**Laboratorio de Bases de Datos Distribuidas**

**Práctica No.: 4**

**Grupo No.:** 2

**Integrantes:**

* Pérez Enrique

**Tema: Servidores Vinculados (PCG)**

**Objetivos:**

* Registro del servidor vinculado
* Establecimiento de seguridad
* Consulta desde [SQL Server](https://aulasvirtuales.epn.edu.ec/mod/url/view.php?id=10440504) para obtener datos de la base de datos de su escenario instalada y configurada en el servidor vinculado

**Marco teórico:**

El informe sigue un formato estructurado, con secciones bien definidas para cada aspecto mencionado. Cada sección contiene explicaciones detalladas, gráficos clarificadores y capturas de pantalla que respaldan la implementación y los procesos de extracción de datos. En síntesis, la combinación del grafo relacional y el diseño distribuido en PostgreSQL permite la administración eficiente de datos en entornos distribuidos, destacando la importancia de la documentación precisa y la visualización clara para garantizar la coherencia y eficacia del sistema distribuido.

**Desarrollo de la práctica:**

**Escenario**

Un taller de reparación de vehículos desea diseñar e implementar una base de datos distribuida para sus empleados, los datos de clientes, y la información sobre los automóviles que repara en cada uno de sus talleres. Los datos de los diferentes talleres estarán almacenados en tres localidades dependiendo de la ciudad en la que esté ubicada la misma. Las localidades de almacenamiento serán: Quito y Guayaquil.

La siguiente lista de especificaciones describe los principales requisitos de operación del taller:

• Cada taller está ubicado en una localidad, se identifica con un código único, tiene un nombre y un director que es empleado del taller.

• El personal contratado por la empresa se identifica mediante un código de empleado que mantendrán mientras trabajen en dicha empresa independientemente del taller al que estén asignados. La Administración almacena para cada empleado el número de cédula de ciudadanía, el nombre, varios números telefónicos, la fecha de comienzo de contrato, el salario y taller en el que trabaja. Cada empleado sólo puede estar asignado a un taller.

• Los vehículos que pasan por los talleres pueden asociarse a más de un cliente y un cliente puede tener más de un vehículo. Cada vehículo se identifica por un número de matrícula. La empresa mantiene para cada vehículo la fecha de compra, las fechas en las que el vehículo fue llevado a reparar, el tipo de reparación, las observaciones y el precio de la reparación. Así pues, cada reparación es única para un determinado vehículo.

• Considérese también que cuando el cliente lleva a reparar a un taller un nuevo vehículo y éste se da de alta en la Base de Datos, se vincula el automóvil a dicho taller. Esto no impide que el cliente pueda llevar después su vehículo a reparar a otros talleres, sin embargo, ya no es relevante almacenar en qué taller se llevan a cabo sucesivas reparaciones.

• Por cada cliente, se almacena número de cédula de ciudadanía, el nombre, apellido y la ciudad donde reside. En el taller de Quito se mantienen los nombres y apellidos de todos los clientes del taller, por otro lado, estos datos también son necesarios en las dos localidades Quito y Guayaquil. No se consideran clientes homónimos.

**Análisis del problema**

Se identifican 5 entidades fundamentales:

