



# **Proyecto Semestral**

## **Etapas 2**

**NOMBRES: Camila Martínez**

**Fredy Moncada**

**Alan Moreno**

**ASIGNATURA: Bases de Datos**

**PROFESOR: M. Angélica Caro G.**

**Chillán, 30 de octubre del 2018.**

## INDICE

### Contenido

INTRODUCCION .....	3
CREACION DE BD Y RESTRICCIONES .....	5
CONSULTAS .....	7
CONCLUSION .....	10

## INTRODUCCION

Tras haber concluido la primera etapa del proyecto (Creación del modelo entidad relación y Entidad relacional) nos vimos en la necesidad de modificar ciertos errores de sintaxis que nuestro supervisor nos recalcó a la hora de analizar nuestro punto de vida de la problemática planteada. La situación nos llevó a buen puerto, ya que, el MER fue modificado de tal manera que en este instante es más comprensible a la vista del administrador de la base de datos. Además consideramos las posibles consultas de datos que se podrían solicitar en un futuro no muy lejano cuando el administrador busque información relevante para su uso en la base de datos que estamos creando.

El primer cambio notorio que hicimos fue la modificación completa por parte del modelo entidad relación, la gran mayoría de entidades fuertes que existían se mantuvieron (Cliente, Arriendo, Maquinaria, Empleado, Mantención) el detalle más notorio para este caso es la generalización de los empleados, ahora lo representamos como entidad “padre” que podrá heredar sus atributos a las entidades hijas (Operador, Administrativo, Mecánico) de esta forma se puede aprovechar de mejor manera los atributos general (Dirección, Fecha de Nacimiento, Teléfono, etc.) y se evitan redundancias en el momento de las consultas próximas.

Además de lo anterior mencionado, se mantuvo la única entidad débil la cual es operador, que sigue dependiendo de la entidad Cliente. Finalmente se agregó la utilización de entidades intermedias como es el caso de Detalle Arriendo, Detalle Mantención, Tipo Maquinaria, etc.; para evitar simular relaciones de la forma  $n: n$  y poder acceder a ellas de una forma más simple.

De esta forma se procedió a la creación del Modelo Relación apoyado del nuevo Modelo Entidad Relación, de la misma forma se utilizaron los 8 pasos (entidades fuertes, débiles,  $1: n$ ,  $n: n$ , etc.) de esa forma optimiza la creación de tablas en la etapa 2 del proyecto.

Finalmente se procedió a modificar la explicación de cada uno de los atributos presentes en las entidades del MER además del dominio que estos conllevan para el almacenamiento de los datos, también se nombraron las claves primarias, foráneas (en caso de existir) y las reglas presentes en cada entidad del modelo.

**Objetivo General:** Implementar la Base de Datos (BD) diseñada en la primera etapa usando el SGBD PostgreSQL.

**Objetivos Específicos:**

1. Ajustar el modelo conceptual de la BD de acuerdo con correcciones de la primera etapa.
2. Generar los scripts en SQL para crear toda la base de datos (incluyendo la inserción de tuplas). Se deberán considerar las restricciones del dominio y el problema particular.
3. Especificar en lenguaje SQL una lista de vistas y consultas dadas

## CREACION DE BD Y RESTRICCIONES

- El tiempo mínimo de arriendo de una máquina es de 4 horas:

```
CREATE TABLE Detalle_Arriendo(  
    ID varchar(4) not null unique,  
    ID_Arriendo varchar(4) not null,  
    ID_Operador varchar(4),  
    ID_Maquinaria varchar(4) not null,  
    horas_arriendo int not null,  
    valor_total int not null,  
    CHECK (horas_arriendo >= 4),  
    PRIMARY KEY(ID),  
    FOREIGN KEY(ID_Arriendo) REFERENCES Arriendo(ID) ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY(ID_Operador) REFERENCES Operador(ID_Empleado) ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY(ID_Maquinaria) REFERENCES Maquinaria(ID) ON DELETE CASCADE ON UPDATE CASCADE);
```

- El estado de arriendo de una maquinaria debe estar limitado a un conjunto concreto como: en proceso, a tiempo, atrasado y con desperfecto:

```
INSERT INTO Estado  
VALUES ('0001','en proceso'),  
      ('0002','a tiempo'),  
      ('0003','atrasado'),  
      ('0004','con desperfecto');
```

- Todos los datos de los clientes son obligatorios:

```
CREATE TABLE Cliente(  
    ID varchar(4) not null unique,  
    ID_Tipo_Cliente varchar(4) not null,  
    ID_Direccion varchar(4) not null,  
    nombre varchar(30) not null,  
    telefono int not null,  
    rut varchar(12) not null,  
    fecha_nacimiento date,  
    cant_compras int not null,  
    PRIMARY KEY(ID),  
    FOREIGN KEY(ID_Direccion) REFERENCES Direccion_Cliente(ID) ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY(ID_Tipo_Cliente) REFERENCES Tipo_Cliente(ID) ON DELETE CASCADE ON UPDATE CASCADE);
```

