

**Licenciatura**

Licenciatura en Ciencias Computacionales

**Semestre y grupo**

6°2

**Nombre de la materia**

Bases de Datos Distribuidas

**Nombre del alumno**

Eduardo Vazquez Bonilla

**Nombre de la tarea**

3.1 Práctica. Consultas a la BD de Distribuidora

**Nombre del profesor**

Eduardo Cornejo Velazquez

**Fecha**

31 de octubre de 2024

# Marco teórico

A continuación, se explican los conceptos relacionados con las bases de datos, fundamentales para la construcción y diseño de sistemas de información relacional.

---

## 1. Grado de una Relación

El grado de una relación hace referencia al número de entidades o tablas involucradas en una relación. Por ejemplo:

- Una relación de **grado 2** es una relación binaria, como la relación entre **purchase** y **supplier**.
  - Una relación de **grado 3** podría involucrar tres tablas, como la relación entre **buyTool**, **purchase** e **inventory**.
- 

## 2. Clave Primaria

Es un atributo o conjunto de atributos que identifica de manera única cada tupla (registro) dentro de una tabla.

Ejemplos:

- En la tabla **supplier**, la clave primaria es **idSupplier**.
  - En la tabla **purchase**, la clave primaria es **idPurchase**.
- 

## 3. Clave Foránea

Es un atributo en una tabla que se refiere a la clave primaria de otra tabla, estableciendo relaciones entre ambas.

Ejemplo:

- En la tabla **purchase**, **idSupplier** es una clave foránea que se refiere a **idSupplier** en la tabla **supplier**.
-

## 4. Clave Candidata

Son los atributos que podrían funcionar como clave primaria porque identifican de manera única las tuplas de una tabla.

Ejemplo:

- En la tabla **warehouse**, el atributo **idWarehouse** es la clave primaria, pero también podrían usarse combinaciones como (**street**, **number**) si garantizan unicidad.
- 

## 5. Superclave

Es un conjunto de atributos que puede identificar de manera única las tuplas, pero puede contener atributos redundantes. Todas las claves primarias son superclaves, pero no todas las superclaves son claves primarias.

Ejemplo:

- En la tabla **tool**, {**idTool**, **name**} es una superclave.
- 

## 6. Cardinalidad

Define el número de elementos de una entidad que se relacionan con los elementos de otra.

Ejemplo:

- Entre **supplier** y **purchase**, la cardinalidad es **uno a muchos** (un proveedor puede tener múltiples compras).
- 

## 7. Fragmentación Horizontal

Consiste en dividir una tabla en subconjuntos de filas basados en una condición lógica, manteniendo todos los atributos originales.

Ejemplo:

- Fragmentar la tabla **inventory** según la ubicación en un almacén específico (**idWarehouse = 1**).

**Expresión en álgebra relacional:**

$\sigma_{idWarehouse=1}(inventory)$

**SQL para crear el fragmento:**

sql

Copy code

```
CREATE TABLE inventory_warehouse_1 AS  
SELECT * FROM inventory WHERE idWarehouse = 1;
```

---

## 8. Fragmentación Vertical

Consiste en dividir una tabla en subconjuntos de columnas, manteniendo las claves primarias para preservar la unicidad.

Ejemplo:

- Fragmentar la tabla `tool` en dos, separando atributos descriptivos de atributos identificadores.

**Expresión en álgebra relacional:**

$$\pi_{idTool, name, model}(tool)$$

**SQL para crear el fragmento:**

sql

Copy code

```
CREATE TABLE tool_identifiers AS  
SELECT idTool, name, model FROM tool;
```

---

## 9. Índices

Los índices son estructuras que optimizan el acceso a los datos, acelerando las consultas y mejorando el rendimiento de la base de datos.

Ejemplo:

- Crear un índice sobre el atributo `name` de la tabla `tool` para búsquedas más rápidas.

**SQL para crear un índice:**

sql

Copy code

```
CREATE INDEX idx_tool_name ON tool(name);
```

---

## 10. Vistas

Las vistas son consultas almacenadas que representan datos en una estructura lógica simplificada. Ejemplo:

- Crear una vista para mostrar las herramientas disponibles en un almacén específico.

**SQL para crear una vista:**

```
sql
Copy code
CREATE VIEW available_tools AS
SELECT t.name, t.model, i.amount, w.city
FROM tool t
JOIN inventory i ON t.idTool = i.idTool
JOIN warehouse w ON i.idWarehouse = w.idWarehouse
WHERE i.amount > 0;
```

