



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
IIC2413-BASES DE DATOS

Entrega 2

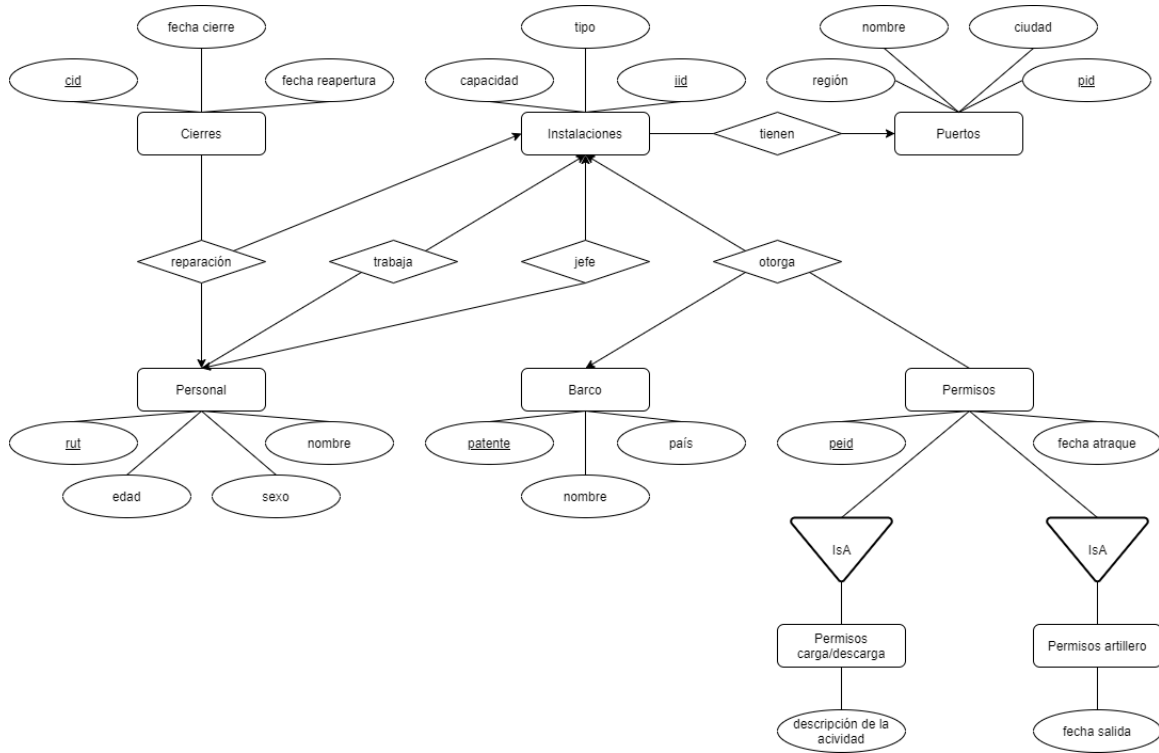
Proyecto

Belén Osse Cuevas
Benjamín Pardo Bravo
Grupo 115

Fecha entrega: 30 de Septiembre del 2020

Respuestas

■ Diagrama E/R



■ Esquema relacional Para describir las relaciones presentadas en el diagrama anterior se considerará:

“reparación” = “CierreInstalacion”

“trabaja” = “PersonalInstalacion”

“tienen” = “PuertosInstalaciones”

“jefe” = “JefeInstalacion”

“otorga” = “PermisoOtorgado”

