, valide que el customer\_num de la orden a la que pertenece cada ítem tenga permiso de compra sobre el producto asociado a dicho ítem (manu\_code+stock\_num).

Legajo: ..... Apellido y Nombre:.....

En caso que el cliente (customer\_num) no tenga permisos (no exista un registro en la tabla **permisosPorProducto**) se deberá cancelar la inserción enviando un mensaje de error y deshacer todas las operaciones realizadas

Nota: Las inserciones pueden ser masivas.

Notas

1	2	3	4	5
10	10	30	30	20