**Taller:** empresa que brinda el servicio de reparación de vehículos

**Empleado:** persona que trabaja en el taller mecánico

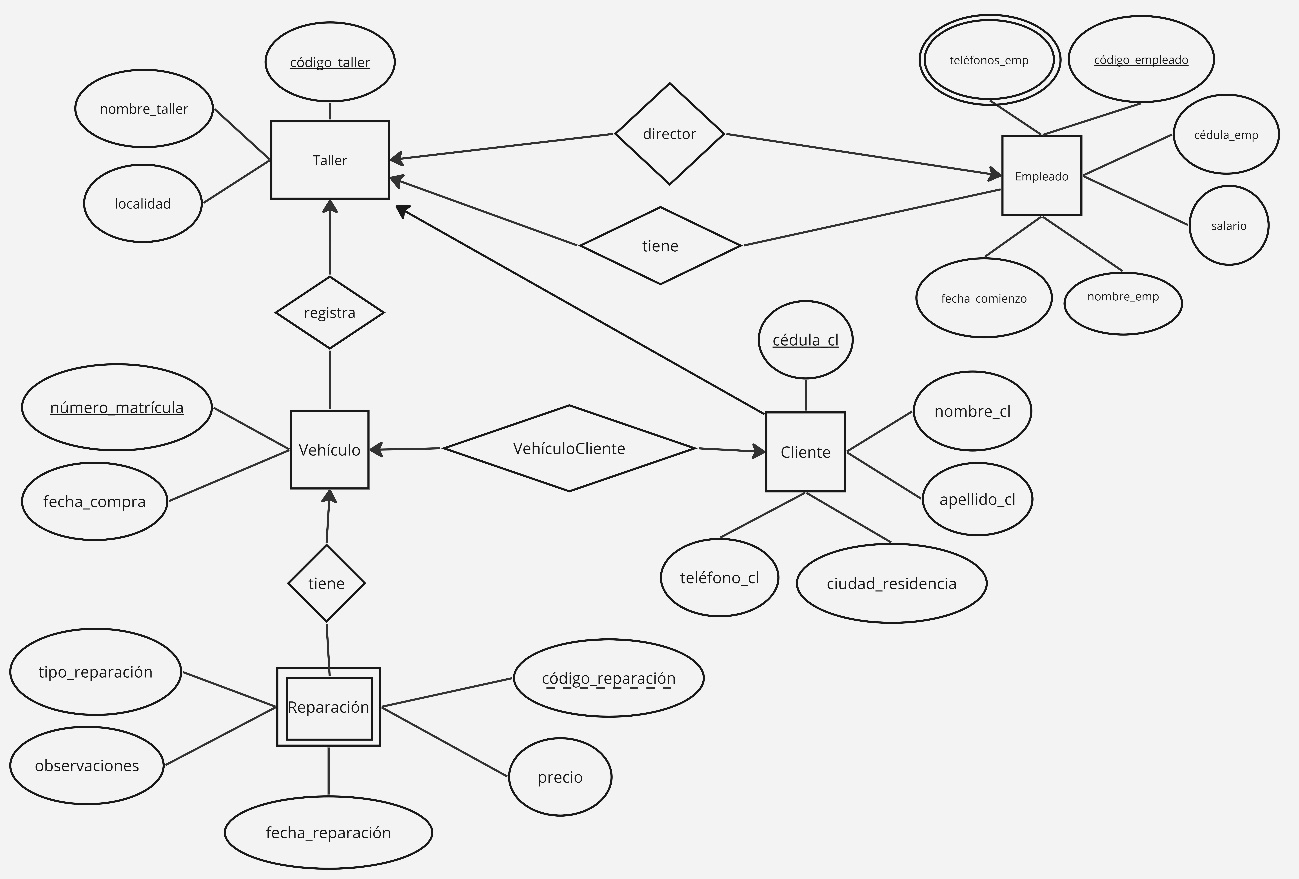
**Cliente:** necesita reparar un vehículo y es el responsable del contrato de reparación

**Vehículo:** objeto central del problema, un vehículo se registra en un taller y es llevado por un cliente que necesita repararlo.

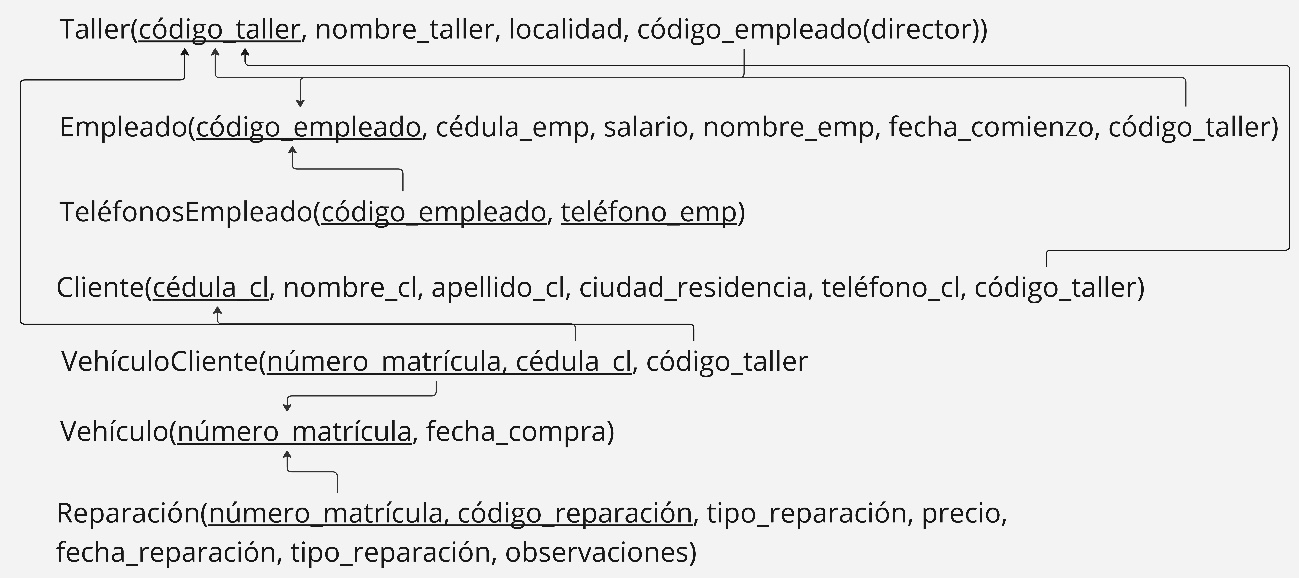
**Reparación:** se define como una entidad débil que depende de la existencia de un vehículo, aunque tiene sus propias características como es el precio, fecha, el tipo de reparación, las observaciones.