- En un arriendo la dirección es un dato obligatorio:

```
CREATE TABLE Arriendo(  
    ID varchar(4) not null unique,  
    ID_Direccion varchar(4) not null,  
    ID_Cliente varchar(4) not null,  
    ID_Estado varchar(4) not null,  
    fecha date not null,  
    PRIMARY KEY (ID),  
    FOREIGN KEY(ID_Estado) REFERENCES Estado(ID) ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY(ID_Direccion) REFERENCES Direccion_Arriendo(ID) ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (ID_Cliente) REFERENCES Cliente(ID) ON DELETE CASCADE ON UPDATE CASCADE);
```

```
CREATE TABLE Direccion_Arriendo(  
    ID varchar(4) not null unique,  
    calle varchar(30) not null,  
    ciudad varchar(20) not null,  
    PRIMARY KEY(ID));
```

- Toda máquina debe tener un valor hora superior a 10.000 pesos por hora:

```
CREATE TABLE Maquinaria(
  ID varchar(4) not null unique,
  ID_Tipo_Maquinaria varchar(4) not null,
  marca varchar(12) not null,
  modelo varchar(12) not null,
  patente varchar(12) DEFAULT 'sin patente',
  kilometraje int not null,
  valor_por_hora int not null,
  CHECK(valor_por_hora > 10000),
  PRIMARY KEY(ID),
  FOREIGN KEY(ID_Tipo_Maquinaria) REFERENCES Tipo_Maquinaria(ID) ON DELETE CASCADE ON UPDATE CASCADE);
```

- Todo operador debe tener un valor hora de trabajo entre 5.000 y 8.000 pesos:

```
CREATE TABLE Operador(
  ID_Empleado varchar(4) not null unique,
  tarifa_por_hora int not null,
  CHECK(tarifa_por_hora BETWEEN 5000 AND 8000),
  FOREIGN KEY(ID_Empleado) REFERENCES Empleado(ID) ON DELETE CASCADE ON UPDATE CASCADE );
```

- Aquellas máquinas que no requieren patente deben contener en este atributo el valor 'sin patente':

```
CREATE TABLE Maquinaria(
  ID varchar(4) not null unique,
  ID_Tipo_Maquinaria varchar(4) not null,
  marca varchar(12) not null,
  modelo varchar(12) not null,
  patente varchar(12) DEFAULT 'sin patente',
  kilometraje int not null,
  valor_por_hora int not null,
  CHECK(valor_por_hora > 10000),
  PRIMARY KEY(ID),
  FOREIGN KEY(ID_Tipo_Maquinaria) REFERENCES Tipo_Maquinaria(ID) ON DELETE CASCADE ON UPDATE CASCADE);
```

- Identificar una restricción según el dominio e implementarla según corresponda:

```
CREATE TABLE Empleado(
  ID varchar(4) not null unique,
  ID_Direccion varchar(4) not null,
  ID_Contrato varchar(4) not null,
  nombre varchar(20) not null,
  rut varchar(12) not null,
  telefono int not null,
  fecha_nacimiento date,
  PRIMARY KEY(ID),
  FOREIGN KEY(ID_Direccion) REFERENCES Direccion_Empleado(ID) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (ID_Contrato) REFERENCES Contrato(ID) ON DELETE CASCADE ON UPDATE CASCADE);
```

## CONSULTAS

- Datos de todos los clientes (Rut y nombre) que contrataron algún arriendo en la ciudad Chillán y su estado sea 'A Tiempo'. Se requiere evitar repeticiones y ordenar los resultados alfabéticamente por nombre.

Data Output	Explain	Messages	Notifications	Query
	rut character varying (12)	nombre character varying (30)		
1	193333333	Marcela Alejandra		
2	194444444	Juliana Moncada		
3	196666666	Jose Moncada		

- Detalle de los arriendos (Dirección, fecha, patente y horas de arriendo) en que participó un Operador durante el mes de enero de 2017.

Data Output

[Explain](#)
[Messages](#)
[Notifications](#)
[Query History](#)

	calle character varying (30)	ciudad character varying (20)	fecha date	patente character varying (12)	horas_arriendo integer
1	Antonio Varas 50	Chillan	2017-01-20	ffffff	6

- Datos de la maquinaria (patente, tipo, marca y modelo) más arrendada en el 2018:

Data Output	Explain	Messages	Notifications	Query History
	patente character varying (12)	nombre character varying (20)	marca character varying (12)	modelo character varying (12)
1	aaaaaa	tractor	acuenta	abc

- Obtener las mantenciones (fecha, descripción y mecánico) que se le han realizado a una maquinaria determinada en un periodo dado.

Data Output	Explain	Messages	Notifications	Query History
	fecha date	descripcion character varying (100)	nombre character varying (20)	
1	2018-05-19	sin descripción	Juan Gabriel	

- Obtener un listado de las maquinarias que no han recibido mantención en los últimos 3 meses.

