

Entrega 2

1. Diagrama E/R

En la Figura 1 se presenta el Diagrama E/R diseñado para el problema.

2. Esquema relacional

A continuación se muestra el esquema relacional del diagrama expuesto anteriormente.

- ports (pid *integer*, name *character varying(100)*, cid *integer*)
- cities (cid *integer*, name *character varying(100)*, region *character varying(100)*)
- facilities (fid *integer*, type *character varying(50)*, rut *character varying(50)*, pid *integer*, capacity *integer*)
- facility_history_entries (fheid *integer*, closed_on *timestamp without time zone*, opened_on *timestamp without time zone*, closeBoss_rut *character varying(50)*)
- employees (rut *character varying(50)*, name *character varying(100)*, age *integer*, sex *character varying(50)*, fid *integer*)
- ships (license_plate *character varying(50)*, name *character varying(100)*, country *character varying(100)*)
- shipyard_permits (spid *integer*, fid *integer*, license_plate *character varying(50)*, arrival_date *timestamp without time zone*, departure_date *timestamp without time zone*)
- dock_permits (dpid *integer*, fid *integer*, license_plate *character varying(50)*, arrival_date *timestamp without time zone*, description *character varying(400)*)

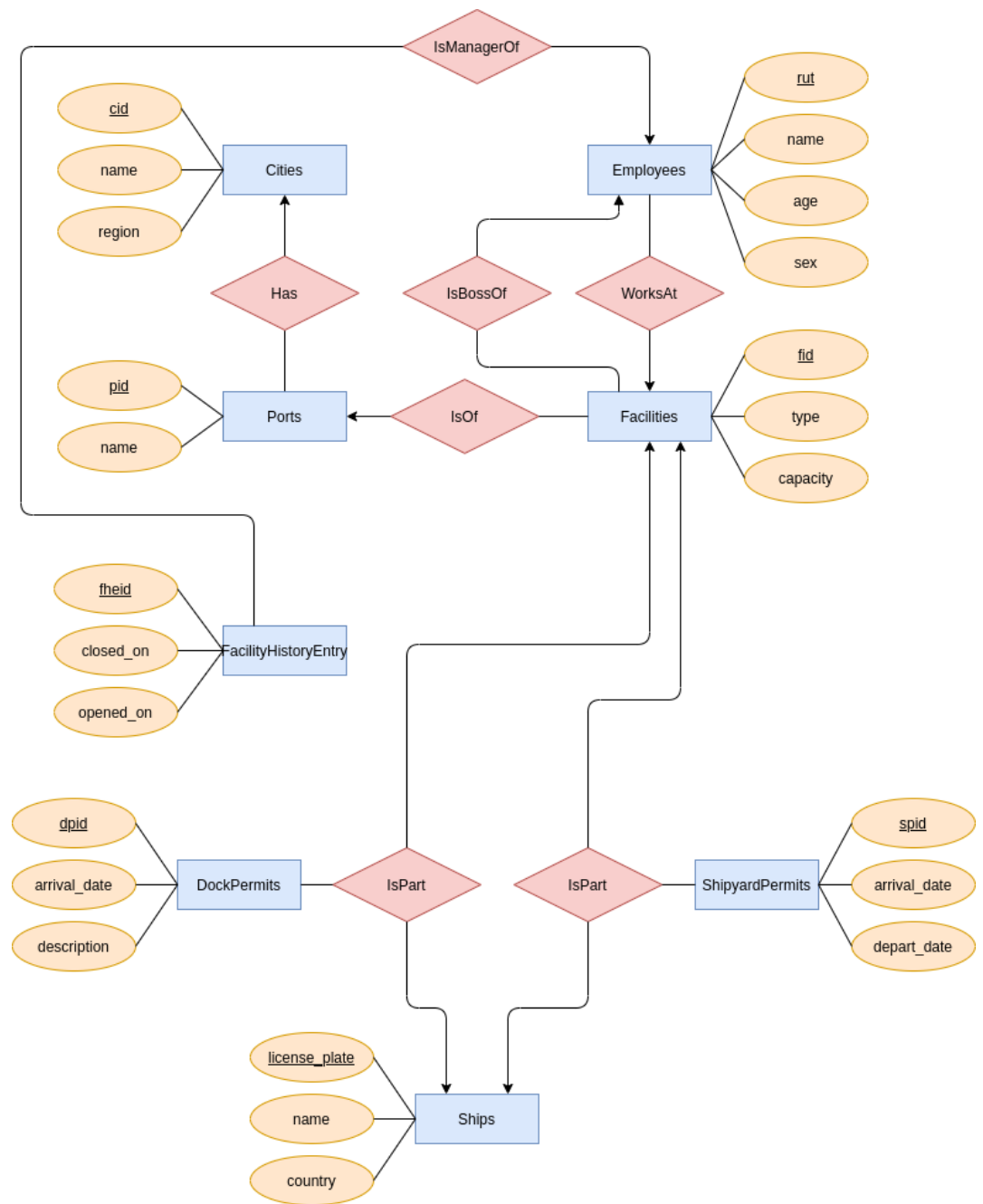


Figura 1: Diagrama E/R

3. Justificación del modelo

El modelo se diseñó para cumplir con las restricciones de 3NF. A continuación se muestran las dependencias funcionales para cada tabla en la base de datos:

