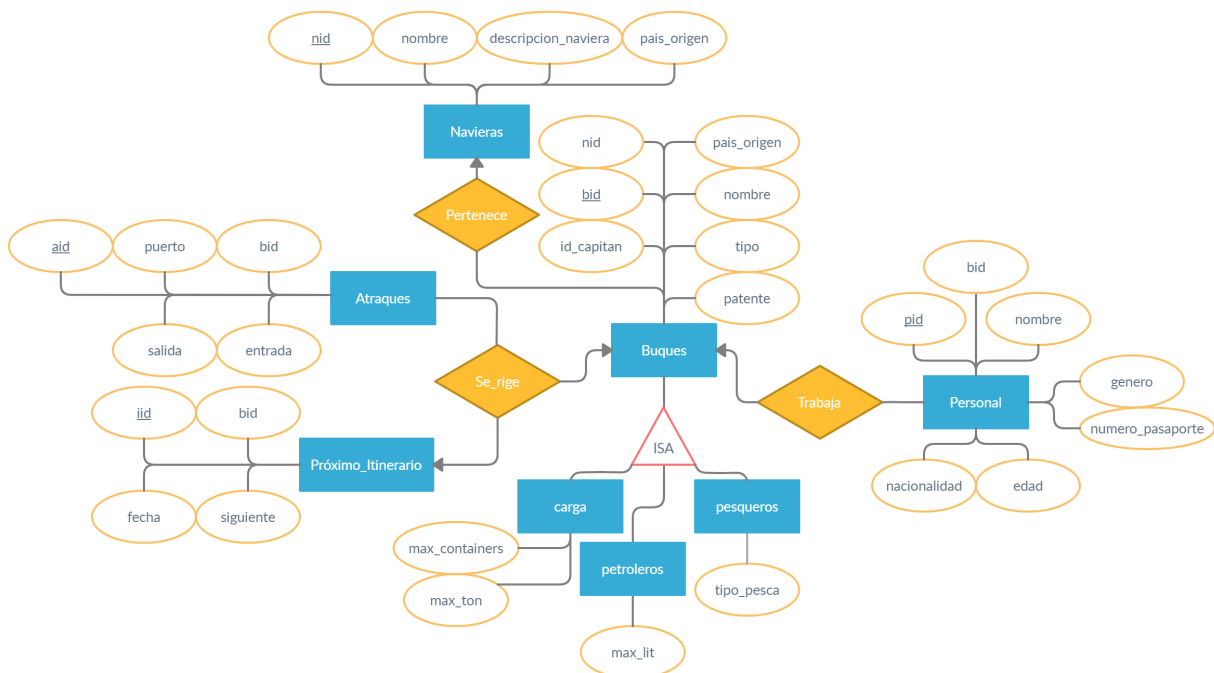


Entrega 02

grupo100 (Beatriz Acuña - Cristóbal Rubio)

Diagrama E/R



Esquema Relacional

1. Navieras(nid: int, nombre: string, pais_origen: string, descripcion_naviera: string)
2. Buques(bid: int, nid: int, nombre: string, patente: string, país_origen: string, tipo: string, id_capitan: int)
3. Personal(pid: int, bid: int, nombre: string, genero: string, edad: int, nacionalidad: string, numero_pasaporte: int)

4. Atraques(aid: int, bid: int, puerto: string, entrada: date, salida: date)
5. Próximo_Itinerario(iid: int, bid: int, siguiente: string, fecha: date)
6. Pesqueros(bid: int, tipo_pesca: string)
7. Carga(bid: int, max_containers: int, max_ton: int)
8. Petroleros(bid: int, max_lit: int)

Justificación Esquema

Escogimos BCNF para el modelo del esquema. A continuación se muestran las dependencias funcionales:

- Navieras: $nid \rightarrow nombre, pais_origen, descripcion_naviera$
Sabiendo el identificador de Navieras podemos saber todos los otros valores, ya que no hay dos navieras con el mismo id que tengan distinto nombre, país de origen y/o descripción de naviera.
- Buques: $bid \rightarrow nid, nombre, patente, pais, tipo, id_capitan$
Si sabemos el identificador del buque podemos obtener todos los otros atributos, debido a que este identificador es único en su clase. Esto no ocurre con los demás atributos, ya que estos pueden o no ser únicos dentro de su categoría, o ser actualizados en el tiempo.
- Pesqueros: $bid \rightarrow tipo_pesca$
Sabiendo el identificador del buque, siendo este pesquero, podemos obtener el tipo de pesca de este. Esto no ocurre al sentido contrario ya que muchos buques pesqueros comparte tipos de pesca.
- Carga: $bid \rightarrow max_containers, max_ton$
Sabiendo el identificador del buque, siendo este de carga, podemos obtener el máximo de containers y toneladas que puede cargar. Esto no ocurre al sentido contrario ya que buques de carga pueden tener el mismo valor de estos atributos.
- Petroleros: $bid \rightarrow max_lit$
Sabiendo el identificador del buque, siendo este petrolero, podemos obtener el máximo de litros que puede transportar. Esto no ocurre al sentido contrario ya que más de un buque petrolero puede tener un valor para max_lit .
- Personal: $pid \rightarrow bid, nombre, genero, edad, nacionalidad, numero_pasaporte$
Con el identificador de personal se pueden identificar todos los otros atributos ya que este valor es único. Esto no ocurre en sentido contrario ya que muchas