- Puertos(**pid**: int, nombre: VARCHAR(25), ciudad: VARCHAR(25), region: VARCHAR(25))
- Instalaciones(**iid**: int, tipo: string, capacidad: int)
- PuertosInstalaciones(**iid**: int, pid: int)
pid: Llave foránea relacionada a “Puertos”
iid: Llave foránea relacionada a “Instalaciones”
- Cierres(**cid**: int, cierre: timestamp, reapertura: timestamp)
- CierreInstalacion(**cid**: int, iid: int, rut: VARCHAR(10))
cid: Llave foránea relacionada a “Cierres”
iid: Llave foránea relacionada a “Instalaciones”
rut: Llave foránea relacionada a “Personal”
- Personal(**rut**: VARCHAR(10), nombre: VARCHAR(25), edad: int, sexo: string)
- JefeInstalacion(**rut**: VARCHAR(10), iid: int)
rut: Llave foránea relacionada a “Personal”
iid: Llave foránea relacionada a “Instalaciones”
- PersonalInstalacion(**rut**: VARCHAR(10), iid: int)
iid: Llave foránea relacionada a “Instalaciones”
rut: Llave foránea relacionada a “Personal”

- Barco(**patente: string**, país: VARCHAR(25), nombre: VARCHAR(25))
- Permisos(**peid: int**, ataque: timestamp)
- PermisosCargaDescarga(**peid: int**, descripcion: VARCHAR(100))
peid: Llave foránea relacionada a “Permisos”
- PermisosAstillero(**peid: int**, salida: timestamp)
peid: Llave foránea relacionada a “Permisos”
- PermisoOtorgado(**peid: int**, iid: int, patente: string)
peid: Llave foránea relacionada a “Permisos”
iid: Llave foránea relacionada a “Instalaciones”
patente: Llave foránea relacionada a “Barco”

■ Justificación

El esquema está en BCNF, para mostrarlo se verá que cada tabla está en BCNF.

1. Puertos: *pid* es la llave, y ya que tanto nombre como ciudad y región pueden estar repetidos entre tuplas, no existe ninguna dependencia entre los otros atributos. Esto es, con ningún conjunto posible de atributos ($\{\text{nombre}\}$, $\{\text{ciudad}\}$, $\{\text{región}\}$, $\{\text{nombre, ciudad}\}$, $\{\text{nombre, región}\}$, $\{\text{ciudad, región}\}$, $\{\text{nombre, ciudad, región}\}$) se puede determinar el valor del resto.
En otras palabras, todo conjunto de atributos que determine al resto, contiene al atributo *pid* (o sea es una superkey).
2. Instalaciones: Igual que el caso anterior, todos los atributos (salvo la llave) pueden tener valores repetidos entre tuplas, por lo que es fácil ver que no existen dependencias aparte de $\text{iid} \rightarrow \{\text{tipo, capacidad}\}$, y que todo conjunto de atributos que determine al resto tendrá *iid* (será superkey). A partir de ahora se omitirá esto último en los casos similares.
3. PuertosInstalaciones: En este caso es fácil ver que está en BCNF, porque hay 2 atributos y solo 1 de ellos puede ser llave del otro: $\text{iid} \rightarrow \text{pid}$.
4. Cierres: Nuevamente, la única dependencia que hay es la de la llave $\text{cid} \rightarrow \{\text{cierre, reapertura}\}$, caso similar a 1 y 2.
5. CierreInstalación: $\text{cid} \rightarrow \{\text{iid, rut}\}$, y no hay ninguna otra dependencia.
6. Personal: La única dependencia que hay es la que proporciona la llave minimal (primaria): $\text{rut} \rightarrow \{\text{nombre, edad, sexo}\}$. Caso similar a 1 y 2.
7. JefeInstalacion: en este caso ambos atributos pueden ser llave, dado que ambos definen inequívocamente una tupla, por lo tanto ambos son parte de una posible llave minimal. Se cumple que para $X \rightarrow Y$, X es llave, para cualquiera de las 2 combinaciones.
8. PersonalInstalacion: $\text{rut} \rightarrow \text{iid}$, y no al revés.
9. Barco: $\text{Patente} \rightarrow \{\text{país, nombre}\}$, caso similar a 1 y 2.
10. Permisos: $\text{peid} \rightarrow \text{ataque}(\text{date})$, puede haber más de un *peid* para una fecha específica pero no al revés.
11. PermisosAstillero: $\text{peid} \rightarrow \{\text{descripción, carga, ataque}\}$, y no hay otra dependencia que no use *peid*. Además, *peid* es llave foránea que hace referencia a Permisos.*peid*, así no se puede agregar un permiso astillero sin agregar un permiso.
12. PermisosCargaDescarga: $\text{peid} \rightarrow \{\text{salida, ataque}\}$, y no hay otra dependencia que no use *peid*. Además, *peid* es llave foránea que hace referencia a Permisos.*peid*, así no se puede agregar un permiso carga/descarga sin agregar un permiso.
13. PermisoOtorgado: $\text{peid} \rightarrow \{\text{pid, patente}\}$, toda dependencia incluye a *peid* al lado izquierdo, caso similar al 1 y 2.

