



Entrega 2: Diseño de una aplicación

Integrantes:

- Ignacio Pasten, 19639759
- Agustín González, 19638280

Fecha de entrega: 28 de mayo de 2021

Reporte:

Creación del modelo:

- **Modelo E/R**

Modelo E/R en la última página del reporte.

- **Esquema Relacional**

Para el siguiente esquema, optamos por dejar la llave primaria identificada por PRIMARY KEY en vez de subrayarla para mantener consistencia con la forma de identificar llaves foráneas.

- **Unidades:** id_unidad **int**, id_direccion **int**, jefe_id **int**, PRIMARY KEY(id_unidad)
- **CoberturaUnidades:** uid **int**, id_unidad **int**, comuna **varchar**, PRIMARY KEY(uid), FOREIGN KEY(id_unidad) REFERENCES Unidades(id_unidad)
- **Direcciones:** id_direccion **int**, nombre_direccion **varchar**, comuna **varchar**, PRIMARY KEY(id_direccion)
- **Despachos:** id_despacho **int**, fecha **date**, id_direccion_origen **int**, id_direccion_destino **int**, id_compra **int**, id_vehiculo **int**, id_repartidor **int**, PRIMARY KEY(id_despacho)
- **Personal:** id_personal **int**, nombre **varchar**, rut **varchar**, sexo **varchar**, edad **int**, PRIMARY KEY(id_personal)
- **Administrativos:** id_personal **int**, clasificacion **varchar**, id_unidad **int**, PRIMARY KEY(id_personal), FOREIGN KEY(id_personal) REFERENCES Personal(id_personal)
- **Repartidores:** vid **int**, id_personal **int**, tipo_licencia **varchar**, id_vehiculo **int**, PRIMARY KEY(vid), FOREIGN KEY(id_personal) REFERENCES Personal(id_personal)

- **Vehiculos:** id_vehiculo **int**, patente **varchar**, estado **varchar**, tipo **varchar**, id_unidad **int**, PRIMARY KEY(id_vehiculo)
- **VehiculosCarga:** id_vehiculo **int**, volumen_m3 **float**, carga_maxima_ton **float**, PRIMARY KEY(id_vehiculo), FOREIGN KEY(id_vehiculo) REFERENCES Vehiculos(id_vehiculo)
- **VehiculosFresco:** id_vehiculo **int**, alcance_km **int**, PRIMARY KEY(id_vehiculo), FOREIGN KEY(id_vehiculo) REFERENCES Vehiculos(id_vehiculo)
- **VehiculosFrio:** id_vehiculo **int**, cantidad_compartimientos **int**, capacidad_compartimientos_kg **int**, PRIMARY KEY(id_vehiculo), FOREIGN KEY(id_vehiculo) REFERENCES Vehiculos(id_vehiculo)

Como se podrá notar, en algunos atributos como “alcance_km”, dejamos la unidad del atributo expresada en “_” puesto que no logramos crear tablas con atributos que tuviesen paréntesis

Justificación del modelo:

Nuestro modelo está en 3NF, debido a que presentamos ciertas complicaciones con las dependencias, especialmente para la normalización del archivo Unidades.csv, puesto que habían repeticiones de (id, dirección, jefe), por lo que no se lograba tener una llave primaria si es que no se contaba toda la tupla.

Por esta razón, optamos por construir el modelo como fue expresado anteriormente, en el cual se tienen las siguientes dependencias:

- **Unidades:**
 - id_unidad → id_direccion, jefe_id
- **CoberturaUnidades:**
 - uid → id_unidad, comuna
- **Direcciones:**
 - id_direccion → nombre_direccion, comuna
- **Despachos:**
 - id_despacho → fecha, id_direccion_origen, id_direccion_destino, id_compra, id_vehiculo, id_repartidor
- **Personal:**
 - id_personal → nombre, rut, sexo, edad
- **Administrativos:**
 - id_personal → clasificacion, id_unidad
- **Repartidores:**
 - vid → id_personal, tipo_licencia, id_vehiculo

- Vehiculos:
 - id_vehiculo → patente, estado, tipo, id_unidad
- VehiculosCarga:
 - id_vehiculo → volumen_m3, carga_maxima_ton
- VehiculosFresco:
 - id_vehiculo → alcance_km
- VehiculosFrio:
 - id_vehiculo → cantidad_compartimientos, capacidad_compartimientos_kg

Como se puede observar, el esquema logra estar en 3NF debido a que cada dependencia cumple que el atributo del lado izquierdo de la implicancia es una super llave al ser capaz de determinar todos los atributos del lado derecho y ninguno de los otros atributos puede hacer lo mismo.

Nota: Los repartidores tienen un vid porque se repite el id_personal que tienen algunos debido al tipo_licencia. Además, considerando lo mencionado en la issue #228, en el procesamiento de datos, como cambiaba la edad de una misma persona, cambiamos los valores diferentes por la primera edad que aparecía.

Consultas SQL:

- 1)

```
SELECT U.id_unidad, D.nombre_direccion
FROM Unidades as U, Direcciones as D
WHERE U.id_direccion = D.id_direccion
ORDER BY U.id_unidad
```
- 2)

```
SELECT D.comuna, V.id_unidad, V.id_vehiculo, V.tipo, V.patente, V.estado
FROM Unidades as U, Direcciones as D, Vehiculos as V
WHERE U.id_direccion = D.id_direccion
AND V.id_unidad = U.id_unidad
AND D.comuna LIKE '%comuna%'
```
- 3)

```
SELECT DISTINCT D.comuna, V.id_unidad, V.id_vehiculo, V.tipo, V.patente,
V.estado
FROM Vehiculos as V, Despachos as De, Direcciones as D
WHERE De.id_direccion_destino = D.id_direccion
AND De.id_vehiculo = V.id_vehiculo
AND D.comuna LIKE '%comuna%'
AND date_part('year', De.fecha) = 'año'
```

- 4) SELECT De.id_despacho, De.fecha, De.id_compra, V.id_unidad, De.id_vehiculo,
V.tipo, De.id_repartidor
FROM Despachos as De, Vehiculos as V, Personal as P
WHERE De.id_vehiculo = V.id_vehiculo
AND De.id_repartidor = P.id_personal
AND V.tipo LIKE '%tipo%'
AND P.edad BETWEEN 'edad_1' AND 'edad_2'
- 5) SELECT *
FROM (SELECT U.id_unidad, P.nombre, P.rut, P.sexo
FROM Personal as P, Unidades as U, CoberturaUnidades as CU
WHERE P.id_personal = U.jefe_id
AND U.id_unidad = CU.id_unidad
AND CU.comuna LIKE '%comuna_1%'
INTERSECT
SELECT U.id_unidad, P.nombre, P.rut, P.sexo
FROM Personal as P, Unidades as U, CoberturaUnidades as CU
WHERE P.id_personal = U.jefe_id
AND U.id_unidad = CU.id_unidad
AND CU.comuna LIKE '%comuna_2%') as C
ORDER BY C.id_unidad;
- 6) SELECT *
FROM (SELECT id_unidad, COUNT(tipo) as cantidad_vehiculo
FROM Vehiculos
WHERE tipo LIKE '%tipo%'
GROUP BY id_unidad) as D
WHERE cantidad_vehiculo = (
SELECT MAX (cantidad_vehiculo)
FROM (SELECT id_unidad, COUNT(tipo) as cantidad_vehiculo
FROM Vehiculos
WHERE tipo LIKE '%tipo%'
GROUP BY id_unidad) as D1)