Cada reparación que tenga un vehículo va a ser diferente de cualquier otra que haya tenido o tenga en el futuro.

**Modelo entidad relación**



**Grafo Relacional**

****

**Esquema de ubicación**

|  |  |  |
| --- | --- | --- |
| SEDES | QUITO | GUAYAQUIL |
| TALLER | TALLERES\_QUITO | TALLERES\_GUAYAQUIL |
| EMPLEADO | EMPLEADOS \_QUITO | EMPLEADOS \_GUAYAQUIL |
| TELEFONOSEMPLEADO | TELEFONO\_QUITO | TELEFONO\_GUAYAQUIL |
| CLIENTE | CLIENTES \_QUITO  NOMINA\_CLIENTE | CLIENTES \_GUAYAQUIL  NOMINA\_CLIENTE |
| VEHICULOCLIENTE | VEHICULOCLIENTE\_QUITO | VEHICULOCLIENTE\_GUAYAQUIL |
| VEHICULO | VEHICULOS \_QUITO  VEHICULO\_NUMEROMATRICULA | VEHICULOS \_GUAYAQUIL  VEHICULO\_NUMEROMATRICULA |
| REPARACION | REPARACION \_QUITO | REPARACION \_GUAYAQUIL |

**Paso a paso de la implementación de la base de datos y sus tablas en cada nodo**

Datos creados desde la base de datos central del Taller Mecánico en PostgreSQL

Tabla Taller

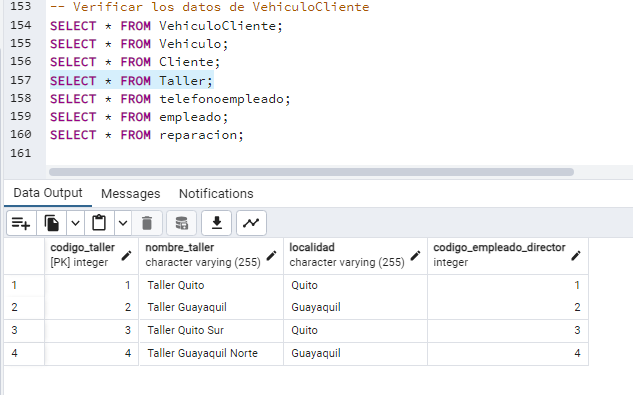


Tabla Empleado

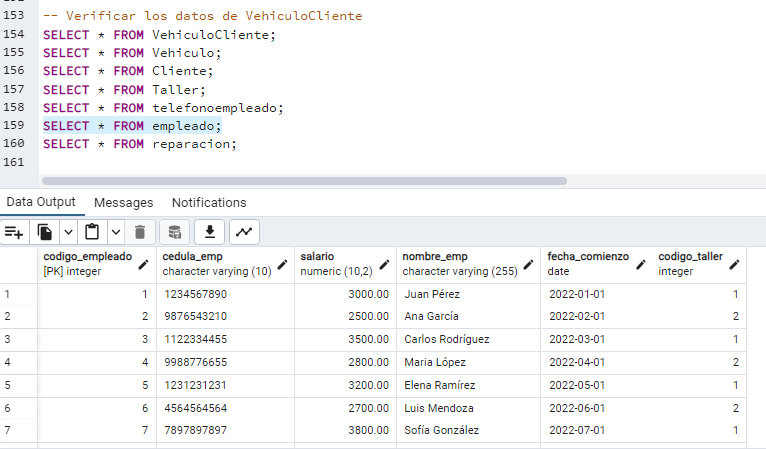


Tabla TelefonoEmpleado

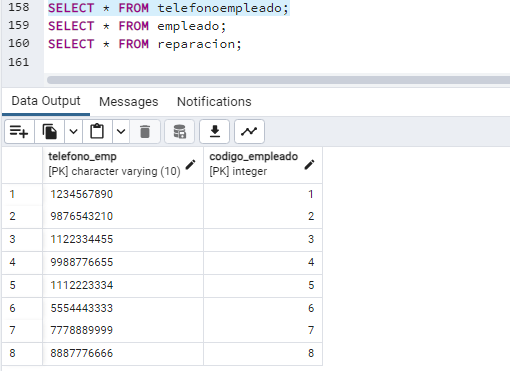


Tabla Cliente

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Tabla Vehículo

Tabla

Descripción generada automáticamente

Tabla VehiculoCliente

Tabla

Descripción generada automáticamente

Tabla Reparacion

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Extraer los datos de la base de datos centralizada**

