



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
IIC2413 - Bases de datos
Primer semestre 2022

Proyecto Entrega 2 Bases de Datos

PROFESOR
Matias Toro

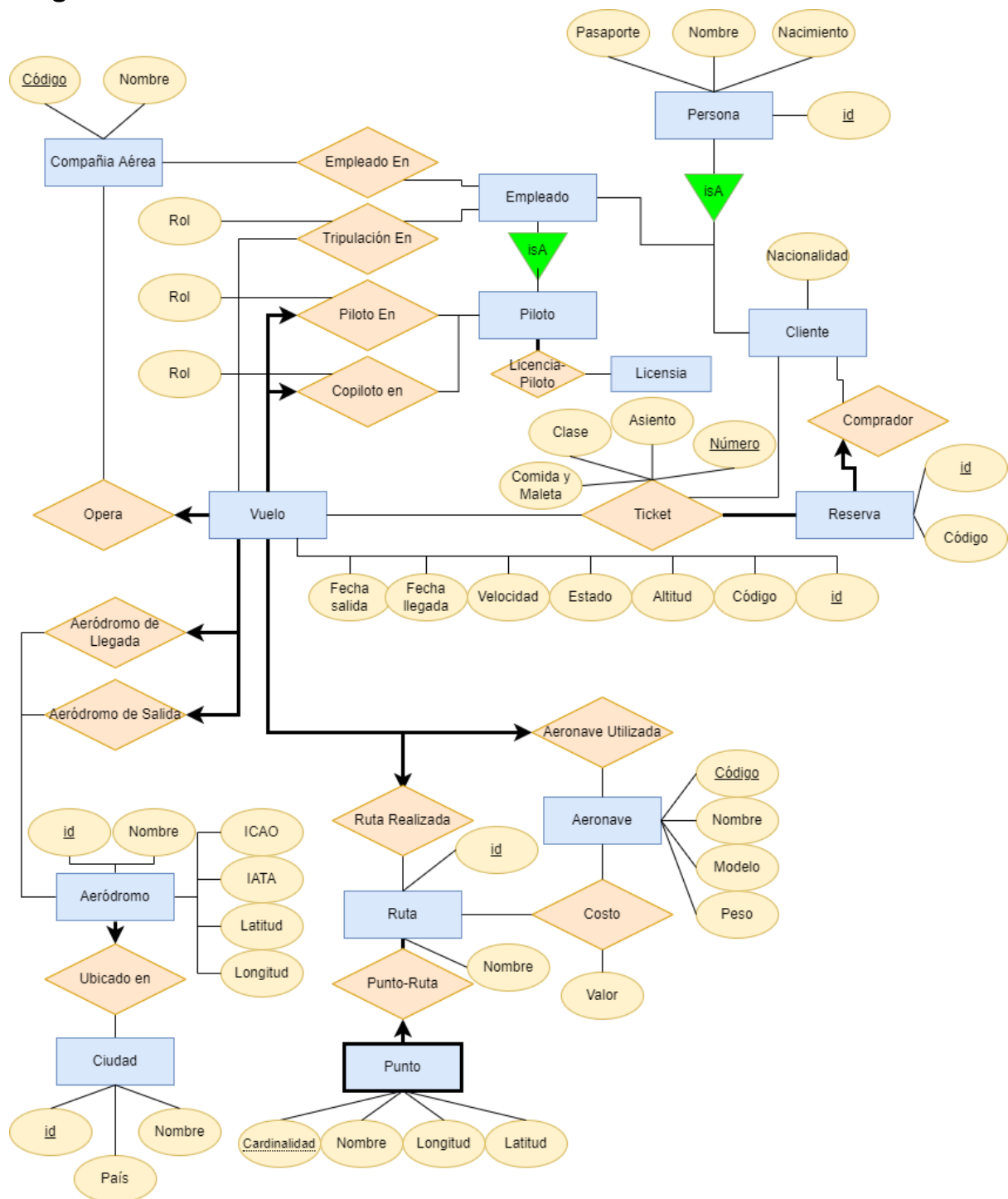
Integrantes:

