

Grupo: 14

Nombre Integrante 1: Daniela Corral

Sección: 2

Nombre integrante 2: Matías Masjuan

Sección: 2



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2413 — Bases de Datos — 2' 2020

Entrega 2

1. Modelo

A continuación se mostrará el modelo relacional utilizando los conceptos aprendidos en clases, donde incluiremos tanto el diagrama como el esquema relacional.

1.1. Diagrama

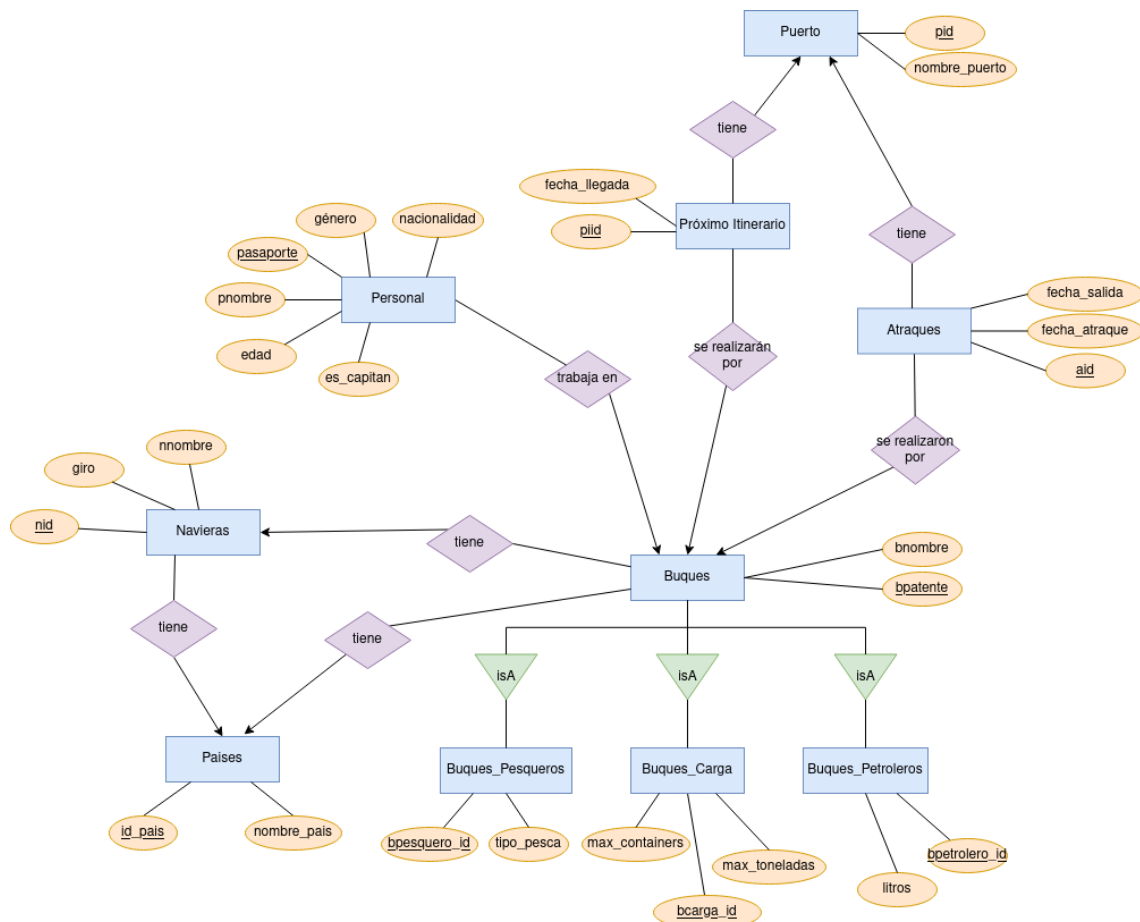


Figura 1: Diagrama Entidad/Relación

1.2. Esquema

Navieras (nid: int, nnombre: varchar(50), idpais: int, giro: varchar(50))

Buques (bpatente: varchar(50), bnombre: varchar(50), idpais: int, nid: int)

Buques_Pesqueros (bpesquero_id: int, bpatente: varchar(50), tipo_pesca: varchar(50))

Buques_Carga (bcarga_id: int, bpatente: varchar(50), max_containers: int, max_toneladas: int)

Buques_Petroleros (bpetrolero_id: int, bpatente: varchar(50), litros: int)

Personal (pasaporte: varchar(50), pnombre: varchar(50), edad: int, genero: varchar(50), nacionalidad: varchar(50), bpatente: varchar(50) es_capitan: bool)

Atraques (aid: int, bpatente: varchar(50), pid: int, fecha_atraque: timestamp, fecha_salida: timestamp)

ProximoItinerario (piid: int, bpatente: varchar(50), pid: int, fecha_llegada: timestamp)

Países (idpais: int, nombre_pais: varchar(50))

Puertos(pid: int, nombre_puerto: varchar(50))

2. Justificación

A continuación se hará un listado de las relaciones definidas en el esquema anterior junto a las razones de por qué se encuentran en BCNF.