---

## Desarrollo: Fragmentación en el Caso Propuesto

```
CREATE DATABASE IF NOT EXISTS SupplyChain;
USE SupplyChain;
```

```
CREATE TABLE supplier (
    idSupplier INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    street VARCHAR(100),
    number INT,
    city VARCHAR(50),
    state VARCHAR(50),
    phone VARCHAR(15),
    email VARCHAR(100),
    contact VARCHAR(100)
);
```

```
CREATE TABLE tool (
    idTool INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    model VARCHAR(100),
    manufacturer VARCHAR(100),
    color VARCHAR(50),
```

```

        size VARCHAR(50),
        description TEXT
    );

CREATE TABLE warehouse (
    idWarehouse INT AUTO_INCREMENT PRIMARY KEY,
    street VARCHAR(100),
    number INT,
    city VARCHAR(50),
    state VARCHAR(50),
    area FLOAT,
    phone VARCHAR(15),
    manager VARCHAR(100)
);

CREATE TABLE inventory (
    idInventory INT AUTO_INCREMENT PRIMARY KEY,
    idWarehouse INT NOT NULL,
    idTool INT NOT NULL,
    zone VARCHAR(50),
    rack VARCHAR(50),
    level INT,
    location VARCHAR(100),
    amount INT,
    purchasePrice DECIMAL(10, 2),
    storeSalePrice DECIMAL(10, 2),
    wholeSalePrice DECIMAL(10, 2),
    FOREIGN KEY (idWarehouse) REFERENCES warehouse(idWarehouse),
    FOREIGN KEY (idTool) REFERENCES tool(idTool)
);

CREATE TABLE purchase (
    idPurchase INT AUTO_INCREMENT PRIMARY KEY,
    idSupplier INT NOT NULL,
    date DATE,
    time TIME,
    folio VARCHAR(50),
    payment VARCHAR(50),
    FOREIGN KEY (idSupplier) REFERENCES supplier(idSupplier)
);

```

```
CREATE TABLE buyTool (  
    idBuyTool INT AUTO_INCREMENT PRIMARY KEY,  
    idPurchase INT NOT NULL,  
    idInventory INT NOT NULL,  
    amount INT,  
    unitPrice DECIMAL(10, 2),  
    note TEXT,  
    FOREIGN KEY (idPurchase) REFERENCES purchase(idPurchase),  
    FOREIGN KEY (idInventory) REFERENCES inventory(idInventory)  
);
```

---

## Insertion de datos:

-- Tabla supplier

```
INSERT INTO supplier (idSupplier, name, street, number, city, state,  
phone, email, contact) VALUES
```

```
(1, 'Proveedor Uno', 'Av. Reforma', 100, 'Ciudad de México', 'CDMX',  
'5551234567', 'contacto1@mail.com', 'Juan Pérez'),
```

```
(2, 'Proveedor Dos', 'Insurgentes Sur', 200, 'Ciudad de México', 'CDMX',  
'5557654321', 'contacto2@mail.com', 'Ana López'),
```

```
(3, 'Proveedor Tres', 'Av. Universidad', 300, 'Guadalajara', 'Jalisco',  
'3345671234', 'contacto3@mail.com', 'Carlos García'),
```

```
(4, 'Proveedor Cuatro', 'Av. Central', 400, 'Monterrey', 'Nuevo León',  
'8187651234', 'contacto4@mail.com', 'Luisa Martínez'),
```

```
(5, 'Proveedor Cinco', 'Paseo de la Reforma', 500, 'Puebla', 'Puebla',  
'2224567890', 'contacto5@mail.com', 'Pedro Sánchez');
```

-- Tabla tool

```
INSERT INTO tool (idTool, name, model, manufacturer, color, size,  
description) VALUES
```

```
(1, 'Taladro', 'TAL123', 'Makita', 'Rojo', 'Mediano', 'Taladro industrial  
para trabajos pesados.'),
```

```
(2, 'Martillo', 'MAR456', 'Stanley', 'Negro', 'Grande', 'Martillo de uso  
general con mango ergonómico.'),
```

```
(3, 'Sierra', 'SIE789', 'Bosch', 'Azul', 'Pequeño', 'Sierra circular de  
precisión.'),
```

```
(4, 'Destornillador', 'DES321', 'Truper', 'Naranja', 'Mediano',  
'Destornillador de estrella con punta magnética.'),
```

```
(5, 'Llave Inglesa', 'LLA654', 'DeWalt', 'Amarillo', 'Grande', 'Llave  
ajustable para múltiples tamaños de tornillos.');
```

```
-- Tabla warehouse
```

```
INSERT INTO warehouse (idWarehouse, street, number, city, state, area,  
phone, manager) VALUES
```

```
(1, 'Av. Hidalgo', 100, 'Ciudad de México', 'CDMX', '500 m2',  
'5559876543', 'Ricardo Sánchez'),
```

```
(2, 'Av. Juárez', 200, 'Guadalajara', 'Jalisco', '600 m2', '3345678901',  
'Laura Gómez'),
```

```
(3, 'Paseo de la Reforma', 300, 'Monterrey', 'Nuevo León', '700 m2',  
'8181234567', 'Javier Díaz'),
```

```
(4, 'Av. Central', 400, 'Puebla', 'Puebla', '400 m2', '2223456789',  
'Carlos Martínez'),
```

```
(5, 'Plaza México', 500, 'Cancún', 'Quintana Roo', '550 m2', '9987654321',  
'Fernando Gómez');
```

```
-- Tabla purchase
```

```
INSERT INTO purchase (idPurchase, idSupplier, date, time, folio, payment)  
VALUES
```

```
(1, 1, '2024-09-01', '10:00:00', 'FOL001', 'Efectivo'),
```

```
(2, 2, '2024-09-02', '11:00:00', 'FOL002', 'Crédito'),
```



```
(3, 3, '2024-09-03', '12:00:00', 'FOL003', 'Efectivo'),
(4, 4, '2024-09-04', '13:00:00', 'FOL004', 'Débito'),
(5, 5, '2024-09-05', '14:00:00', 'FOL005', 'Efectivo'),
(6, 1, '2024-09-06', '15:00:00', 'FOL006', 'Crédito'),
(7, 2, '2024-09-07', '16:00:00', 'FOL007', 'Efectivo'),
(8, 3, '2024-09-08', '17:00:00', 'FOL008', 'Débito'),
(9, 4, '2024-09-09', '18:00:00', 'FOL009', 'Crédito'),
(10, 5, '2024-09-10', '19:00:00', 'FOL010', 'Efectivo');
```