- ports (pid, name, cid)
 - $pid \rightarrow name, cid$
- cities (cid, name, region)
 - $cid \rightarrow name, region$
- facilities (fid, type, rut, pid, capacity)
 - $fid \rightarrow type, rut, pid, capacity$
- facility_history_entries (fheid, closed_on, opened_on, close_boss_rut)
 - $fheid \rightarrow closed_on, opened_on, close_boss_rut$
- employees (rut, name, age, sex, fid)
 - $rut \rightarrow name, age, sex, fid$
- ships (license_plate, name, country)
 - $license_plate \rightarrow name, country$
- shipyard_permits (spid, fid, license_plate, arrival_date, departure_date)
 - $spid \rightarrow fid, license_plate, arrival_date, departure_date$
- dock_permits (dpid, fid, license_plate, arrival_date, description)
 - $dpid \rightarrow fid, license_plate, arrival_date, description$

Como se puede ver que el modelo propuesto no presenta anomalías. Se cumple que para cada tabla R_i en el modelo se da que *para toda dependencia funcional no trivial $X \rightarrow Y$, X es llave*. Por lo tanto, el modelo está en BCNF y por consecuencia está también en 3NF.

4. Consultas SQL

A continuación se incluyen las consultas SQL implementadas en la página web:

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

```
SELECT name, cid  
FROM ports ;
```

2. Muestre todos los jefes de las instalaciones del puerto con nombre 'Mejillones'.

```
SELECT facilities.boss_rut  
FROM (SELECT pid FROM ports WHERE UPPER(name) LIKE "%MEJILLONES %")  
AS puertos, facilities WHERE puertos.pid = facilities.pid;
```

3. Muestre todos los puertos que tienen al menos un astillero.

```
SELECT DISTINCT pid  
FROM facilities  
WHERE UPPER(type) LIKE "%SHIPYARD %";
```

4. Muestre todas las veces en que el barco 'Calypso' ha atracado en 'Arica'.

```
SELECT atraques.arrival_date  
FROM (SELECT facilities.fid FROM (SELECT ports.pid FROM cities, ports  
WHERE UPPER(cities.name) LIKE "%ARICA %" AND cities.cid = ports.cid)  
AS puertos, facilities WHERE puertos.pid = facilities.pid) AS instalaciones, (SE-  
LECT arrival_date, fid FROM shipyard_permits, ships WHERE UPPER(ships.name)  
LIKE "%CALYPSO %" UNION SELECT arrival_date, fid FROM dock_per-  
mits, ships WHERE UPPER(ships.name) LIKE "%CALYPSO %") AS atraques  
WHERE atraques.fid = instalaciones.fid;
```

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

```
SELECT puertos.pid, AVG(employees.age)  
FROM (SELECT ports.pid, facilities.fid FROM ports, facilities WHERE ports.pid  
= facilities.pid) AS puertos, employees  
WHERE puertos.fid = employees.fid  
GROUP BY puertos.pid;
```

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

```
SELECT pid  
FROM (SELECT COUNT(*), pid FROM (SELECT f.pid, t.license_plate FROM  
facilities as f, ((SELECT license_plate, fid FROM dock_permits WHERE arri-  
val_date ≥ '2020-08-01' AND arrival_date < '2020-09-01') UNION (SELECT  
license_plate, fid FROM shipyard_permits WHERE arrival_date ≥ '2020-08-01'  
AND arrival_date < '2020-09-01')) as t WHERE f.fid=t.fid) AS t1 GROUP BY  
pid) as t4, (SELECT MAX(count) FROM (SELECT COUNT(*), pid FROM
```

```
(SELECT f.pid, t.license_plate FROM facilities as f, ((SELECT license_plate,
fid FROM dock_permits WHERE arrival_date ≥ '2020-08-01' AND arrival_date < '2020-09-01') UNION (SELECT license_plate, fid FROM shipyard_permits
WHERE arrival_date ≥ '2020-08-01' AND arrival_date < '2020-09-01')) as t WHERE f.fid=t.fid) AS t1 GROUP BY pid) AS t3) AS t5 WHERE t4.count=t5.max;
```

5. Supuestos y consideraciones

El procesamiento de datos fue realizado con Python. A la relación `facility_history_entry` se le eliminó la columna `fid` mediante el comando `DROP`. Esto se hizo debido a que con la columna `close_boss_rut` es posible obtener la instalación en cuestión.