

## EJERCICIOS SQL PRACTICA: FUNCIONES AGREGACIÓN

Almacen(**Nro**, Nombre, Responsable)

Articulo(**CodArt**, Descripcion, precio)

Material(**CodMat**, Descripcion)

Proveedor(**CodProv**, Nombre, Domicilio, CodCiudad)

Ciudad(**CodCiudad**, Nombre)

Contiene(**Cod\_Contiene**, Nro, CodArt)

Compuesto\_por(**Cod\_Composicion**, CodArt, CodMat)

Provisto\_por(**Cod\_Provisto**, CodMat, CodProv)

Referencias: Negrita y con color azul -> PK

Subrayado con guiones-> FK

Generar las sentencias necesarias para modelar la base de datos con información.

### # Creo las tablas

```
CREATE TABLE ALMACEN(  
  Nro INT(5) PRIMARY KEY,  
  Nombre VARCHAR(20) NOT NULL,  
  Responsable VARCHAR(50) NOT NULL);
```

```
CREATE TABLE ARTICULO(  
  CodArt INT(5) PRIMARY KEY,  
  Descripcion VARCHAR (50) NOT NULL,  
  Precio DOUBLE);
```

```
CREATE TABLE MATERIAL(  
  CodMat INT(6) PRIMARY KEY,  
  Descripcion VARCHAR(100) NOT NULL);
```

```
CREATE TABLE CIUDAD(  
CodCiudad INT(2) PRIMARY KEY,  
Nombre VARCHAR(100) NOT NULL);
```

```
CREATE TABLE PROVEEDOR(  
CodProv INT(5) PRIMARY KEY,  
Nombre VARCHAR(20) NOT NULL,  
Domicilio VARCHAR(100),  
CodCiudad INT(2) NOT NULL,  
FOREIGN KEY (CodCiudad) REFERENCES Ciudad (CodCiudad));
```

```
CREATE TABLE CONTIENE(  
Cod_Contiene INT(3) PRIMARY KEY AUTO_INCREMENT,  
Nro INT(5) NOT NULL,  
CodArt INT(5) NOT NULL,  
FOREIGN KEY (Nro) REFERENCES ALMACEN (Nro),  
FOREIGN KEY (CodArt) REFERENCES ARTICULO (CodArt));
```

```
CREATE TABLE COMPUESTO_POR(  
Cod_Composicion INT(3) PRIMARY KEY AUTO_INCREMENT,  
CodArt INT(5) NOT NULL,  
CodMat INT(6) NOT NULL,  
FOREIGN KEY (CodArt) REFERENCES ARTICULO (CodArt),  
FOREIGN KEY (CodMat) REFERENCES MATERIAL (CodMat));
```

```
CREATE TABLE PROVISTO_POR(  
Cod_Provisto INT(3) PRIMARY KEY AUTO_INCREMENT,  
CodMat INT(6) NOT NULL,  
CodProv INT(5) NOT NULL,  
FOREIGN KEY (CodMat) REFERENCES MATERIAL (CodMat),  
FOREIGN KEY (CodProv) REFERENCES PROVEEDOR (CodProv));
```

## # Inserto Datos

```
INSERT INTO ALMACEN (Nro, Nombre, Responsable)
VALUES (001, 'La Original', 'Alfredo'),
(002, 'Galpon', 'Esteban'),
(003, 'Almacen de Don Juan', 'Juan'),
(004, 'La Tiendita', 'Roberto');
```

```
INSERT INTO ARTICULO (CodArt, Descripcion, Precio)
VALUES (001, 'Pan', 130.70),
(002, 'Facturas', 300.00),
(003, 'Cheese Cake', 450.87),
(004, 'Pasta Frola', 278.90);
```

```
INSERT INTO MATERIAL (CodMat, Descripcion)
VALUES (001, 'Aceite'),
(002, 'Harina'),
(003, 'Levadura'),
(004, 'Huevo'),
(005, 'Azucar'),
(006, 'Sal'),
(007, 'Agua');
```

```
INSERT INTO CIUDAD(CodCiudad, Nombre)
VALUES (1, 'La Plata'),
(2, 'Capital Federal'),
(3, 'Ramos Mejia'),
(4, 'La Matanza');
```

```
INSERT INTO PROVEEDOR (CodProv, Nombre, Domicilio,
CodCiudad)
```

```
VALUES(1, 'Arcor', 'Ayacucho 1234', 1),  
(2, 'Molinos', 'Yatay 456', 4),  
(3, 'Ledesma', 'Mario Bravo 987', 1),  
(4, 'Marolio', 'Potosi 098', 2),  
(5, 'Glaciar', 'Sarmiento 555', 3),  
(6, 'Johnson', 'Potosi 123', 1);
```

```
INSERT INTO CONTIENE (Nro, CodArt)  
VALUES (001, 001),  
(001, 002),  
(001, 003),  
(001, 004),  
(002, 003),  
(002, 004),  
(003, 001),  
(004, 002);
```

```
INSERT INTO COMPUESTO_POR (CodArt, CodMat)  
VALUES(001, 001),  
(001, 002),  
(001, 003),  
(002, 002),  
(002, 005),  
(002, 007),  
(003, 001),  
(003, 002),  
(003, 006),
```