Jaime Pérez Lisboa; Sección 3

Bryan Fry; Sección 3

Lunes 23 de Abril 2022

Diagrama Entidad-Relación



Esquema Relacional

1. Persona(id:serial, Pasaporte:varchar, Nombre:varchar, Nacimiento: date)
2. Cliente(Persona.id: integer, Nacionalidad: varchar)
3. Compañía Aérea(Código:char(3), Nombre:varchar)
4. Ticket(Numero: serial, Vuelo.id: integer, Reserva.id: integer, Cliente.id: integer, Clase:varchar, Numero_asiento:varchar, Comida_y_maleta: boolean)
5. Reserva(id: serial, Cliente.id: integer, codigo: varchar)
6. Tripulación En(Empleado.id: integer, Vuelo.id: integer, Rol: varchar)
7. Empleado En(Empleado.id: integer, Compañía_Aérea.Código: char(3))
8. Piloto En(Piloto.id: integer, Vuelo.id: integer, Rol: varchar)
9. Copiloto En(Piloto.id: integer, Vuelo.id: integer, Rol: varchar)
10. Licencia_Piloto(Piloto.id: integer, Licencia.id: integer)
11. Vuelo(id: serial, Aerodromo_salida.id: integer, Aerodromo_llegada.id: integer, Ruta.id: integer, Aeronave.codigo: char(7), Compañía_aerea.codigo: char(3), Código:varchar, Fecha_llegada: timestamp, Fecha_salida: timestamp, Velocidad: real, Altitud: real, Estado:varchar)
12. Aeródromo(id: serial, Ciudad.id:integer, Nombre: varchar, ICAO:char(4), IATA: char(3), Latitud: real, Longitud: real)
13. Ciudad(id: serial, Nombre: varchar, País: varchar)
14. Ruta(id: serial, Nombre: varchar)
15. Punto(Ruta.id:integer, Cardinalidad: integer, Nombre: varchar, Latitud:real, Longitud:real)
16. Aeronave(Código: char(7), Nombre:varchar, Modelo:varchar, Peso:real)
17. Costo(Ruta.id:integer, Aeronave.peso: real, Valor: real)

Justificación del modelo

La justificación del modelo consiste en que las relaciones se encuentran en BCNF, lo que se muestra a continuación mediante las entidades del diagrama Entidad - Relación mostrado anteriormente.

1. Persona: id -> Pasaporte, Nombre, Nacimiento
2. Cliente: Persona.id -> Nacionalidad
3. Compañía Aérea: Código -> Nombre
4. Ticket: Numero -> Vuelo.id: integer, Reserva.id, Cliente.id, Clase, Numero_asiento, Comida_y_maleta
5. Reserva: id -> Cliente.id, codigo
6. Tripulación En: Empleado.id, Vuelo.id -> Rol
7. Empleado En: Sin dependencias no-triviales.
8. Piloto En: Piloto.id, Vuelo.id -> Rol
9. Copiloto En: Piloto.id, Vuelo.id -> Rol
10. Licencia_Piloto: Sin dependencias no-triviales.
11. Vuelo: id -> Aerodromo_salida.id, Aerodromo_llegada.id, Ruta.id, Aeronave.codigo, Compañía_aerea.codigo, Código, Fecha_llegada, Fecha_salida, Velocidad, Altitud, Estado

- 12. Aeródromo: id -> Ciudad.id, Nombre, ICAO, IATA, Latitud, Longitud
- 13. Ciudad: id -> Nombre, País
- 14. Ruta: id -> Nombre
- 15. Punto: Ruta.id, Cardinalidad -> Nombre, Latitud, Longitud
- 16. Aeronave: Código -> Nombre, Modelo, Peso
- 17. Costo: Ruta.id, Aeronave.peso -> Valor

Consultas SQL de la página web

Nota: los parámetros están debidamente preparados para evitar inyecciones SQL.