**Taller de Guayaquil**

SELECT \* FROM TALLER\_MECANICO...cliente

CREATE DATABASE Taller\_Guayaquil

-- Taller

CREATE TABLE Taller\_Guayaquil (

codigo\_taller int ,

nombre\_taller VARCHAR(50),

localidad VARCHAR(50),

codigo\_director CHAR(10) NOT NULL

PRIMARY KEY (codigo\_taller,localidad)

);

INSERT INTO Taller\_Guayaquil (codigo\_taller,nombre\_taller,localidad,codigo\_director)

SELECT codigo\_taller,nombre\_taller,localidad,codigo\_empleado\_director

FROM TALLER\_MECANICO...Taller

WHERE localidad = 'Guayaquil'

--verificar

SELECT \* FROM Taller\_Guayaquil

SELECT \* FROM TALLER\_MECANICO...Taller

WHERE localidad = 'Guayaquil'

DROP TABLE Clientes\_Guayaquil

-- Cliente

CREATE TABLE Clientes\_Guayaquil(

cedula\_cl VARCHAR(10)NOT NULL,

nombre\_cl VARCHAR(50)NOT NULL,

apellido\_cl VARCHAR(50)NOT NULL,

ciudad\_residencia VARCHAR(50)NOT NULL,

--telefono\_cl VARCHAR(50)NOT NULL,

codigo\_taller INT NOT NULL,

localidad VARCHAR(50),

PRIMARY KEY (cedula\_cl,localidad),

FOREIGN KEY (codigo\_taller,localidad) REFERENCES Taller\_Guayaquil

(codigo\_taller,localidad)

);

INSERT INTO Clientes\_Guayaquil (cedula\_cl,nombre\_cl,apellido\_cl,ciudad\_residencia, codigo\_taller,localidad)

SELECT a.cedula\_cl,a.nombre\_cl,a.apellido\_cl,a.ciudad\_residencia,b.codigo\_taller,b.localidad

FROM TALLER\_MECANICO...cliente a

JOIN Taller\_Guayaquil b ON a.codigo\_taller=b.codigo\_taller

WHERE localidad = 'Guayaquil'

--verificar

SELECT \* FROM TALLER\_MECANICO...cliente

WHERE ciudad\_residencia = 'Guayaquil'

SELECT \* FROM Clientes\_Guayaquil

-- VehiculoCliente

CREATE TABLE VehiculoCliente\_Guayaquil (

numero\_matricula VARCHAR(10) ,

cedula\_cl VARCHAR(10)NOT NULL,

codigo\_taller INT NOT NULL,

localidad VARCHAR(50),

PRIMARY KEY (numero\_matricula,localidad),

FOREIGN KEY (cedula\_cl, localidad) REFERENCES Clientes\_Guayaquil(cedula\_cl, localidad),

FOREIGN KEY (codigo\_taller,localidad) REFERENCES Taller\_Guayaquil

(codigo\_taller,localidad)

);

INSERT INTO VehiculoCliente\_Guayaquil (numero\_matricula,cedula\_cl,codigo\_taller,localidad)

SELECT a.numero\_matricula,a.cedula\_cl,b.codigo\_taller,b.localidad

FROM TALLER\_MECANICO...VehiculoCliente a -----nnnnn

JOIN Taller\_Guayaquil b ON a.codigo\_taller=b.codigo\_taller

--verificar

SELECT \* FROM TALLER\_MECANICO...VehiculoCliente

SELECT \* FROM VehiculoCliente\_Guayaquil

-- Empleado

CREATE TABLE Empleados\_Guayaquil (

codigo\_empleado INT ,

cedula\_emp VARCHAR(15),

nombre\_emp VARCHAR(100),

fecha\_comienzo DATE,

salario DECIMAL(10, 2),

codigo\_taller INT NOT NULL,

localidad VARCHAR(50),

PRIMARY KEY(codigo\_empleado,localidad),

FOREIGN KEY (codigo\_taller,localidad) REFERENCES Taller\_Guayaquil

(codigo\_taller,localidad)

);

INSERT INTO Empleados\_Guayaquil (codigo\_empleado,cedula\_emp,nombre\_emp,fecha\_comienzo,salario,codigo\_taller,localidad)

SELECT a.codigo\_empleado,a.cedula\_emp,a.nombre\_emp,a.fecha\_comienzo,a.salario,b.codigo\_taller,b.localidad

FROM TALLER\_MECANICO...Empleado a ------ nnn

JOIN Taller\_Guayaquil b ON a.codigo\_taller=b.codigo\_taller

--verificar

SELECT \* FROM TALLER\_MECANICO...Empleado

SELECT \* FROM Empleados\_Guayaquil

-- TelefonosEmpleado

CREATE TABLE TelefonosEmpleados\_Guayaquil (

codigo\_empleado INT,

telefono\_emp VARCHAR(20),

localidad VARCHAR(50),

PRIMARY KEY (codigo\_empleado, telefono\_emp,localidad),

FOREIGN KEY (codigo\_empleado, localidad) REFERENCES Empleados\_Guayaquil(codigo\_empleado, localidad)