1. Navieras (nid: int, nnombre: varchar(50), idpais: int, giro: varchar(50))

Actualmente la llave es nid, ya que con ella podemos obtener el nnombre, el pais y el tipo de giro de la naviera. El nnombre no nos sirve como llave, ya que existe la posibilidad, aunque sea escasa, de que hayan dos navieras con el mismo nombre. Lo mismo sucede con el idpais, ya que pueden haber distintas navieras con el mismo pais. Por último, el giro tampoco es considerado una llave, ya que pueden haber distintas navieras con el mismo giro. Como el nid es el único atributo que implica al resto y además es llave, entonces esta relación se encuentra en BCNF.

2. Buques (bpatente: varchar(50), bnombre: varchar(50), idpais: int, nid: int)

Actualmente la llave es bpatente, ya que con ella podemos obtener el bnombre, el pais y la naviera a la que pertenece. El bnombre no nos sirve como llave, ya que existe la posibilidad, aunque sea escasa, de que hayan dos buques con el mismo nombre. Lo mismo sucede con el idpais, ya que pueden haber distintos buques con el mismo pais. Por último, como cada buque pertenece a una naviera, es redundante incluir todos los atributos de la naviera en esta relación, por lo que se utiliza una llave foránea del nid para contar con estos atributos. Cabe recalcar que si conocemos el buque, conocemos la naviera, pero no es el caso contrario ya que una naviera puede tener múltiples buques. Como la bpatente es el único atributo que implica al resto y además es llave, entonces esta relación se encuentra en BCNF.

3. Buques_Pesqueros (bpesquero_id: int, bpatente: varchar(50), tipo_pesca: varchar(50))

Actualmente la llave es bpesquero_id, ya que con ella podemos obtener la bpatente y el tipo de pesca. El tipo de pesca no nos sirve como llave, ya que hay muchos buques distintos con el mismo tipo de pesca. No es muy conveniente utilizar la bpatente como llave, ya que también es la llave de Buques, por lo que en ese caso correspondería ingresar todos los atributos de los Buques_Pesqueros en la relación Buques. Esto provocaría una anomalía en la relación Buques, ya que no todos son considerados como Buques_Pesqueros, lo que podría generar muchos valores vacíos o nulos. El hecho de tener el identificador bpesquero_id permitiría tener información solo acerca de los buques pesqueros incluyendo el tipo de pesca que no es considerado dentro de la relación Buques, y nos evitamos el problema de tratar con valores nulos. Por lo tanto, dejando el bpesquero_id como llave y la bpatente como llave foránea, se estaría respetando el estado de BCNF.

4. Buques_Carga (bcarga_id: int, bpatente: varchar(50), max_containers: int, max_toneladas: int)

Actualmente la llave es bcarga_id, ya que con ella podemos obtener la bpatente, el max_containers y max_toneladas. Tanto los max_containers como el max_toneladas no nos sirven como llaves, ya que hay muchos buques distintos con las mismas capacidades de contenedores y las toneladas máximas que pueden sostener. No es muy conveniente utilizar la bpatente como llave, ya que también es la llave de Buques, por lo que en ese caso correspondería ingresar todos los atributos de los Buques_Carga en la relación Buques. Esto provocaría una anomalía en la relación Buques, ya que no todos son considerados como Buques_Carga, lo que podría generar muchos valores vacíos o nulos. El hecho de tener el identificador bcarga_id permitiría tener información solo acerca de los buques de carga incluyendo el max_containers y el max_toneladas que no es considerado dentro de la relación Buques, y nos evitamos el problema de tratar con valores nulos. Por lo tanto, dejando el bcarga_id como llave y la bpatente como llave foránea, se estaría respetando el estado de BCNF.