Data Output	Explain	Messages	Notifications	Query History			
id character varying (4)	id_tipo_maquinaria character varying (4)	marca character varying (12)	modelo character varying (12)	patente character varying (12)	kilometraje integer	valor_por_hora integer	
1	0001	acuenta	abc	aaaaaa	50000	15000	
2	0002	lider	def	sin patente	100000	20000	
3	0003	frog	ghi	cccccc	90000	25000	
4	0004	xxx	jkl	dddddd	80000	30000	
5	0005	jhon deere	mnñ	eeeeee	120000	11000	
6	0006	resident	opq	ffffff	90000	15000	
7	0007	cumnsi	rst	gggggg	100000	30000	
8	0009	yyy	xyz	iiiiii	70000	26000	
9	0010	zzz	aab	jjjjjj	130000	35000	

- Mostrar un listado con los datos de los clientes (Rut, nombre y cantidad) que han realizado arriendos en el año 2017, pero no en el año 2018.

Data Output					Explain	Messages	Notifications	Query History
	rut character varying (12)	nombre character varying (30)	cant_compras integer					
1	196666666	Jose Moncada		3				
2	197777777	Flor Cortes		1				

- Obtener un listado con todos los arriendos (fecha, nombre del cliente, dirección de arriendo y total pagado) realizados este año. Considere que el total pagado depende del valor por hora de la maquinaria y operador.

Data Output	Explain	Messages	Notifications	Query History			
fecha date	nombre character varying (30)	calle character varying (30)	ciudad character varying (20)	valor_total integer			
1	2018-09-20	Ignacio Gutierrez	Mariano Egaña 150	Chillan	312000		
2	2018-09-29	Antonio Norambuena	Los Puelches 59	Portezuelo	480000		
3	2018-10-01	Federico Rojas	Arturo Prat 960	San Carlos	180000		
4	2018-10-02	Juliana Moncada	Los Naranjos 940	Chillan	370000		
5	2018-10-02	Marcela Alejandra	El Roble 700	Chillan	900000		
6	2018-10-03	Joaquin Moncada	Circunvalacion 1090	Pinto	128000		
7	2018-10-18	Felipe Soto	Conde del Maule 269	Pinto	420000		
8	2018-10-19	Fredy Moncada	Baquedano 1282	Los Colihues	169000		



- Mostrar toda la información de todos los clientes que son preferenciales con su respectivo descuento y su cantidad de compras

	nombre character varying (30)	telefono integer	rut character varying (12)	fecha_nacimiento date	cant_compras integer	nombre character varying (30)	descuento integer
1	Antonio Norambuena	22222222	192222222	1989-02-11	1	Preferencial	5
2	Federico Rojas	11111111	191111111	1990-01-10	1	Preferencial	5
3	Felipe Soto	12345678	191234567	1990-07-29	1	Preferencial	5
4	Flor Cortes	77777777	197777777	1998-04-02	1	Preferencial	5
5	Fredy Moncada	88888888	198888888	1985-10-04	1	Preferencial	5
6	Ignacio Gutierrez	99999999	199999999	1989-11-07	2	Preferencial	10
7	Joaquin Moncada	55555555	195555555	1991-04-21	3	Preferencial	15
8	Jose Moncada	66666666	196666666	1995-06-01	3	Preferencial	15
9	Juliana Moncada	44444444	194444444	1990-02-20	2	Preferencial	10
10	Marcela Alejandra	33333333	193333333	1992-03-06	1	Preferencial	5

## CONCLUSION

PostgreSQL es un SGBD que utilizamos para la creación y organización de nuestra base de datos, en ella utilizamos comando para la incorporación de tablas, tuplas, atributos, restricciones y posteriormente las consultas pertinentes.

En primer lugar, se crearon las tablas de cada uno de las entidades propuestas en el Modelo Relacional creado en la primera etapa (que fue modificado previo a la segunda etapa), para cada una de ellas se utilizaron sus respectivas claves primarias y además se enlazaron las relaciones a través de las claves foráneas cuando correspondía.

Al momento de la creación de las tablas se tomaron algunas consideraciones, en el caso de la eliminación de tuplas (Delete) se utilizó el comando ON Delete Set Null, de esta manera en caso de que un cliente deje ser frecuente en la empresa y se retire, este tenga la oportunidad de volver a contar con nuestros servicios y sus datos se mantengan en la empresa hasta el día de su vuelta, por otro lado para la actualización de datos (Update) se utilizó el comando On Update Cascade, para que de esa forma en caso de actualizar los datos en un área los datos sean actualizados para todas sus conexiones y de esa manera ahorrar tiempo.

También se especificó que las claves primarias fueran estrictamente únicas para evitar algún problema futuro de confusión de datos entre algún cliente, arriendo, etc.; además se utiliza el comando Not Null para evitar que al momento de la inserción de los datos el administrador por algún tipo de descuido deje un casillero vacío y a la hora de consultar informaciones se entregue de manera correcta.

Posterior a la creación de las tablas y sus restricciones, se procedió a probar la base de datos insertando información simulada por nuestra cuenta para ver de qué forma se comporta la base de datos a la hora de hacer consultas para obtener información relevante del gusto del administrador.

Finalmente tras haber ejecutado las consultas de manera correcta, la base de datos está creada de manera satisfactoria y con buenos resultados a la espera de la aplicación para poder manejar de mejor manera la base de datos.