);

INSERT INTO TelefonosEmpleados\_Guayaquil (codigo\_empleado,telefono\_emp,localidad)

SELECT b.codigo\_empleado,a.telefono\_emp,b.localidad

FROM TALLER\_MECANICO...TelefonoEmpleado a ---------nnnn

JOIN Empleados\_Guayaquil b ON a.codigo\_empleado=b.codigo\_empleado

--verificar

SELECT \* FROM TALLER\_MECANICO...TelefonoEmpleado

SELECT \* FROM TelefonosEmpleados\_Guayaquil

-- Reparacion

CREATE TABLE Reparaciones\_Guayaquil (

numero\_matricula VARCHAR(10),

codigo\_reparacion INT,

tipo\_reparacion VARCHAR(255),

precio DECIMAL(10, 2),

fecha\_reparacion DATE,

observaciones TEXT,

localidad VARCHAR(50),

PRIMARY KEY (codigo\_reparacion,localidad),

FOREIGN KEY (numero\_matricula,localidad) REFERENCES VehiculoCliente\_Guayaquil(numero\_matricula,localidad)

);

INSERT INTO Reparaciones\_Guayaquil (numero\_matricula,codigo\_reparacion,tipo\_reparacion,precio,

fecha\_reparacion,observaciones,localidad)

SELECT a.numero\_matricula,a.codigo\_reparacion,a.tipo\_reparacion,a.precio,

a.fecha\_reparacion,a.observaciones,b.localidad

FROM TALLER\_MECANICO...Reparacion a

JOIN VehiculoCliente\_Guayaquil b ON a.numero\_matricula=b.numero\_matricula

--verificar

SELECT \* FROM Reparaciones\_Guayaquil

SELECT \* FROM TALLER\_MECANICO...Reparacion

/\*Vertical\*/

-- Vehiculo

CREATE TABLE Vehiculos (

numero\_matricula VARCHAR(20) PRIMARY KEY,

fecha\_compra DATE

);

INSERT INTO Vehiculos(numero\_matricula, fecha\_compra)

SELECT numero\_matricula, fecha\_compra

FROM TALLER\_MECANICO...Vehiculo --nnnnnnnnnnnn

--verificar

SELECT \* FROM Vehiculos

SELECT numero\_matricula FROM TALLER\_MECANICO...Vehiculo

CREATE TABLE Info\_Clientes(

cedula\_cl VARCHAR(10)NOT NULL,

nombre\_cl VARCHAR(50)NOT NULL,

apellido\_cl VARCHAR(50)NOT NULL,

PRIMARY KEY (nombre\_cl,apellido\_cl),

);

INSERT INTO Info\_Clientes(cedula\_cl, nombre\_cl, apellido\_cl)

SELECT cedula\_cl, nombre\_cl, apellido\_cl

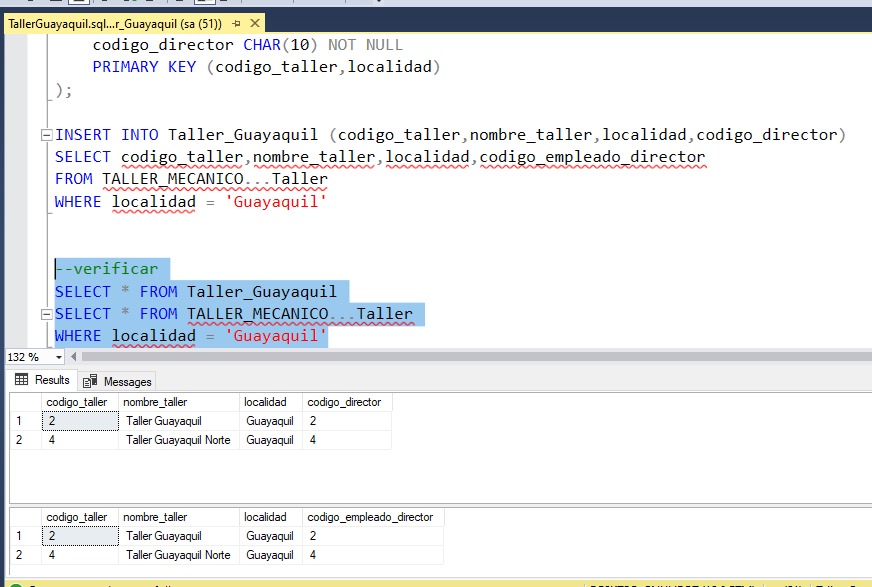
FROM TALLER\_MECANICO...cliente

SELECT \* FROM Info\_Clientes

SELECT nombre\_cl, apellido\_cl

FROM TALLER\_MECANICO...Cliente

Vamos a extraer que los datos del taller a nuestro taller para la localidad de Guayaquil

