Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
Facultad de Ingeniería IIC2413

Entrega 2

Diseño de un aplicación

Iñaki Ballesteros
Catalina Cifuentes
23 de mayo de 2022

Crear Modelo

1. Construir un modelo de Entidad/Relación que represente de manera fiel la situación planteada en la entrega 1, pero ahora respaldados por datos provenientes tanto de las aerolíneas como de la DGAC (recuerda que la aplicación es distinta para los grupos par o impar). Se deben ilustrar las entidades, asociaciones y cardinalidades.

- Compania_aerea(codigo_compania char(3), nombre_compania varchar(30))
    - Primary key: codigo_compania

- Trabajador(trabajador_id serial, pasaporte char(9), nombre varchar(30), fecha_nacimiento date)
    - Primary key: trabajador_id
    - Child: Piloto, Tripulacion

- Piloto(trabajador_id serial, pasaporte char(9), nombre varchar(30), fecha_nacimiento date, licencia_actual_id serial)
    - Inherits: Trabajador

- Tripulacion(trabajador_id serial, pasaporte char(9), nombre varchar(30), fecha_nacimiento date)
    - Inherits: Trabajador

- TrabajaEn(trabajador_id serial, codigo_compania char(3))
    - Primary key: trabajador_id, codigo_compania
    - Foreign key: trabajador_id referencia Trabajador, codigo_compania referencia Compania_aerea

- Aerodromo (aerodromo_id serial, codigo_ICAO char(4), codigo_IATA char(3), nombre_aerodromo varchar(64), latitud float, longitud float, nombre_ciudad varchar(30), nombre_pais varchar(30))
    - Primary key: aerodromo_id

- Aeronave(codigo_aeronave char(7), nombre_aeronave varchar(20), modelo varchar(20), peso float)
    - Primary key: codigo_aeronave

- Vuelo(vuelo_id serial, codigo_vuelo char(7), aerodromo_salida_id serial, aerodromo_llegada_id serial, codigo_compania char(3), fecha_salida timestamp, fecha_llegada timestamp, velocidad float, altitud float, ruta_id serial, codigo_aeronave char(7), estado varchar(10))
    - Primary key: vuelo_id
    - Foreign key: (aerodromo_salida_id, aerodromo_llegada_id) referencia Aerodromo, codigo_aeronave referencia Aeronave, codigo_compania referencia Compania_aerea, ruta_id referencia Ruta

- VuelaEn(vuelo_id serial, trabajador_id serial, rol varchar(30))
    - Primary key: vuelo_id, trabajador_id
    - Foreign key: vuelo_id referencia Vuelo, trabajador_id referencia Trabajador

- Ruta(ruta_id serial, nombre_ruta char(6))
    - Primary key: ruta_id

- Punto(nombre_punto varchar(5), longitud float, latitud float)
    - Primary key: nombre_punto

- Recorrido(ruta_id serial, nombre_punto varchar(5), cardinalidad int)
    - Primary key: ruta_id, cardinalidad
    - Foreign key: ruta_id referencia Ruta, nombre_punto referencia Punto

- CostoTicket(valor int, ruta_id serial, codigo_aeronave char(7))
    - Primary key: ruta_id, codigo_aeronave
    - Foreign key: ruta_id referencia Ruta, codigo_aeronave referencia Aeronave

