PONTIFICIA UNIVERSIDAD CATÓLICA

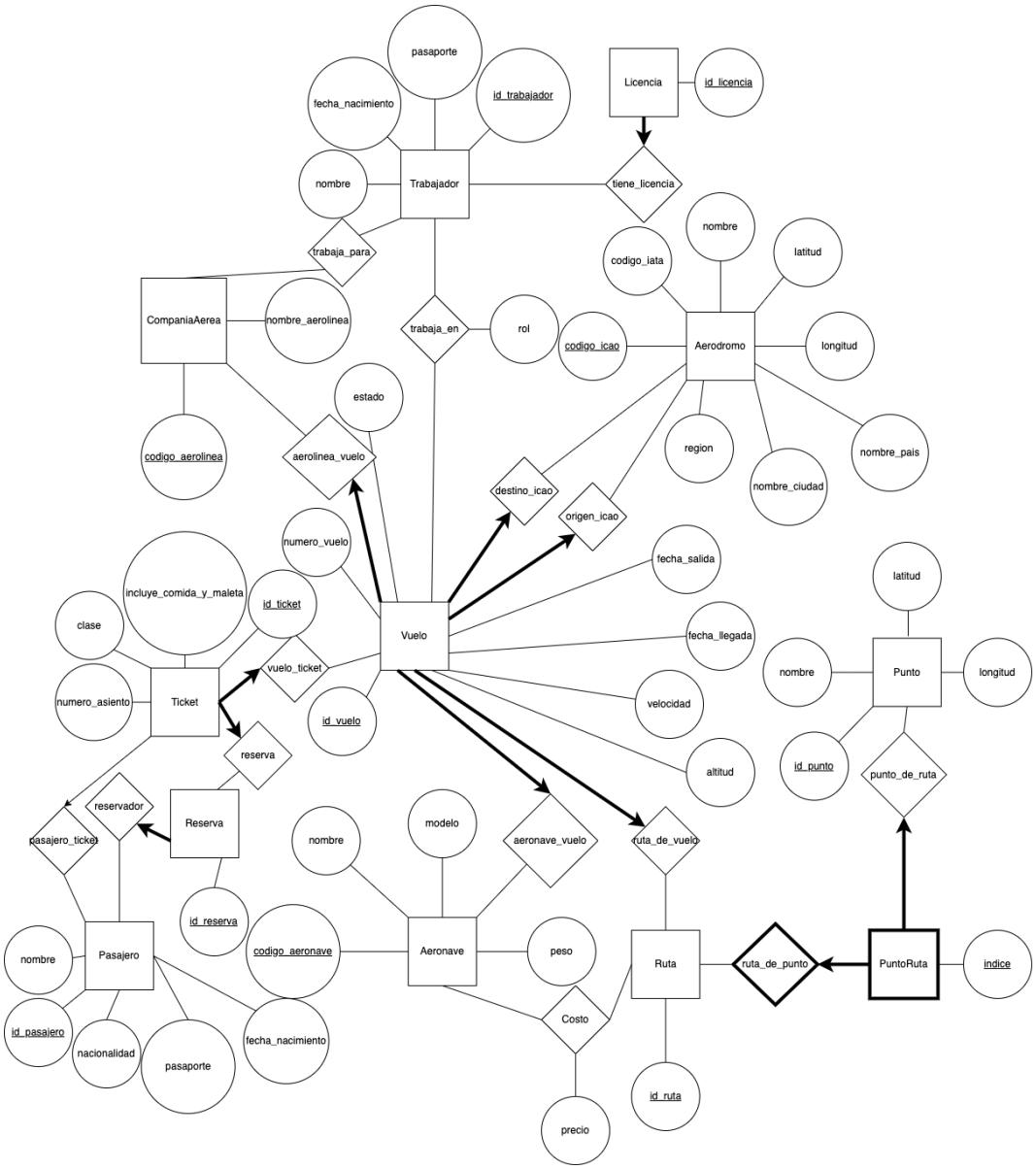
IIC2413

Entrega 2

Nombres: Martín Andrighetti, Lucas Fernández

Fecha entrega: 2022-05-23

1. Diagrama E/R



2. Modelo relacional

- Aerodromo

  ($\underline{codigo\_icao}$ **char**(**4**), $codigo\_iata$ **char**(**3**), $nombre$ **varchar**(**255**), $latitud$ **float**, $longitud$ **float**, $nombre\_ciudad$ **varchar**(**255**), $nombre\_pais$ **varchar**(**255**))

- Aeronave

  ($\underline{codigo\_aeronave}$ **char**(**7**), $nombre$ **varchar**(**255**), $modelo$ **varchar**(**255**), $peso$ **float**)