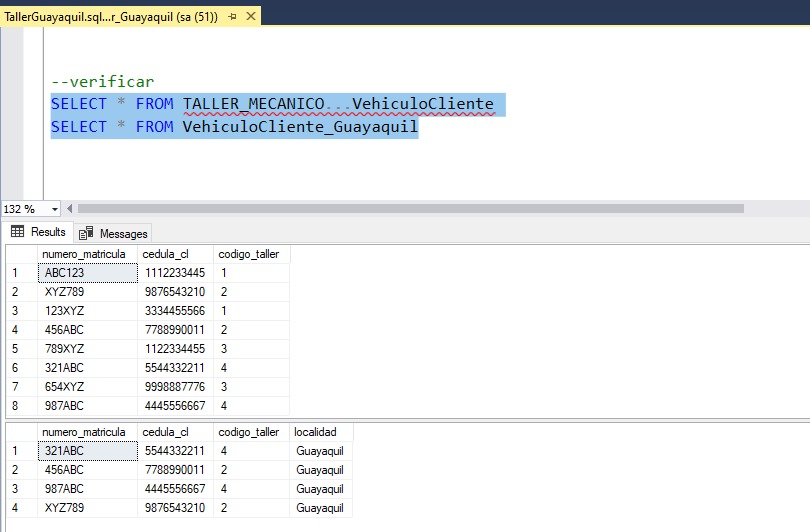


Hacemos lo mismo para los clientes

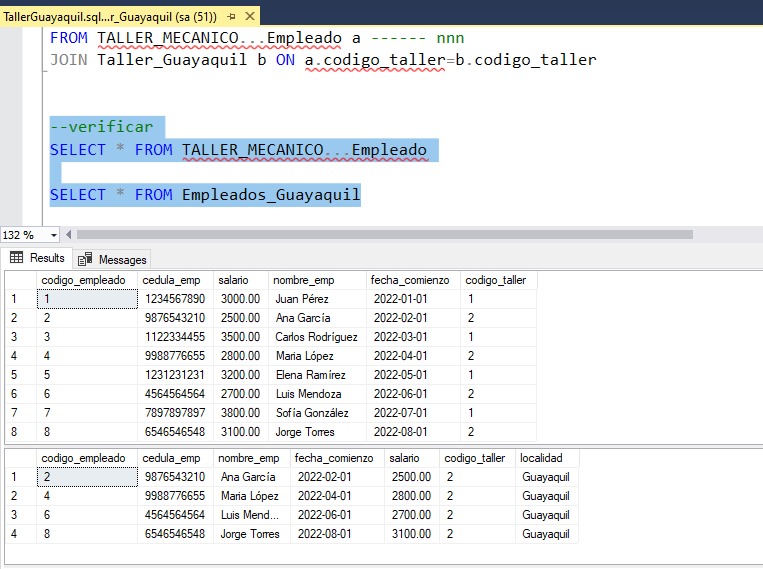
Imagen que contiene Aplicación

Descripción generada automáticamente

Después vamos a verificar los vehículos que se encuentren en el taller de Guayaquil



Lo mismo vamos a hacer para los empleados



Se prosigue con los teléfonos del empleado

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Lo mismo para reparaciones

Tabla

Descripción generada automáticamente

Para vehículos

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Después vamos a obtener la información de los clientes

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Al final tenemos las siguientes tablas para el taller de Guyaquil

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Taller de Quito**

CREATE DATABASE TALLER\_QUITO

USE TALLER\_QUITO

DROP TABLE Taller\_Quito

-- Taller

CREATE TABLE Taller\_Quito (

codigo\_taller int ,

nombre\_taller VARCHAR(50),

localidad VARCHAR(50),

codigo\_director CHAR(10) NOT NULL

PRIMARY KEY (codigo\_taller,localidad)

);

INSERT INTO Taller\_Quito (codigo\_taller,nombre\_taller,localidad,codigo\_director)

SELECT codigo\_taller,nombre\_taller,localidad,codigo\_empleado\_director

FROM TALLER\_MECANICO...Taller

WHERE localidad = 'Quito'

--verificar

SELECT \* FROM Taller\_Quito

SELECT \* FROM TALLER\_MECANICO...Taller

WHERE localidad = 'Quito'

DROP TABLE Clientes\_Quito

-- Cliente

CREATE TABLE Clientes\_Quito(

cedula\_cl VARCHAR(10)NOT NULL,

nombre\_cl VARCHAR(50)NOT NULL,

apellido\_cl VARCHAR(50)NOT NULL,

ciudad\_residencia VARCHAR(50)NOT NULL,

--telefono\_cl VARCHAR(50)NOT NULL,

codigo\_taller INT NOT NULL,

localidad VARCHAR(50),

PRIMARY KEY (cedula\_cl,localidad),

FOREIGN KEY (codigo\_taller,localidad) REFERENCES Taller\_Quito(codigo\_taller,localidad)

);

