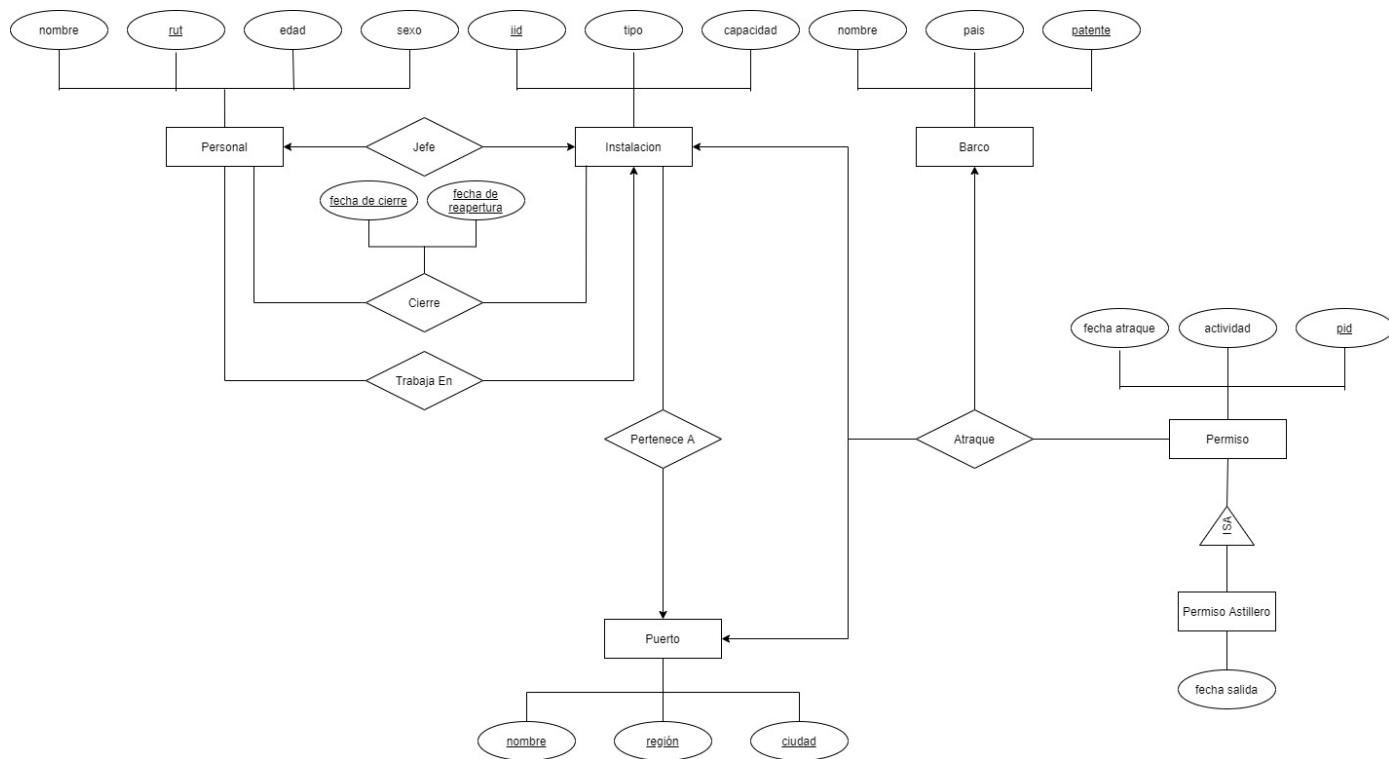


Entrega 2 BDD

Jadrina De Andrade y Ignacio Medel M.

Septiembre 2020

1 Modelo Entidad Relación



2 Esquemas

Esquema Viejo

Personal(rut char[12], nombre char[100], edad int, sexo char[1])
 Instalaciones(iid int, tipo char[100], capacidad int)
 Barcos(patente char[100], nombre char[100], pais char[100])
 Puertos(nombre char[100], region char[100], ciudad char[100])
 Permisos(pid int, fecha_atraque date, actividad char[100])
 Permisos_Astilleros(pid int, fecha_salida date)
 Jefes(rut char[12], iid int)
 Trabaja_En(rut char[100], iid int)
 Pertenece_A(iid int, *nombre_puerto* char[100], *region* char[100], *ciudad* char[100])
 Atrques(pid int, iid int, *patente* char[100], *nombre_puerto* char[100], *region* char[100], *ciudad* char[100])
 Cierres(rut char[100], iid int, fecha_cierre int, fecha_apertura int)

Esquema Nuevo

Personal(rut char[12], nombre char[100], edad int, sexo char[1])
 Instalaciones(iid int, tipo char[100], capacidad int)
 Barcos(patente char[100], nombre char[100], pais char[100])
 Puertos(nombre char[100], ciudad char[100])
 Ciudad_Region(ciudad char[100], region char[100])
 Permisos(pid int, fecha_atraque date)
 Permisos_Astilleros(pid int, fecha_salida date)
 Permisos_Muelles(pid int, actividad char[100])
 Jefes(rut char[12], iid int)
 Trabaja_En(rut char[12], iid int)
 Pertenece_A(iid int, *nombre_puerto* char[100], *ciudad* char[100])
 Atrques(pid int, iid int, *patente* char[100])
 Cierres(rut char[12], iid int, fecha_cierre int, fecha_apertura int)

Llave Foránea	Referencia
rut	Personal
iid	Instalaciones
nombre_puerto	Puertos
ciudad	Ciudad_Region/Puertos
region (solo viejo)	Puertos

Table 1: Llaves foráneas y a que entidad hacen referencia.