-- Tabla inventory

```
INSERT INTO inventory (idInventory, idWarehouse, idTool, zone, rack,
level, location, amount, purchasePrice, storeSalePrice, wholeSalePrice)
VALUES

(1, 1, 1, 'Zona A', 'Rack 1', 'Nivel 1', 'Ubicación 1', 10, 100, 150,
120),

(2, 1, 2, 'Zona B', 'Rack 2', 'Nivel 2', 'Ubicación 2', 15, 50, 75, 60),

(3, 1, 3, 'Zona C', 'Rack 3', 'Nivel 3', 'Ubicación 3', 20, 200, 300,
240),

(4, 2, 4, 'Zona D', 'Rack 4', 'Nivel 4', 'Ubicación 4', 25, 30, 45, 36),
(5, 2, 5, 'Zona E', 'Rack 5', 'Nivel 5', 'Ubicación 5', 30, 80, 120, 96),
(6, 3, 1, 'Zona A', 'Rack 6', 'Nivel 6', 'Ubicación 6', 5, 120, 180, 144),
(7, 3, 2, 'Zona B', 'Rack 7', 'Nivel 7', 'Ubicación 7', 10, 55, 80, 64),
(8, 3, 3, 'Zona C', 'Rack 8', 'Nivel 8', 'Ubicación 8', 7, 220, 330, 264),
(9, 4, 4, 'Zona D', 'Rack 9', 'Nivel 9', 'Ubicación 9', 12, 35, 53, 42),
```

(10, 4, 5, 'Zona E', 'Rack 10', 'Nivel 10', 'Ubicación 10', 8, 85, 128, 102),

(11, 1, 1, 'Zona A', 'Rack 11', 'Nivel 11', 'Ubicación 11', 12, 110, 165, 132),

(12, 1, 2, 'Zona B', 'Rack 12', 'Nivel 12', 'Ubicación 12', 16, 60, 90, 72),

(13, 1, 3, 'Zona C', 'Rack 13', 'Nivel 13', 'Ubicación 13', 18, 210, 315, 252),

(14, 2, 4, 'Zona D', 'Rack 14', 'Nivel 14', 'Ubicación 14', 22, 40, 60, 48),

(15, 2, 5, 'Zona E', 'Rack 15', 'Nivel 15', 'Ubicación 15', 28, 75, 113, 90),

(16, 3, 1, 'Zona A', 'Rack 16', 'Nivel 16', 'Ubicación 16', 9, 130, 195, 156),

(17, 3, 2, 'Zona B', 'Rack 17', 'Nivel 17', 'Ubicación 17', 13, 65, 97, 78),

(18, 3, 3, 'Zona C', 'Rack 18', 'Nivel 18', 'Ubicación 18', 10, 230, 345, 276),

(19, 4, 4, 'Zona D', 'Rack 19', 'Nivel 19', 'Ubicación 19', 17, 37, 55, 44),

(20, 4, 5, 'Zona E', 'Rack 20', 'Nivel 20', 'Ubicación 20', 14, 90, 135, 108),

(21, 1, 1, 'Zona A', 'Rack 21', 'Nivel 21', 'Ubicación 21', 11, 105, 158, 126),

(22, 1, 2, 'Zona B', 'Rack 22', 'Nivel 22', 'Ubicación 22', 19, 55, 82, 66),

(23, 1, 3, 'Zona C', 'Rack 23', 'Nivel 23', 'Ubicación 23', 12, 215, 325, 260),

(24, 2, 4, 'Zona D', 'Rack 24', 'Nivel 24', 'Ubicación 24', 30, 33, 50, 40);

---

## Conclusión

Este diseño implementa principios de organización y optimización de bases de datos relacionales, asegurando escalabilidad y rendimiento. Con la aplicación de fragmentación, índices y vistas, se mejora la administración de datos y la eficiencia en consultas.