INSERT INTO Clientes\_Quito (cedula\_cl,nombre\_cl,apellido\_cl,ciudad\_residencia, codigo\_taller,localidad)

SELECT a.cedula\_cl,a.nombre\_cl,a.apellido\_cl,a.ciudad\_residencia,b.codigo\_taller,b.localidad

FROM TALLER\_MECANICO...cliente a

JOIN Taller\_Quito b ON a.codigo\_taller=b.codigo\_taller

WHERE localidad = 'Quito'

--verificar

SELECT \* FROM TALLER\_MECANICO...cliente

WHERE ciudad\_residencia = 'Quito'

SELECT \* FROM Clientes\_Quito

DROP TABLE VehiculoCliente\_Quito

-- VehiculoCliente

CREATE TABLE VehiculoCliente\_Quito (

numero\_matricula VARCHAR(10) ,

cedula\_cl VARCHAR(10)NOT NULL,

codigo\_taller INT NOT NULL,

localidad VARCHAR(50),

PRIMARY KEY (numero\_matricula,localidad),

FOREIGN KEY (cedula\_cl, localidad) REFERENCES Clientes\_Quito(cedula\_cl, localidad),

FOREIGN KEY (codigo\_taller,localidad) REFERENCES Taller\_Quito(codigo\_taller,localidad)

);

INSERT INTO VehiculoCliente\_Quito (numero\_matricula,cedula\_cl,codigo\_taller,localidad)

SELECT a.numero\_matricula,a.cedula\_cl,b.codigo\_taller,b.localidad

FROM TALLER\_MECANICO...VehiculoCliente a -----nnnnn

JOIN Taller\_Quito b ON a.codigo\_taller=b.codigo\_taller

--verificar

SELECT \* FROM TALLER\_MECANICO...VehiculoCliente

SELECT \* FROM VehiculoCliente\_Quito

-- Empleado

CREATE TABLE Empleados\_Quito (

codigo\_empleado INT ,

cedula\_emp VARCHAR(15),

nombre\_emp VARCHAR(100),

fecha\_comienzo DATE,

salario DECIMAL(10, 2),

codigo\_taller INT NOT NULL,

localidad VARCHAR(50),

PRIMARY KEY(codigo\_empleado,localidad),

FOREIGN KEY (codigo\_taller,localidad) REFERENCES Taller\_Quito(codigo\_taller,localidad)

);

INSERT INTO Empleados\_Quito (codigo\_empleado,cedula\_emp,nombre\_emp,fecha\_comienzo,salario,codigo\_taller,localidad)

SELECT a.codigo\_empleado,a.cedula\_emp,a.nombre\_emp,a.fecha\_comienzo,a.salario,b.codigo\_taller,b.localidad

FROM TALLER\_MECANICO...Empleado a ------ nnn

JOIN Taller\_Quito b ON a.codigo\_taller=b.codigo\_taller

--verificar

SELECT \* FROM TALLER\_MECANICO...Empleado

SELECT \* FROM Empleados\_Quito

-- TelefonosEmpleado

CREATE TABLE TelefonosEmpleados\_Quito (

codigo\_empleado INT,

telefono\_emp VARCHAR(20),

localidad VARCHAR(50),

PRIMARY KEY (codigo\_empleado, telefono\_emp,localidad),

FOREIGN KEY (codigo\_empleado, localidad) REFERENCES Empleados\_Quito(codigo\_empleado, localidad)

);

INSERT INTO TelefonosEmpleados\_Quito (codigo\_empleado,telefono\_emp,localidad)

SELECT b.codigo\_empleado,a.telefono\_emp,b.localidad

FROM TALLER\_MECANICO...TelefonoEmpleado a ---------nnnn

JOIN Empleados\_Quito b ON a.codigo\_empleado=b.codigo\_empleado

--verificar

SELECT \* FROM TALLER\_MECANICO...TelefonoEmpleado

SELECT \* FROM TelefonosEmpleados\_Quito

-- Reparacion

CREATE TABLE Reparaciones\_Quito (

numero\_matricula VARCHAR(10),

codigo\_reparacion INT,

tipo\_reparacion VARCHAR(255),

precio DECIMAL(10, 2),

fecha\_reparacion DATE,

observaciones TEXT,

localidad VARCHAR(50),

PRIMARY KEY (codigo\_reparacion,localidad),

FOREIGN KEY (numero\_matricula,localidad) REFERENCES VehiculoCliente\_Quito(numero\_matricula,localidad)

);

INSERT INTO Reparaciones\_Quito (numero\_matricula,codigo\_reparacion,tipo\_reparacion,precio,

fecha\_reparacion,observaciones,localidad)