5. Buques_Petroleros (bpetrolero_id: int, bpatente: varchar(50), litros: int)

Actualmente la llave es bpetrolero_id, ya que con ella podemos obtener la bpatente y los litros del buque petrolero. Los litros no nos sirven como llave, ya que hay muchos buques distintos con la misma cantidad de litros. No es muy conveniente utilizar la bpatente como llave, ya que también es la llave de Buques, por lo que en ese caso correspondería ingresar todos los atributos de los Buques_Petroleros en la relación Buques. Esto provocaría una anomalía en la relación Buques, ya que no todos son considerados como Buques_Petroleros, lo que podría generar muchos valores vacíos o nulos. El hecho de tener el identificador bpetrolero_id permitiría tener información solo acerca de los buques petroleros incluyendo la cantidad de litros que no es considerado dentro de la relación Buques, y nos evitamos el problema de tratar con valores nulos. Por lo tanto, dejando el bpetrolero_id como llave y la bpatente como llave foránea, se estaría respetando el estado de BCNF.

6. Personal (pasaporte: varchar(50), pnombre: varchar(50), edad: int, genero: varchar(50), nacionalidad: varchar(50), bpatente: varchar(50) es_capitan: bool)

Actualmente la llave es pasaporte, ya que, como identificador único de una persona, con él podemos obtener el nombre, edad, género y nacionalidad de la misma. Ninguno de estos atributos recién mencionados sirven como llave, ya que existe la posibilidad de que hayan dos personas con el mismo nombre, con la misma edad, del mismo género o nacionalidad. Cabe recalcar que si conocemos a la persona, sabremos en qué buque trabaja, lo cual no aplica en sentido contrario pues en un buque hay múltiples empleados. Finalmente, si conocemos a la persona, podemos saber su profesión y en particular si ésta consiste en la capitanía o no. Sin embargo, no podemos conocer a la persona solo sabiendo su profesión pues muchos pueden tener el mismo cargo (hay muchos tripulantes y capitanes en la base de datos). Como el pasaporte es el único atributo que implica al resto y además es llave, entonces esta relación se encuentra en BCNF.

7. Atrques (aid: int, bpatente: varchar(50), pid: int, fecha_atraque: timestamp, fecha_salida: timestamp)

Actualmente la llave es aid, ya que al ser un identificador en sí, con él podemos acceder a toda la información relativa al atraque. La patente del buque no nos sirve como llave, ya que un mismo buque puede haber efectuado múltiples atraques durante su tiempo de funcionamiento. Lo mismo sucede con las fechas, ya que pueden haber atracado o partido dos buques, por improbable que sea, en el mismo instante. Cabe recalcar que si conocemos el atraque, conocemos por tanto el puerto en que se efectuó, pero no es el caso contrario ya que un puerto pudo haber sido visitado por múltiples buques. Para minimizar redundancias, la información relativa a los puertos (que es tarea de los grupos impares) y buques en esta tabla se referencia solo con el identificador del puerto. Como el aid es el único atributo que implica al resto y además es llave, entonces esta relación se encuentra en BCNF.

8. ProximoItinerario (piid: int, bpatente: varchar(50), pid: int, fecha_llegada: timestamp)

Actualmente la llave es piid, ya que al ser un identificador en sí, con él podemos acceder a toda la información relativa al itinerario. La patente del buque no nos sirve como llave, ya que un mismo buque puede tener agendados múltiples atraques para el futuro. Lo mismo sucede con la fecha de llegada, ya que puede planificarse la llegada de dos buques, por improbable que sea, en el mismo instante. Cabe

recaltar que si conocemos el itinerario, conocemos por tanto el puerto en que se efectuó, pero no es el caso contrario ya que un puerto puede tener agendadas visitas de múltiples buques. Para minimizar redundancias, la información relativa a los puertos (que es tarea de los grupos impares) y buques en esta tabla se referencia solo con el identificador del puerto. Como el `pid` es el único atributo que implica al resto y además es llave, entonces esta relación se encuentra en BCNF.

9. Países (`idpais`: int, `nombre_pais`: varchar(50))

Tabla creada con el fin de evitar redundancias dada muchas repeticiones de los strings (VARCHAR) que componen los nombres de los países, además de evitar anomalías de actualización en que si se quisiera eventualmente alterar el nombre de un país no se tengan que editar todas las tuplas de las tablas Navieras y Buques que lo contengan. Si, en caso contrario, hubiéramos puesto los nombres de los países directamente como atributos de las tablas Navieras y Buques, correríamos el riesgo de tener en nuestra base de datos anomalías de eliminación, ya que de tener una única instancia proveniente de un cierto país y yo quisiera eliminar dicha tupla, estaría borrando de la base de datos con ella el país completo. Además, sabemos que los países para futuras entregas podrían potencialmente contar con nuevos atributos y referencias en otras tablas nuevas, por lo que contar con esta tabla nos hace este aspecto mucho más escalable. Podemos ver sin dificultad que la tabla Países cumple con el modelo BCNF ya que tenemos un identificador que cumple la función de llave, el cual determina el nombre del país.