1. SELECT codigo, compania_codigo, aeronave_codigo,
 fecha_salida, fecha_llegada
 FROM vuelo
 WHERE estado = 'pendiente';
2. SELECT compania_aerea.codigo, compania_aerea.nombre,
 vuelo.codigo, vuelo.estado, aerodromo.nombre, aerodromo.icao
 FROM vuelo
 LEFT JOIN aerodromo ON aerodromo.id = vuelo.aerodromo_llegada_id
 LEFT JOIN compania_aerea ON compania_aerea.codigo =
 vuelo.compania_codigo
 WHERE estado = 'aceptado' AND
 compania_aerea.nombre ILIKE:aerolinea AND aerodromo.icao
 ILIKE:icao;
3. SELECT reserva.id,reserva.codigo, comprador.nombre,
 pasajero.nombre, vuelo.codigo, ticket.clase, ticket.numero_asiento,
 costo.valor
 FROM reserva
 LEFT JOIN ticket ON reserva.id = ticket.reserva_id
 LEFT JOIN vuelo ON ticket.vuelo_id = vuelo.id
 LEFT JOIN aeronave ON vuelo.aeronave_codigo = aeronave.codigo
 LEFT JOIN costo ON aeronave.peso = costo.peso AND vuelo.ruta_id =
 costo.ruta_id
 LEFT JOIN persona AS pasajero ON ticket.persona_id = pasajero.id
 LEFT JOIN persona AS comprador ON reserva.cliente_id =
 comprador.id
 WHERE reserva.codigo ILIKE :codigo
 ORDER BY reserva.id;
4. WITH aceptado as (
 SELECT compania_aerea.codigo, count(estado)
 FROM compania_aerea

```

LEFT JOIN vuelo ON compania_aerea.codigo =
vuelo.compania_codigo
WHERE estado = 'aceptado'
GROUP BY compania_aerea.codigo, compania_aerea.nombre, estado
), rechazado as (
SELECT compania_aerea.codigo, count(estado)
FROM compania_aerea
LEFT JOIN vuelo ON compania_aerea.codigo =
vuelo.compania_codigo
WHERE estado = 'rechazado'
GROUP BY compania_aerea.codigo, compania_aerea.nombre, estado
), borrador as (
SELECT compania_aerea.codigo, count(estado)
FROM compania_aerea
LEFT JOIN vuelo ON compania_aerea.codigo =
vuelo.compania_codigo
WHERE estado = 'borrador'
GROUP BY compania_aerea.codigo, estado
), pendiente as (
SELECT compania_aerea.codigo, count(estado)
FROM compania_aerea
LEFT JOIN vuelo ON compania_aerea.codigo =
vuelo.compania_codigo
WHERE estado = 'pendiente'
GROUP BY compania_aerea.codigo, compania_aerea.nombre, estado
), publicado as (
SELECT compania_aerea.codigo, count(estado)
FROM compania_aerea
LEFT JOIN vuelo ON compania_aerea.codigo =
vuelo.compania_codigo
WHERE estado = 'publicado'
GROUP BY compania_aerea.codigo, compania_aerea.nombre, estado
), total as (
SELECT compania_codigo, COUNT(vuelo.id)
FROM vuelo
GROUP BY compania_codigo)
SELECT compania_aerea.codigo, compania_aerea.nombre as
compania_nombre,
coalesce(aceptado.count,0) as aceptado,
coalesce(aceptado.count,0)/total.count::float as aceptado_percent,

```

```

coalesce(rechazado.count,0) as rechazado,
coalesce(rechazado.count,0)/total.count::float as rechazado_percent,
coalesce(borrador.count,0) as borrador,
coalesce(borrador.count,0)/total.count::float as borrador_percent,
coalesce(pendiente.count,0) as pendiente,
coalesce(pendiente.count,0)/total.count::float as pendiente_percent,
coalesce(publicado.count,0) as publicado,
coalesce(publicado.count,0)/total.count::float as publicado_percent,
total.count as total
FROM compania_aerea
LEFT JOIN aceptado ON compania_aerea.codigo = aceptado.codigo
LEFT JOIN rechazado ON compania_aerea.codigo = rechazado.codigo
LEFT JOIN borrador ON compania_aerea.codigo = borrador.codigo
LEFT JOIN pendiente ON compania_aerea.codigo = pendiente.codigo
LEFT JOIN publicado ON compania_aerea.codigo = publicado.codigo
LEFT JOIN total ON compania_aerea.codigo = total.compania_codigo
WHERE compania_aerea.nombre ILIKE :compania
ORDER BY compania_aerea.codigo;