SELECT a.numero\_matricula,a.codigo\_reparacion,a.tipo\_reparacion,a.precio,

a.fecha\_reparacion,a.observaciones,b.localidad

FROM TALLER\_MECANICO...Reparacion a

JOIN VehiculoCliente\_Quito b ON a.numero\_matricula=b.numero\_matricula

--verificar

SELECT \* FROM Reparaciones\_Quito

SELECT \* FROM TALLER\_MECANICO...Reparacion

/\*Vertical\*/

-- Vehiculo

CREATE TABLE Vehiculos (

numero\_matricula VARCHAR(20) PRIMARY KEY,

fecha\_compra DATE

);

INSERT INTO Vehiculos(numero\_matricula, fecha\_compra)

SELECT numero\_matricula, fecha\_compra

FROM TALLER\_MECANICO...Vehiculo --nnnnnnnnnnnn

--verificar

SELECT \* FROM Vehiculos

SELECT numero\_matricula FROM TALLER\_MECANICO...Vehiculo

CREATE TABLE Info\_Clientes(

cedula\_cl VARCHAR(10)NOT NULL,

nombre\_cl VARCHAR(50)NOT NULL,

apellido\_cl VARCHAR(50)NOT NULL,

PRIMARY KEY (nombre\_cl,apellido\_cl),

);

INSERT INTO Info\_Clientes(cedula\_cl, nombre\_cl, apellido\_cl)

SELECT cedula\_cl, nombre\_cl, apellido\_cl

FROM TALLER\_MECANICO...cliente

SELECT \* FROM Info\_Clientes

SELECT nombre\_cl, apellido\_cl

FROM TALLER\_MECANICO...Cliente

Extraemos la tabla de Taller para la el Taller de Quito

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Verificamos que existe en nuestro nodo

Tabla

Descripción generada automáticamente

Hacemos lo mismo con Clientes

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Verificamos

Tabla

Descripción generada automáticamente

Repetimos para vehículocliente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Verificamos

Tabla

Descripción generada automáticamente

Hacemos lo mismo para empleados

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Verificamos

Tabla

Descripción generada automáticamente

Hacemos lo mismo para telefonosEmpleados

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Verificamos

Tabla

Descripción generada automáticamente

Hacemos lo mismo para las reparaciones

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Verificamos

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Se realiza una fragmentación vertical para vehículos

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Verificamos

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Hacemos lo mismo para la información de los clientes

Texto

Descripción generada automáticamente

Verificamos

Tabla

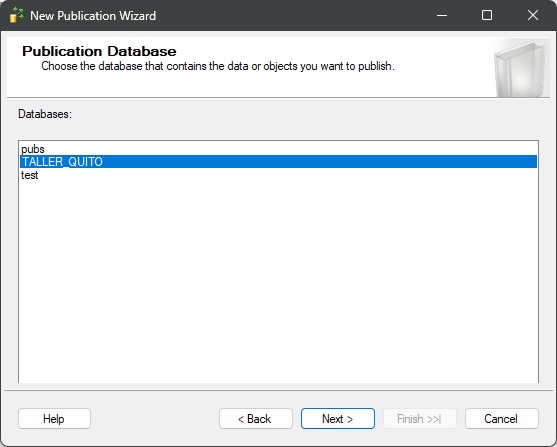
Descripción generada automáticamente

Al final obtenemos las siguientes tablas

Tabla

Descripción generada automáticamente con confianza media

**Creación del publicador**



Usamos la publicación Merge

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Seleccionamos las siguientes tablas

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

**Consultas Distribuidas**

Ejecutamos en el Nodo de Guayaquil una consulta para recuperar todos los clientes de los talleres. Además de una consulta para de todos los empleados que tiene el taller y los dos talleres existentes

Tabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Tabla

Descripción generada automáticamente con confianza media

**Conclusiones y recomendaciones:**

* En el proceso de concepción de una base de datos distribuida para un taller de reparación de vehículos, se ha abordado la creación de un esquema que refleja la descentralización de la información entre las localidades de Quito y Guayaquil, derivadas de una base de datos central en PostgreSQL.
* Analizar bien el problema es fundamental para realizar una óptima distribución de los datos en cada sede

**Contribución:**

Los tres miembros del equipo participaron activamente en el desarrollo de la actividad, de forma equitativa para alcanzar el éxito del proyecto.

# **Bibliografía**

|  |  |
| --- | --- |
| [1] | I. C. Clavijero, «Base de datos distribuidas,» 18 Enero 2023. [En línea]. Available: https://cursos.clavijero.edu.mx/cursos/080\_bdd/modulo3/contenidos/tema3.1.html?opc=0. |
| [2] | U. a. d. e. d. Hidalgo, «Apuntes de Administración de Bases de Datos,» 18 Enero 2023. [En línea]. Available: http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro21/index.html. |
| [3] | U. E. d. Milagro, «Diseño de Bases de Datos Distribuidas,» 18 Enero 2023. [En línea]. Available: https://sga.unemi.edu.ec/media/archivologo/2021/10/26/archivologocompendio\_20211026123418.pdf. |