(004, 007);

```
INSERT INTO PROVISTO_POR(CodMat, CodProv)
VALUES (001, 1),
(002, 3),
(003, 5),
(004, 4),
(005, 2),
(006, 2),
(007, 5);
```

## CONSULTAS

#1) Indicar la cantidad de proveedores que comienzan con la letra L

```
SELECT COUNT(*) AS CANT_PROV
FROM PROVEEDOR
WHERE NOMBRE LIKE 'L%';
```

#2) Listar el promedio de precios de los artículos por cada almacén (nombre)

```
SELECT AL.Nombre as ALMACEN_NOMBRE,
ROUND(AVG(A.PRECIO), 2) PROM_ART
FROM ALMACEN AL JOIN CONTIENE C ON AL.Nro = C.Nro
JOIN ARTICULO A ON C.CodArt = A.CodArt
GROUP BY AL.Nro;
```

#3) Listar la descripción de artículos compuestos por al menos 2 materiales

AYUDITA: Miro por cuantos materiales está compuesto un artículo:

```
SELECT A.CodArt, COUNT(CodMat) Cant_mat_art
FROM ARTICULO A JOIN COMPUESTO_POR CP ON A.CodArt =
CP.CodArt
GROUP BY A.CodArt;
```

Finalmente:

```
SELECT A.Descripcion
FROM ARTICULO A JOIN COMPUESTO_POR CP ON A.CodArt =
CP.CodArt
GROUP BY A.CodArt
HAVING count(CodMat) >= 2;
```

#4) Listar cantidad de materiales que provee cada proveedor y el código, nombre y domicilio del proveedor.

```
SELECT P.CodProv, P.Nombre Nombre_Prov, P.Domicilio,
count(CodMat) Cant_materiales
FROM PROVEEDOR P LEFT JOIN PROVISTO_POR PP ON
P.CodProv= PP.CodProv GROUP
BY P.CodProv
ORDER BY P.CodProv;
```

#5)Cuál es el precio máximo de los artículos que estan compuestos por materiales que proveen los proveedores de la ciudad de La Plata.

```
SELECT MAX(A.Precio) Precio_Max
FROM PROVEEDOR P JOIN CIUDAD C ON P.CodCiudad =
C.CodCiudad
JOIN PROVISTO_POR PP ON P.CodProv = PP.CodProv
JOIN COMPUESTO_POR CP ON CP.CodMat = PP.CodMat
JOIN ARTICULO A ON A.CodArt = CP.CodArt
WHERE C.Nombre = 'La Plata';
```

#6) Listar los nombres de aquellos proveedores que no proveen ningún material

```
SELECT P.NOMBRE NOMBRE_PROV  
FROM PROVEEDOR P LEFT JOIN PROVISTO_POR PP ON  
P.CodProv= PP.CodProv;  
WHERE PP.CodMat IS NULL;
```

Otra forma de hacerlo:

```
SELECT P.NOMBRE NOMBRE_PROV  
FROM PROVEEDOR P LEFT JOIN PROVISTO_POR PP ON  
P.CodProv= PP.CodProv;  
GROUP BY P.CodProv  
HAVING COUNT(PP.CodMat) = 0;
```