```

```

5.    WITH aceptado as (
SELECT compania_aerea.codigo, count(estado)
FROM compania_aerea
LEFT JOIN vuelo ON compania_aerea.codigo =
vuelo.compania_codigo
WHERE estado = 'aceptado'
GROUP BY compania_aerea.codigo, compania_aerea.nombre, estado
), rechazado as (
SELECT compania_aerea.codigo, count(estado)
FROM compania_aerea
LEFT JOIN vuelo ON compania_aerea.codigo =
vuelo.compania_codigo
WHERE estado = 'rechazado'
GROUP BY compania_aerea.codigo, compania_aerea.nombre, estado
), borrador as (
SELECT compania_aerea.codigo, count(estado)
FROM compania_aerea
LEFT JOIN vuelo ON compania_aerea.codigo =
vuelo.compania_codigo
WHERE estado = 'borrador'
GROUP BY compania_aerea.codigo, estado
), pendiente as (

```

```

SELECT compania_aerea.codigo, count(estado)
FROM compania_aerea
LEFT JOIN vuelo ON compania_aerea.codigo =
vuelo.compania_codigo
WHERE estado = 'pendiente'
GROUP BY compania_aerea.codigo, compania_aerea.nombre, estado
), publicado as (
SELECT compania_aerea.codigo, count(estado)
FROM compania_aerea
LEFT JOIN vuelo ON compania_aerea.codigo =
vuelo.compania_codigo
WHERE estado = 'publicado'
GROUP BY compania_aerea.codigo, compania_aerea.nombre, estado
), total as (
SELECT compania_codigo, COUNT(vuelo.id)
FROM vuelo
GROUP BY compania_codigo)
SELECT compania_aerea.codigo, compania_aerea.nombre as
compania_nombre,
coalesce(aceptado.count,0) as aceptado,
coalesce(aceptado.count,0)/total.count::float as aceptado_percent,
coalesce(rechazado.count,0) as rechazado,
coalesce(rechazado.count,0)/total.count::float as rechazado_percent,
coalesce(borrador.count,0) as borrador,
coalesce(borrador.count,0)/total.count::float as borrador_percent,
coalesce(pendiente.count,0) as pendiente,
coalesce(pendiente.count,0)/total.count::float as pendiente_percent,
coalesce(publicado.count,0) as publicado,
coalesce(publicado.count,0)/total.count::float as publicado_percent,
total.count as total
FROM compania_aerea
LEFT JOIN aceptado ON compania_aerea.codigo = aceptado.codigo
LEFT JOIN rechazado ON compania_aerea.codigo = rechazado.codigo
LEFT JOIN borrador ON compania_aerea.codigo = borrador.codigo
LEFT JOIN pendiente ON compania_aerea.codigo = pendiente.codigo
LEFT JOIN publicado ON compania_aerea.codigo = publicado.codigo
LEFT JOIN total ON compania_aerea.codigo = total.compania_codigo
WHERE compania_aerea.nombre ILIKE :compania
ORDER BY compania_aerea.codigo;
6. WITH total as (

```

```

SELECT compania_codigo, COUNT(vuelo.id)
FROM vuelo
GROUP BY compania_codigo
), vuelos_count as (
SELECT compania_aerea.codigo, estado, count(estado),
count(estado)/total.count::float as percent
FROM compania_aerea
LEFT JOIN vuelo ON compania_aerea.codigo =
vuelo.compania_codigo
JOIN total ON total.compania_codigo = compania_aerea.codigo
WHERE estado = :estado
GROUP BY compania_aerea.codigo, estado, total.count)
SELECT compania_aerea.codigo, compania_aerea.nombre, :estado,
coalesce(vuelos_count.count,0), coalesce(vuelos_count.percent,0),
total.count as total
FROM vuelos_count
FULL JOIN total ON vuelos_count.codigo = total.compania_codigo
LEFT JOIN compania_aerea ON compania_aerea.codigo =
total.compania_codigo
WHERE coalesce(vuelos_count.$metrica,0)= (
SELECT $orden(coalesce(vuelos_count.$metrica,0)) as extreme
FROM vuelos_count
FULL JOIN total ON vuelos_count.codigo = total.compania_codigo);

```

Supuestos

- Para ejecutar y cargar los datos, se tiene que estar dentro del repositorio git (~/Proyecto-IIC2413-23) y ejecutar el comando:
`psql grupo23e2 < schema.sql`