10. Puertos(`pid`: int, `nombre_puerto`: varchar(50))

Tabla referencial creada con el único fin de poder usar los puertos en las consultas relacionadas a los ataques y próximos itinerarios. Posiblemente en las próximas entregas se agreguen más atributos a ellos. Por el momento, solo consiste en un identificador `pid` junto con el nombre del puerto, los cuales evidentemente respetan el BCNF ya que el identificador determina el nombre del puerto y es llave.

3. Consultas

1. Muestre el nombre de todas las navieras.

```
SELECT Navieras.nnombre FROM Navieras;
```

2. Muestre todos los buques de la naviera 'Francis Drake S.A.'.

```
SELECT Buques.bpatente, Buques.bnombre, Buques.idpais, Buques.nid FROM Buques,
Navieras WHERE Buques.nid = Navieras.nid AND LOWER(Navieras.nnombre) LIKE
LOWER('%Francis Drake S.A.%') ORDER BY Buques.nid;
```

3. Muestre todos los buques que hayan atracado en 'Valparaiso' el 2020.

```
SELECT DISTINCT Buques.bpatente, Buques.bnombre, Buques.idpais, Buques.nid FROM
Buques, Ataques, Puertos WHERE Buques.bpatente = Ataques.bpatente AND Puertos.pid =
Ataques.pid AND LOWER(Puertos.nombre_puerto) LIKE LOWER('%Valparaiso%') AND
Ataques.fecha_atraque <= '2020-12-31' AND Ataques.fecha_atraque >= '2020-01-01';
```

4. Muestre todos los buques que hayan estado en 'Mejillones' al mismo tiempo que el buque 'Magnolia'¹.

```
SELECT DISTINCT Buques.bpatente, Buques.bnombre, Buques.idpais, Buques.nid FROM
Buques, Ataques, Puertos, (SELECT Buques.bpatente, Ataques.fecha_atraque,
Ataques.fecha_salida FROM Buques, Ataques, Puertos WHERE Buques.bpatente =
Ataques.bpatente AND Ataques.pid = Puertos.pid AND LOWER(Buques.bnombre) LIKE
LOWER('%Magnolia%') AND LOWER(Puertos.nombre_puerto) LIKE LOWER('%Mejillones%')) AS
Periodo WHERE Buques.bpatente = Ataques.bpatente AND Ataques.pid = Puertos.pid AND
LOWER(Puertos.nombre_puerto) LIKE LOWER('%Mejillones%') AND Ataques.fecha_atraque
```

¹ Asumimos por la issue #173 que para que esto pase, el periodo en que permanezcan en Mejillones los buques que cumplan con la consulta debe estar completamente autocontenido en el del buque Magnolia: no pueden llegar antes ni irse después que el buque Magnolia.

```
>= Periodo.fecha_atraque AND Atrques.fecha_salida <= Periodo.fecha_salida AND
Periodo.bpatente != Buques.bpatente;
```

5. Encuentre todos los capitanes mujeres que han pasado por el puerto ‘Talcahuano’.

```
SELECT DISTINCT Personal.pasaporte, Personal.pnombre, Personal.edad, Personal.genero,
Personal.nacionalidad, Personal.bpatente, Personal.es_capitan FROM Personal, Buques,
Atrques, Puertos WHERE Personal.bpatente = Buques.bpatente AND Buques.bpatente =
Atrques.bpatente AND Atrques.pid = Puertos.pid AND LOWER(Personal.genero) LIKE
LOWER('%mujer%') AND Personal.es_capitan = true AND LOWER(Puertos.nombre_puerto)
LIKE LOWER('%Talcahuano%');
```

6. Encuentre el buque pesquero que tiene más personas trabajando.

```
SELECT Buques_Pesqueros.bpesquero_id, Buques_Pesqueros.bpatente,
Buques_Pesqueros.tipo_pesca FROM Buques_Pesqueros, Personal WHERE Personal.bpatente =
Buques_Pesqueros.bpatente GROUP BY Buques_Pesqueros.bpesquero_id HAVING COUNT(*) =
(SELECT MAX(Trabajadores.cantidad) AS maximo FROM (SELECT COUNT(*) AS Cantidad FROM
Buques_Pesqueros, Personal WHERE Personal.bpatente = Buques_Pesqueros.bpatente GROUP
BY Personal.bpatente) AS Trabajadores);
```

4. Consideraciones Adicionales

Los archivos csv para poblar la base de datos se ubicaron en la carpeta home/Entrega2. Estos archivos no tienen relación directa con la aplicación web, por lo que no se ubicaron en la carpeta Sites. Por lo tanto, estarán ubicados en la misma carpeta que el informe. Esperamos que no sea un inconveniente.