- CompaniaAerea

  ($\underline{codigo\_aerolinea}$ **char**(**3**), $nombre\_aerolinea$ **varchar**(**255**))

- Costo

  ($\underline{id\_ruta}$ **int**, $\underline{codigo\_aeronave}$ **char**(**7**), $precio$ **float**)

  id_ruta es llave foránea de $Ruta.id\_ruta$

  $codigo\_aeronave$ es llave foránea de $Aeronave.codigo\_aeronave$

- Licencia

  ($\underline{id\_licencia}$ **int**, $id\_trabajador$ **int**)

  $id\_trabajador$ es llave foránea de $Trabajador.id\_trabajador$

- Pasajero

  ($\underline{id\_pasajero}$ **int**, $pasaporte$ **varchar**(**15**), $nombre$ **varchar**(**255**),

  $fecha\_nacimiento$ **date**, $nacionalidad$ **varchar**(**255**))

- Punto

  ($\underline{id\_punto}$ **int**, $nombre$ **varchar**(**255**), $latitud$ **float**, $longitud$ **float**)

- PuntoRuta

  ($\underline{id\_ruta}$ **int**, $\underline{indice}$ **int**, id_punto **int**)

  Restricción: Para cada ruta existe una colección secuencial de índices, comenzando desde cero.

  id_ruta es llave foránea de $Ruta.id\_ruta$

  id_punto es llave foránea de $Punto.id\_punto$

- Reserva

  ($\underline{id\_reserva}$ **int**, $id\_reservador$ **int**)

  $id\_reservador$ es llave foránea de $Pasajero.id\_pasajero$

- Ruta

  ($\underline{id\_ruta}$ **int**)

- Trabajador

  ($\underline{id\_trabajador}$ **int**, $pasaporte$ **varchar**(**15**), $nombre$ **varchar**(**255**),

  $fecha\_de\_nacimiento$ **date**)

- TrabajaEn

  ($\underline{id\_vuelo}$ **int**, $\underline{id\_trabajador}$ **int**, rol **varchar**(**255**))

  id_vuelo es llave foránea de $Vuelo.id\_vuelo$

  $id\_trabajador$ es llave foránea de $Trabajador.id\_trabajador$

- TrabajaPara

  ($\underline{id\_trabajador}$ **int**, $\underline{codigo\_aerolinea}$ **char**(**3**))

  $id\_trabajador$ es llave foránea de $Trabajador.id\_trabajador$

  $codigo\_aerolinea$ es llave foránea de $CompaniaAerea.codigo\_aerolinea$

- Ticket

  ($\underline{id\_ticket}$ **int**, $id\_reserva$ **int**, $id\_pasajero$ **int**, id_vuelo **int**, $numero\_asiento$ **int**,

  $clase$ **varchar**(**31**), $incluye\_comida\_y\_maleta$ **boolean**)

  $id\_pasajero$ puede tomar valores nulos.

  $id\_reserva$ es llave foránea de $Reserva.id\_reserva$

  $id\_pasajero$ es llave foránea de $Pasajero.id\_pasajero$

  id_vuelo es llave foránea de $Vuelo.id\_vuelo$

- Vuelo

  ($\underline{id\_vuelo}$ **int**, $numero\_vuelo$ **varchar**(**31**), $origen\_icao$ **char**(**4**), $destino\_icao$ **char**(**4**),

  $codigo\_aerolinea$ **char**(**3**), $fecha\_salida$ **timestamp**, $fecha\_llegada$ **timestamp**, $velocidad$ **float**,

  $altitud$ **float**, id_ruta **int**, $codigo\_aeronave$ **char**(**7**), $estado$ **varchar**(**31**))

  $origen\_icao$ es llave foránea de $Aerodromo.codigo\_icao$

  $destino\_icao$ es llave foránea de $Aerodromo.codigo\_icao$

  $codigo\_aerolinea$ es llave foránea de $CompaniaAerea.codigo\_aerolinea$

  id_ruta es llave foránea de $Ruta.id\_ruta$

  $codigo\_aeronave$ es llave foránea de $Aeronave.codigo\_aeronave$

2.1. Consideraciones

1. A menos que se especifique lo contrario, no se aceptan NULLs.

2.2. Supuestos

1. El número de pasaporte consiste en hasta 9 letras y números, según lo acordado internacionalmente.

2. Las coordenadas geográficas se representan como un par de latitud y longitud, cada uno representado como un número real. Para la latitud, un número positivo indica coordenada este. Para la longitud, un número positivo indica coordenada norte.