personas pueden compartir los valores de los atributos nombre, bid, genero, edad y/o nacionalidad. Además, el número de pasaporte puede cambiar en el tiempo, por lo que este atributo no puede ser llave de la relación.

- **Atraques:** $\text{aid} \rightarrow \text{bid}, \text{puerto}, \text{entrada}, \text{salida}$
Con el identificador de atraque se pueden obtener la información del puerto, el identificador del buque, la entrada y/o la salida de este atraque, ya que este valor es único en su categoría. Notamos que los atraques pueden tener el mismo valor en todos los demás atributos por lo que estos no pueden ser la llave de la relación
- **Proximo_itinerario:** $\text{iid} \rightarrow \text{bid}, \text{siguiente}, \text{fecha}$
Si se sabe el identificador del itinerario, se puede saber el identificador del buque, el siguiente puerto por visitar y la fecha, pero esto no ocurre en sentido cualquiera, sea cual sea la combinación de los atributos restantes, ya que estos no son únicos dentro de la relación.

Notamos en un principio que todas las dependencias mostradas de la forma $X \rightarrow Y$ tienen como X a la llave de la relación, por lo que se está cumpliendo BCNF. Luego, sabemos que las llaves escogidas son únicas, debido a que todas fueron definidas de esa forma al ser representadas por identificadores, en algunos casos venían descritos en los archivos csv, y en otros fueron creados por nosotros.

Consultas en SQL

1. `SELECT nombre
FROM navieras;`
2. `SELECT *
FROM buques
INNER JOIN navieras
ON buques.nid = navieras.nid
WHERE UPPER(navieras.nombre)
LIKE UPPER('%francis drake s.a.%');`
3. `SELECT *
FROM buques
INNER JOIN atraques
ON buques.bid = atraques.bid
WHERE UPPER(atraques.puerto)
LIKE UPPER('%Valpara%')
AND entrada`

```
BETWEEN '20200101 00:00:00'
AND '20201231 23:59:59';
```

4.

```
SELECT *
FROM
(SELECT a.entrada, a.salida
FROM
(SELECT *
FROM buques
INNER JOIN atraques
ON buques.bid = atraques.bid
WHERE UPPER(buques.nombre)
LIKE UPPER('%magnolia%')) AS a
WHERE UPPER(a.puerto)
LIKE UPPER('%mejillones%')) AS b,
(SELECT buques.bid, buques.nid, buques.nombre, buques.patente,
buques.pais_origen, buques.tipo, buques.id_capitan,
atraques.puerto, atraques.entrada, atraques.salida
FROM buques
INNER JOIN atraques
ON buques.bid = atraques.bid
WHERE UPPER(atraques.puerto)
LIKE UPPER('%mejillones%')) AS c
WHERE c.entrada
BETWEEN b.entrada
AND b.salida
OR c.salida
BETWEEN b.entrada
AND b.salida;
```
5.

```
SELECT *
FROM
(SELECT buques.bid, buques.nombre, buques.id_capitan,
personal.nombre, personal.genero, personal.edad,
personal.nacionalidad, personal.numero_pasaporte
FROM buques
INNER JOIN personal
ON buques.id_capitan = personal.pid
WHERE personal.genero = 'mujer') AS a,
(SELECT buques.bid, atraques.puerto
FROM buques
```

```

INNER JOIN atraques
ON buques.bid = atraques.bid
WHERE UPPER(atraques.puerto)
LIKE UPPER('%talcahuano%')) AS b
WHERE a.bid = b.bid;

```

```

6. SELECT *
FROM
(SELECT buques.bid, buques.nid, buques.nombre,
buques.patente, buques.pais_origen, buques.tipo,
buques.id_capitan, COUNT(buques.bid) AS trabajadores
FROM buques
INNER JOIN personal
ON buques.bid = personal.bid
GROUP BY buques.bid
ORDER BY trabajadores DESC) AS a,
(SELECT MAX(c.trabajadores)
FROM(SELECT buques.bid, buques.nid, buques.nombre,
buques.patente, buques.pais_origen, buques.tipo,
buques.id_capitan, COUNT(buques.bid) AS trabajadores
FROM buques
INNER JOIN personal
ON buques.bid = personal.bid
GROUP BY buques.bid
ORDER BY trabajadores DESC) AS c) AS b
WHERE a.trabajadores = b.max;

```