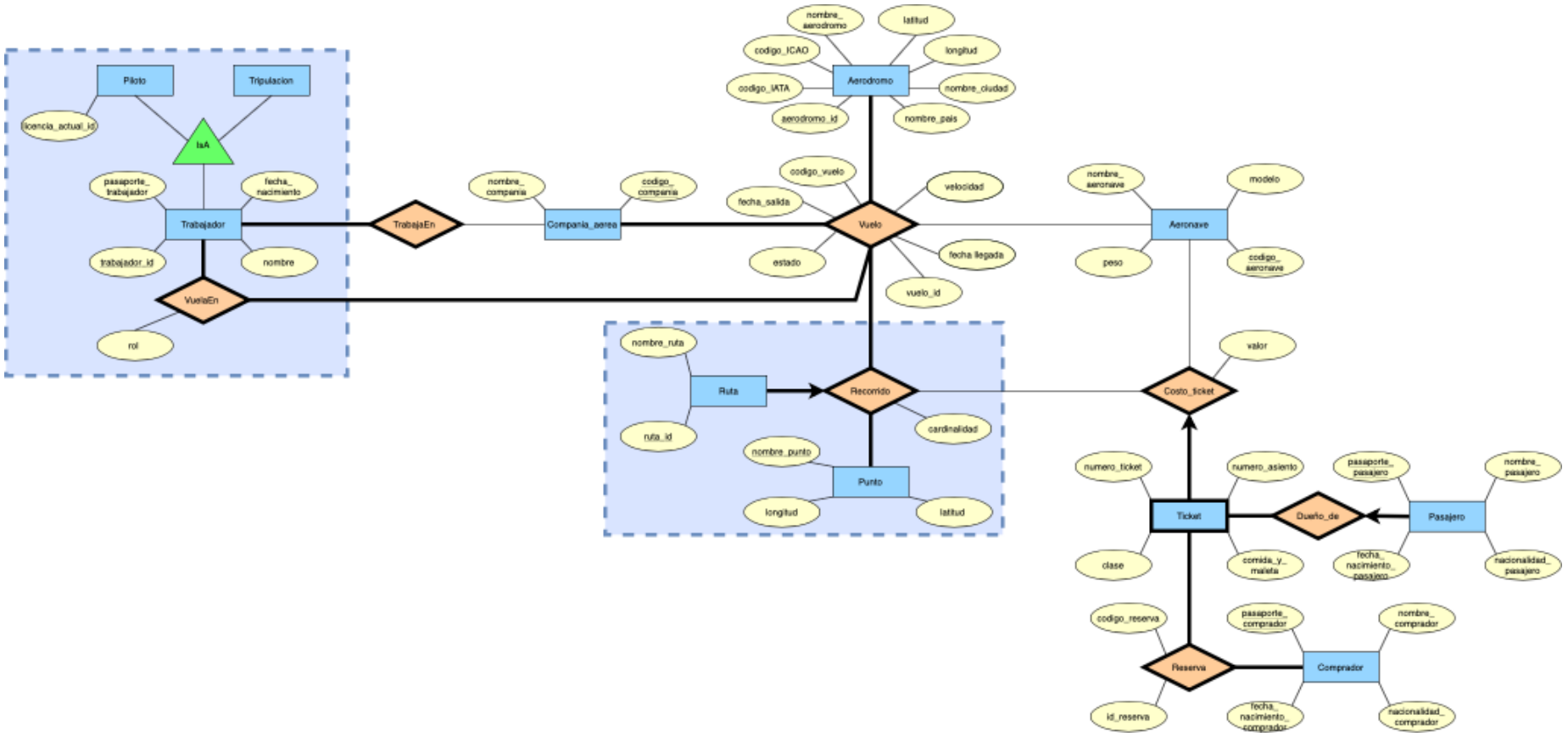
- Ticket(numero_ticket serial, vuelo_id serial, numero_asiento int, clase varchar(20), comida_y_maleta bool, pasaporte_pasajero char(9))
    - Primary key: numero_ticket
    - Foreign key: vuelo_id referencia Vuelo, pasaporte_pasajero referencia Pasajero

- Comprador(pasaporte_comprador char(9), nombre_comprador varchar(30), nacionalidad_comprador varchar(20), fecha_nacimiento_comprador date)
    - Primary key: pasaporte_comprador

- Pasajero(pasaporte_pasajero char(9), nombre_pasajero varchar(30), fecha_nacimiento_pasajero date, nacionalidad_pasajero varchar(20))
    - Primary key: pasaporte_pasajero

- Reserva(reserva_id serial, codigo_reserva char(12), pasaporte_comprador char(9), numero_ticket serial)
    - Primary key: reserva_id, numero_ticket
    - Foreign key: numero_ticket referencia Ticket, pasaporte_comprador referencia Comprador

2. Traspasar el modelo Entidad/Relación a un esquema relacional especificando los da- tatypes correctos e incluyendo llaves primarias y llaves foráneas como restricciones de integridad (no es necesario agregar otro tipo de restricciones adicionales).