3. El precio se especifica como un número flotante de dólares. Es coincidencia que los datos contengan únicamente números enteros de dólares.

4. No se incluye la dependencia del país a partir de la latitud y longitud. Esto se realiza por simplicidad y porque hay puntos del planeta en que la soberanía es ambigua o incluso polémica.

2.3. Justificación de cumplimiento con BCNF

- Aerodromo

  $codigo\_icao \rightarrow < todos >$

  $codigo\_iata \rightarrow < todos >$

  $codigo\_icao$ y $codigo\_iata$ son llaves candidatas. Dado que el lado izquierdo de todas las dependencias no triviales son llaves candidatas se cumple BCNF.

- Aeronave

  $codigo\_aeronave \rightarrow < todos >$

  Dado que el lado izquierdo de la única dependencia no trivial es llave, se cumple BCNF.

- CompaniaAerea

  $codigo\_aerolinea \rightarrow nombre\_aerolinea$

  Dado que la parte izquierda de la dependencia no trivial es una llave, se cumple BCNF.

- Costo

  $id\_ruta, codigo\_aeronave \rightarrow precio$

  Dado que el lado izquierdo de la única dependencia es llave, se cumple BCNF.

- Licencia

  $id\_licencia \rightarrow id\_trabajador$

  Dado que el lado izquierdo de la única dependencia funcional es llave, se cumple BCNF.

- Pasajero

  $id\_pasajero \rightarrow < todos >$

  $pasaporte \rightarrow < todos >$

  Dado que los lados izquierdos de ambas dependencias no triviales son llaves candidatas, se cumple BCNF.

- Punto

  $id\_punto \rightarrow < todos >$

  Asumiendo que no hay puntos duplicados: $latitud, longitud \rightarrow < todos >$

  Dado que el lado izquierdo de ambas dependencias no triviales son llaves candidatas, se cumple BCNF.

- PuntoRuta

  $id\_ruta, indice \rightarrow id\_punto$

  Asumiendo que las rutas no pasan dos veces por el mismo punto: $id\_ruta, id\_punto \rightarrow indice$

  $id\_ruta, indice$ es llave, y id_ruta, id_punto es llave candidata. Como estas son las partes izquierdas de las dependencias no triviales, se cumple BCNF.

- Reserva

  $id\_reserva \rightarrow id\_reservador$

  Dado que el lado izquierdo de la única dependencia no trivial, $id\_reserva$, es llave candidata, se cumple BCNF.

- Ruta

  No hay atributos suficientes para formar una dependencia, por lo que se cumple BCNF.

- Trabajador

  $id\_trabajador \rightarrow < todos >$

  $pasaporte \rightarrow < todos >$

  Las llaves candidatas son $id\_trabajador$ y $pasaporte$. Las dependencias funcionales no triviales tienen en el lado izquierdo llaves candidatas, por lo que se cumple BCNF.

- TrabajaEn

  $id\_vuelo, id\_trabajador \rightarrow rol$

  Dado que el lado izquierdo de la única dependencia funcional no trivial es llave, se cumple BCNF.

- TrabajaPara

  No hay dependencias funcionales no triviales, por lo que se cumple BCNF.

- Ticket

  $id\_ticket \rightarrow < todos >$

  Asumiendo que un pasajero no puede comprar 2 asientos para sí: $id\_pasajero, id\_vuelo \rightarrow < todos >$

  Asumiendo que no se puede comprar dos veces el mismo asiento: $id\_vuelo, numero\_asiento \rightarrow < todos >$

  Todas las dependencias no triviales definen llaves candidatas, por lo que se cumple BCNF.

- Vuelo

  $id\_vuelo \rightarrow < todos >$

  Dado que el lado izquierdo de la única dependencia funcional no trivial es llave, se cumple BCNF.

3. Consultas SQL

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

```
SELECT Vuelo.numero_vuelo, Origen.nombre as origen, Destino.nombre as destino,
    Vuelo.fecha_salida, Vuelo.fecha_llegada, Vuelo.estado
FROM Vuelo, Aerodromo as Origen, Aerodromo as Destino
WHERE estado = 'pendiente'
    AND Vuelo.origen_icao = Origen.codigo_icao
    AND Vuelo.destino_icao = Destino.codigo_icao;
```

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.

   Se entrega el nombre del origen y destino para ser más user-friendly, también se entrega el código ICAO ya que al funci con matching parcial.