Cambios al esquema: Al revisar nuestro esquema preliminar encontramos dos dependencias funcionales que incluían atributos que no eran llaves.

$\text{ciudad} \rightarrow \text{region}$

$\text{iid} \rightarrow \text{nombre_puerto}, \text{region}, \text{ciudad}$

Fue por estas dependencias que tuvimos que dividir la relación Puerto en dos tablas, una que contenga el nombre del puerto y su ciudad y otra a la cual llamamos Ciudad_Region que contiene la ciudad y su respectiva región. Así evitamos redundancia en estas tablas. También con esta nueva relación pudimos sacar atributos redundantes de otras tablas como por ejemplo Pertenece_A y acortar la relación Atrake ya que su información ya estaba expresada en Pertenece_A.

Además, notamos que los astilleros poseen fecha de salida, pero no registro de actividad, y así mismo, los muelles poseen registro de actividad, pero no fecha de salida. Lo anterior llevaría a que la relación Permisos tuviese valores NULL en cada una de las instancias, y es debido a lo anterior que se dividió la relación Permisos en 3 relaciones: Permisos, Permisos_Astilleros y Permisos_Muelles, donde Permisos_Astillero y Permisos_Muelles son ISAs de Permisos.

Este nuevo esquema está en BCNF, donde las dependencias funcionales no triviales son las siguientes:

$\text{rut} \rightarrow \text{nombre}, \text{edad}, \text{sexo}$

$\text{iid} \rightarrow \text{tipo}, \text{capacidad}$

$\text{patente} \rightarrow \text{nombre}, \text{pais}$

$\text{nombre}, \text{ciudad} \rightarrow \text{region}$

$\text{ciudad} \rightarrow \text{region}$

$\text{pid} \rightarrow \text{fecha_atraque}, \text{actividad}$

$\text{pid} \rightarrow \text{fecha_salida}$

$\text{rut} \rightarrow \text{iid}$

$\text{iid} \rightarrow \text{nombre_puerto}, \text{region}, \text{ciudad}$

$\text{pid} \rightarrow \text{iid}, \text{patente}$

Se puede apreciar que todos los atributos del lado izquierdo de estas dependencias son llaves en nuestro nuevo esquema. Por lo tanto no hay anomalías y nuestro esquema está normalizado.

3 Consultas SQL

1. `SELECT * FROM puertos;` (Por el formato de nuestra tabla Puertos).
2. `SELECT personal.nombre AS Nombre_jefe,
personal.rut AS Rut_jefe,
personal.edad AS Edad_jefe,personal.sexo AS Sexo_Jefe
FROM personal,jefes,trabaja_en,pertenece_a
WHERE personal.rut=jefes.rut
AND personal.rut=trabaja_en.rut
AND trabaja_en.iid=pertenece_a.iid
AND pertenece_a.nombre_puerto = 'Mejillones';`
3. `SELECT puertos.nombre FROM puertos,pertenece_a,instalaciones
WHERE puertos.nombre=pertenece_a.nombre_puerto
AND pertenece_a.iid=instalaciones.iid
AND instalaciones.tipo='astillero'
GROUP BY puertos.nombre
HAVING COUNT(instalaciones.tipo)>0;`
4. `SELECT barcos.nombre,pertenece_a.nombre_puerto
FROM barcos,pertenece_a,atraques
WHERE barcos.patente=atraques.patente
AND atraques.iid=pertenece_a.iid
AND pertenece_a.ciudad='Arica'
AND barcos.nombre='Zarza';`
5. `SELECT puertos.nombre,
AVG(personal.edad) AS Edad_Promedio_Por_Puerto
FROM personal,trabaja_en,pertenece_a, puertos
WHERE personal.rut=trabaja_en.rut
AND trabaja_en.iid=pertenece_a.iid
AND pertenece_a.nombre_puerto=puertos.nombre
GROUP BY puertos.nombre;`
6. `SELECT puertos.nombre FROM puertos,pertenece_a,atraques,permisos
WHERE puertos.nombre=pertenece_a.nombre_puerto
AND pertenece_a.iid=atraques.iid
AND atraques.pid=permisos.pid`

```

AND permisos.fecha_ataque>'2020-08-01'
AND permisos.fecha_ataque<'2020-08-31'
GROUP BY puertos.nombre
HAVING COUNT(puertos.ciudad)> ANY (SELECT COUNT(p.ciudad)
FROM puertos AS p,pertenece_a AS pa,atraques AS a,permisos AS pe
WHERE p.nombre=pa.nombre_puerto AND pa.iid=a.iid AND a.pid=pe.pid
AND pe.fecha_ataque>'2020-08-01'
AND pe.fecha_ataque<'2020-08-31' GROUP BY p.nombre);

```

4 Supuestos y Comentarios

1. Para la sexta consulta consideramos matching parcial solo en el mes en que uno puede poner o el nombre o el número del mes, si se daba más flexibilidad se perdería la precisión por ejemplo como diferenciar "ju" de Junio o Julio. O por ejemplo si se pone 20 para el año no se sabe si el usuario quiere los datos de 1920 o 2020.
2. Para señalar las llaves foráneas en el esquema pusimos sus nombres en *itálico*.