Piloto
Tripulacion
licencia_actual_id
IsA
pasaporte_trabajador
fecha_nacimiento
Trabajador
trabajador_id
nombre
VuelaEn
rol
TrabajaEn
nombre_compania
codigo_compania
Compania_aerea
nombre_aerodromo
latitud
codigo_ICAO
longitud
codigo_IATA
Aerodromo
nombre_ciudad
aerodromo_id
nombre_pais
codigo_vuelo
velocidad
fecha_salida
Vuelo
estado
fecha llegada
vuelo_id
nombre_aeronave
modelo
Aeronave
peso
codigo_aeronave
nombre_ruta
Ruta
ruta_id
Recorrido
cardinalidad
nombre_punto
Punto
longitud
latitud
valor
Costo_ticket
numero_ticket
numero_asiento
Ticket
Dueño_de
Pasajero
clase
comida_y_maleta
pasaporte_pasajero
nombre_pasajero
fecha_nacimiento_pasajero
nacionalidad_pasajero
codigo_reserva
pasaporte_comprador
nombre_comprador
Reserva
Comprador
id_reserva
fecha_nacimiento_comprador
nacionalidad_comprador

Justificar Modelo

1. Debes mostrar que tu modelo está en BCNF, o si prefieres en 3NF. Para ello debes listar las dependencias funcionales de cada tabla y mostrar que está todo normalizado.

- Compania_aerea(codigo_compania char(3), nombre varchar(30))

codigo_compania → nombre

- Trabajador(trabajador_id int, pasaporte_trabajador char(9), nombre varchar(30), fecha_nacimiento date)

trabajador_id → pasaporte_trabajador
pasaporte_trabajador → nombre, fecha_nacimiento

- Piloto(trabajador_id serial, pasaporte char(9), nombre varchar(30), licencia_actual_id serial, fecha_nacimiento date)

trabajador_id → pasaporte
pasaporte → nombre, fecha_nacimiento, licencia_actual_id

- Tripulacion(trabajador_id int, pasaporte_trabajador char(9), nombre varchar(30))

trabajador_id → pasaporte_trabajador
pasaporte_trabajador → nombre, fecha nacimiento

- Aerodromo (aerodromo_id serial, codigo_ICAO char(4), codigo_IATA char(3), nombre_aerodromo varchar(64), latitud float, longitud float)

id_aerodromo → codigo_ICAO, codigo_IATA
codigo_ICAO, codigo_IATA → nombre_aerodromo, latitud, longitud

- Aeronave(codigo_aeronave char(7), nombre_aeronave varchar(20), modelo varchar(20), peso float)

código → nombre, modelo, peso

- Vuelo(vuelo_id int, codigo_vuelo char(7), aerodromo_salida_id int, aerodromo_llegada_id int, codigo_compania char(3), fecha_salida timestamptz, fecha_llegada timestamptz, velocidad float, altitud float, ruta_id int, codigo_aeronave char(7), estado varchar(30))

vuelo_id → codigo_vuelo, codigo_compania, código_aeronave, id_aerodromo_salida, id_aerodromo_llegada, ruta_id, altitud
vuelo_id, fecha_salida → fecha_llegada
id_aerodromo_salida, id_aerodromo_llegada, fecha_salida, llegada → velocidad

- VuelaEn(vuelo_id varchar(30), pasaporte_trabajador char(9), rol varchar(30))

pasaporte_trabajador, vuelo_id → rol

- Ruta(ruta_id serial, nombre_ruta char(6))

ruta_id → nombre_ruta

- Punto(nombre_punto varchar(5), longitud float, latitud float)

nombre_punto → longitud, latitud

- Recorrido(ruta_id serial, nombre_punto varchar(5), cardinalidad int)

ruta_id, cardinalidad → nombre_punto

- CostoTicket(valor int, ruta_id serial, codigo_aeronave char(7))

ruta_id, codigo_aeronave → costo

- Ticket(numero ticket int, vuelo_id serial, numero_asiento varchar(30), clase varchar(30), comida_y_maleta bool, pasaporte_pasajero char(9))

numero ticket → pasaporte_pasajero, codigo_vuelo, numero_asiento, clase, comida_y_maleta

- Comprador(pasaporte_comprador char(9), nombre_comprador varchar(30), nacionalidad_comprador varchar(20), fecha_nacimiento_comprador date)

pasaporte_comprador -> nombre_comprador, fecha_nacimiento_comprador, nacionalidad_comprador

- Pasajero(pasaporte_pasajero char(9), nombre_pasajero varchar(30), fecha_nacimiento_pasajero date, nacionalidad_pasajero varchar(20))

pasaporte_pasajero → nombre, fecha_nacimiento, nacionalidad.