■ Consultas

En esta parte se mostrarán las consultas escritas de dos maneras distintas, en donde la primera será la pedida (y resuelta en relación a lo que se presentó con anterioridad) y la segunda será a como la consulta fue implementada en la página.

1. Muestre todos los puertos junto la ciudad a la que son asignados.

```
SELECT nombre, ciudad
FROM Puertos
```

```
SELECT nombre, ciudad
FROM puertos
WHERE upper(nombre) LIKE '%$puerto%'
```
2. Muestre todos los jefes de las instalaciones del puerto con nombre “Mejillones”.

```
SELECT Personal.rut, Personal.nombre, Personal.edad, Personal.sexo
FROM Puertos INNER JOIN PuertosInstalaciones INNER JOIN Instalaciones INNER
JOIN JefeInstalacion INNER JOIN Personal
ON Puertos.pid = PuertosInstalaciones.pid
AND PuertosInstalaciones.iid = Instalaciones.iid
AND JefeInstalacion.iid = Instalaciones.iid
AND Personal.rut = JefeInstalacion.rut
WHERE Puertos.nombre = “Mejillones”
```

```
SELECT personal.rut, personal.nombre, personal.edad, personal.sexo
FROM puertos, puertosinstalaciones, instalaciones, jefeinstalacion, personal
WHERE puertos.pid = puertosinstalaciones.pid
AND puertosinstalaciones.iid = instalaciones.iid
AND jefeinstalacion.iid = instalaciones.iid
AND personal.rut = jefeinstalacion.rut
AND upper(puertos.nombre) LIKE '%$puerto%'
```
3. Muestre todos los puertos que tienen al menos un astillero.

```
SELECT Puertos.nombre
FROM Puertos INNER JOIN PuertosInstalaciones INNER JOIN Instalaciones
ON Puertos.pid = PuertosInstalaciones.pid
AND PuertosInstalaciones.iid = Instalaciones.iid
WHERE tipo = “astillero”
```

```
SELECT nombre, cantidad
FROM (SELECT puertos.nombre, COUNT (instalaciones.iid) AS cantidad
FROM puertos, puertosinstalaciones, instalaciones
WHERE puertos.pid = puertosinstalaciones.pid
AND puertosinstalaciones.iid = instalaciones.iid
AND tipo = ‘astillero’
GROUP BY puertos.nombre) AS tabla
WHERE tabla.cantidad >= $astilleros
```
4. Muestre todas las veces en que el barco “Calypso” ha atracado en “Arica”.

```
SELECT Permisos.atraque
FROM Barco INNER JOIN PermisoOtorgado INNER JOIN Instalaciones INNER JOIN
PuertosInstalaciones INNER JOIN Puertos INNER JOIN Permisos
ON Barco.patente = PermisoOtorgado.patente
AND Instalaciones.iid = PuertosInstalaciones.iid
AND PuertosInstalaciones.pid = Puertos.pid
AND Instalaciones.iid = PermisoOtorgado.iid
AND Permisos.peid = PermisoOtorgado.peid
WHERE Barco.nombre = “Calypso” AND Puertos.ciudad = “Arica”
```

```

SELECT barco.nombre, permisos.atraque, puertos.nombre
FROM barco, permisootorgado, instalaciones, puertosinstalaciones, puertos, permisos
WHERE barco.patente = permisootorgado.patente
AND instalaciones.iid = puertosinstalaciones.iid
AND puertosinstalaciones.pid = puertos.pid
AND instalaciones.iid = permisootorgado.iid
AND permisos.peid = permisootorgado.peid
AND upper(barco.nombre) LIKE '%$barco%'
AND upper(puertos.ciudad) LIKE '%$ciudad%'

