```
CREATE TABLE IF NOT EXISTS ship(
ship_id INT PRIMARY KEY CHECK (ship_id >= 200000000 AND ship_id <= 799999999),
name VARCHAR(64) NOT NULL,
flag VARCHAR(64) NOT NULL,
length NUMERIC NOT NULL CHECK (length > 0),
width NUMERIC NOT NULL CHECK (width > 0)
);

CREATE TABLE IF NOT EXISTS berth(
berth_id INT PRIMARY KEY CHECK (berth_id > 0)
);

CREATE TABLE IF NOT EXISTS container(
container_id VARCHAR(11) PRIMARY KEY
);

CREATE TABLE IF NOT EXISTS yard(
bay INT NOT NULL CHECK (bay > 0),
row INT NOT NULL CHECK (row > 0),
tier INT NOT NULL CHECK (tier > 0),
CONSTRAINT pk_yard PRIMARY KEY (bay, row, tier)
);

CREATE TABLE IF NOT EXISTS transfer(
container VARCHAR(11) NOT NULL,
from_bay INT NOT NULL,
from_row INT NOT NULL,
from_tier INT NOT NULL,
to_bay INT NOT NULL,
to_row INT NOT NULL,
to_tier INT NOT NULL,
start TIMESTAMP NOT NULL,
finish TIMESTAMP NOT NULL,
CONSTRAINT fk_container FOREIGN KEY (container) REFERENCES
container(container_id),
CONSTRAINT fk_from_yard FOREIGN KEY (from_bay, from_row, from_tier) REFERENCES
yard(bay,row,tier),
CONSTRAINT fk_to_yard FOREIGN KEY (to_bay, to_row, to_tier) REFERENCES
yard(bay,row,tier)
);

DO $$ BEGIN
   CREATE TYPE container_movement AS ENUM ('load', 'unload');
EXCEPTION
   WHEN duplicate_object THEN NULL;
END $$;
```

```
CREATE TABLE IF NOT EXISTS load (
bay INT NOT NULL,
row INT NOT NULL,
tier INT NOT NULL,
ship INT NOT NULL,
container VARCHAR(11) NOT NULL,
direction container_movement NOT NULL,
start TIMESTAMP NOT NULL,
finish TIMESTAMP NOT NULL,
CONSTRAINT valid_duration CHECK (start < finish),
CONSTRAINT fk_container FOREIGN KEY (container) REFERENCES
container(container_id),
CONSTRAINT fk_ship FOREIGN KEY (ship) REFERENCES ship(ship_id),
CONSTRAINT fk_yard FOREIGN KEY (bay, row, tier) REFERENCES yard(bay,row,tier)
);

CREATE TABLE IF NOT EXISTS schedule (
ship INT NOT NULL,
berth INT NOT NULL,
arrival TIMESTAMP NOT NULL,
departure TIMESTAMP NOT NULL,
CONSTRAINT valid_schedule CHECK (arrival < departure),
CONSTRAINT fk_ship FOREIGN KEY (ship) REFERENCES ship(ship_id),
CONSTRAINT fk_berth FOREIGN KEY (berth) REFERENCES berth(berth_id)
);
```