- Reserva(reserva_id serial, codigo_reserva char(12), pasaporte_comprador char(9), numero_ticket serial)

reserva_id, numero_ticket → codigo_reserva, pasaporte_comprador

Consultas en SQL

1. Muestre todos los vuelos pendientes de ser aprobados por la DGAC

```
SELECT vuelo_id, codigo_vuelo
FROM Vuelo
WHERE estado = 'pendiente';
```

2. Dado un código ICAO de un aeródromo ingresado por el usuario y una aerolínea seleccionada por el usuario, liste todos los vuelos aceptados de dicha aerolínea que tienen como destino el aeródromo.

```
SELECT vuelo_id, codigo_vuelo, vuelo.estado
FROM vuelo
WHERE aerodromo_llegada_id IN (SELECT aerodromo_id
                               FROM aerodromo
                               WHERE codigo_icao = 'CODIGO')
AND codigo_compania IN (SELECT codigo_compania
                        FROM compania_aerea
                        WHERE nombre_compania = 'AEROLINEA')
AND vuelo.estado = 'aceptado';
```

3. Dado un código de reserva ingresado por el usuario, liste los tickets asociados a esta junto a sus pasajeros y costos.

```
SELECT ticket.numero_ticket, ticket.pasaporte_pasajero,
costo_ticket.valor
FROM ticket, costo_ticket, vuelo, reserva
WHERE ticket.numero_ticket  = reserva.numero_ticket AND vuelo.ruta_id
= costo_ticket.ruta_id AND vuelo.codigo_aeronave =
costo_ticket.codigo_aeronave AND ticket.vuelo_id = vuelo.vuelo_id AND
reserva.codigo_reserva = 'CODIGO RESERVA';
```

4. Por cada aerolínea, muestre al cliente que ha comprado la mayor cantidad de tickets

```
SELECT MIN(tabla.pasaporte_comprador), tabla.nombre_compania,
MAX(tabla.maximo_ticket) as maximo_ticket
FROM (SELECT reserva.pasaporte_comprador,
compania_aerea.nombre_compania, COUNT(reserva.numero_ticket) as
maximo_ticket
     FROM reserva, ticket, vuelo, compania_aerea
```

```
     WHERE reserva.numero_ticket = ticket.numero_ticket AND
vuelo.vuelo_id = ticket.vuelo_id AND compania_aerea.codigo_compania =
vuelo.codigo_compania
     GROUP BY reserva.pasaporte_comprador,
compania_aerea.nombre_compania)as tabla
GROUP BY tabla.nombre_compania;
```

5. Al ingresar el nombre de una aerolínea, liste la cantidad de vuelos que tienen en cada uno de los estados

```
SELECT compania_aerea.nombre,vuelo.estado, COUNT(vuelo.estado)as
numero
FROM compania_aerea, vuelo
WHERE compania_aerea.codigo_compania = vuelo.codigo_compania AND
compania_aerea.nombre = 'NOMBRE AEROLINEA'
GROUP BY compania_aerea.nombre, vuelo.estado;
```

6. Muestre la aerolínea que tiene el mayor porcentaje de vuelos aceptados

```
SELECT compania_aerea.nombre_compania, (COUNT(vuelo.estado):: float /
MAX(tabla_2.vuelos_totales) :: float *100) as porcentaje
FROM vuelo, compania_aerea, (SELECT compania_aerea.nombre_compania,
COUNT(vuelo.estado) as vuelos_totales
                          FROM vuelo, compania_aerea
                          WHERE vuelo.codigo_compania =
compania_aerea.codigo_compania
                          GROUP BY compania_aerea.nombre_compania)
as tabla_2
WHERE vuelo.codigo_compania = compania_aerea.codigo_compania AND
vuelo.estado = 'aceptado' AND compania_aerea.nombre_compania =
tabla_2.nombre_compania
GROUP BY  compania_aerea.nombre_compania
ORDER BY DESC porcentaje
LIMIT 1;
```