```

5. Muestre la edad promedio de los trabajadores de cada puerto.

```

SELECT Puertos.nombre, AVG(edad)
FROM Personal INNER JOIN PersonalInstalacion INNER JOIN Instalaciones INNER
JOIN PuertosInstalaciones INNER JOIN Puertos
ON Personal.rut = PersonalInstalacion.rut
AND PersonalInstalacion.iid = Instalaciones.iid
AND Instalaciones.iid = PuertosInstalaciones.iid
AND PuertosInstalaciones.pid = Puertos.pid
GROUP BY Puertos.nombre

```

```

SELECT puertos.nombre, AVG(edad) AS edad_promedio
FROM personal, personalinstalacion, instalaciones, puertosinstalaciones, puertos
WHERE personal.rut = personalinstalacion.rut
AND personalinstalacion.iid = instalaciones.iid
AND instalaciones.iid = puertosinstalaciones.iid
AND puertosinstalaciones.pid = puertos.pid
AND upper(puertos.nombre) LIKE '%$puerto%'
GROUP BY puertos.nombre

```

6. Muestre el puerto que ha recibido más barcos en Agosto del 2020.

```

SELECT nombre, MAX(cantidad)
FROM (SELECT Puertos.nombre, COUNT (Puertos.pid) AS cantidad
FROM Permisos INNER JOIN PermisoOtorgado INNER JOIN Instalaciones INNER
JOIN PuertosInstalaciones INNER JOIN Puertos
ON Permisos.peid = PermisoOtorgado.peid
AND PermisoOtorgado.iid = Instalaciones.iid
AND Instalaciones.iid = PuertosInstalaciones.iid
AND PuertosInstalaciones.pid = Puertos.pid
WHERE atraque >= "2020-08-01" AND atraque < "2020-09-01"
GROUP BY Puertos.pid)

```

```

SELECT nombre, MAX(barcos_recibidos) AS cantidad
FROM (SELECT puertos.nombre, COUNT (puertos.pid) AS barcos_recibidos
FROM permisos, permisootorgado, instalaciones, puertosinstalaciones, puertos
WHERE permisos.peid = permisootorgado.peid
AND permisootorgado.iid = instalaciones.iid
AND instalaciones.iid = puertosinstalaciones.iid
AND puertosinstalaciones.pid = puertos.pid
AND atraque >= '$año-$mes-01'
AND atraque < '$año2-$mes2-01'
GROUP BY puertos.pid) AS tabla
GROUP BY tabla.nombre

```

■ Supuestos/Detalles

En la parte de *Esquema relacional* se considerará que las llaves primarias son aquellas que están subrayadas y ennegrecidas.

En la parte de *Consultas* se separaron con salto de línea diversos conectores de manera que fuese más fácil leer.

Las consultas se pusieron de dos maneras debido a que en el enunciado decía en la primera parte que se debían agregar las consultas del ejemplo y al final del enunciado aparecía que se debían agregar las utilizadas en la aplicación.

Las relaciones en las consultas de la aplicación están escritas en minúsculas dado que las estas en la base de datos fueron escritas así. En estas mismas se utiliza *upper* para así poder hacer *case-insensitive*.

Esperamos que te guste la página, está muy cool jj.