   Tomando $\$codigo$ y $\$aerolinea\_escogida$ como lo entregado por el usuario:

```
SELECT Vuelo.numero_vuelo, Origen.codigo_icao,
    Origen.nombre as origen, Destino.codigo_icao, Destino.nombre as destino,
    Vuelo.fecha_salida, Vuelo.fecha_llegada, Vuelo.estado
FROM Vuelo, CompaniaAerea, Aerodromo as Origen, Aerodromo as Destino
WHERE UPPER(CompaniaAerea.nombre_aerolinea) LIKE '%$aerolinea_escogida%'
    AND CompaniaAerea.codigo_aerolinea = Vuelo.codigo_aerolinea
    AND UPPER(Vuelo.destino_icao) LIKE '%$codigo%'
    AND Vuelo.estado = 'aceptado'
    AND Vuelo.origen_icao = Origen.codigo_icao
    AND Vuelo.destino_icao = Destino.codigo_icao;
```

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

   Tomando $\$codigo\_reserva$ como lo entregado por el usuario y asumiendo que este codigo tiene matching completo debido a la naturaleza numérica de la variable:

```
SELECT Ticket.id_ticket, Vuelo.numero_vuelo, Origen.nombre as origen,
    Destino.nombre as destino, Vuelo.fecha_salida, Vuelo.fecha_llegada,
    Vuelo.estado, Ticket.numero_asiento, Ticket.clase,
    Ticket.incluye_comida_y_maleta, Pasajero.pasaporte,
    Pasajero.nombre, Costo.precio
FROM Ticket, Pasajero, Vuelo, Costo, Aerodromo as Origen, Aerodromo as Destino
WHERE Ticket.id_reserva = $codigo_reserva
    AND ticket.id_pasajero = Pasajero.id_pasajero
    AND Vuelo.id_vuelo = Ticket.id_vuelo
    AND Vuelo.id_ruta = Costo.id_ruta
    AND Vuelo.codigo_aeronave = Costo.codigo_aeronave
    AND Vuelo.origen_icao = Origen.codigo_icao
    AND Vuelo.destino_icao = Destino.codigo_icao;
```

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

```
SELECT CompaniaAerea.nombre_aerolinea, Pasajero.pasaporte,
    Pasajero.nombre, Cantidades.cantidad
FROM CompaniaAerea, Pasajero, (
    SELECT Cantidades.codigo_aerolinea,
        MAX(Cantidades.cantidad) as max_cantidad
    FROM (
        SELECT Vuelo.codigo_aerolinea,
            Reserva.id_reservador as id_cliente, COUNT(*) as cantidad
        FROM Ticket, Vuelo, Reserva
        WHERE Ticket.id_reserva = Reserva.id_reserva
            AND Ticket.id_vuelo = Vuelo.id_vuelo
        GROUP BY Vuelo.codigo_aerolinea, Reserva.id_reservador
    ) as Cantidades
    GROUP BY Cantidades.codigo_aerolinea
) as MaxCantidades, (
    SELECT Vuelo.codigo_aerolinea,
        Reserva.id_reservador as id_cliente, COUNT(*) as cantidad
    FROM Ticket, Vuelo, Reserva
    WHERE Ticket.id_reserva = Reserva.id_reserva
        AND Ticket.id_vuelo = Vuelo.id_vuelo
    GROUP BY Vuelo.codigo_aerolinea, Reserva.id_reservador
) as Cantidades
WHERE Cantidades.codigo_aerolinea = MaxCantidades.codigo_aerolinea
    AND Cantidades.cantidad = MaxCantidades.max_cantidad
    AND Cantidades.codigo_aerolinea = CompaniaAerea.codigo_aerolinea
    AND Cantidades.id_cliente = Pasajero.id_pasajero
ORDER BY CompaniaAerea.nombre_aerolinea, Pasajero.nombre;
```

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

   Tomando $\$nombre\_escogido$ como lo entregado por el usuario:

```
SELECT CompaniaAerea.nombre_aerolinea, Vuelo.estado, COUNT(Vuelo.id_vuelo) as cantid
FROM Vuelo, CompaniaAerea
WHERE UPPER(CompaniaAerea.nombre_aerolinea) LIKE '%$nombre_escogido%'
    AND CompaniaAerea.codigo_aerolinea = Vuelo.codigo_aerolinea
GROUP BY CompaniaAerea.nombre_aerolinea, Vuelo.estado;
```

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

```
SELECT PorcentajesAprobados.nombre_aerolinea, PorcentajesAprobados.porcentaje
FROM (
    SELECT porcentaje
    FROM (
        SELECT CompaniaAerea.nombre_aerolinea,
            100 * VuelosAprobados.cantidad_aprobada/VuelosTotales.cantidad
            as porcentaje
        FROM CompaniaAerea, (
            SELECT codigo_aerolinea, count(id_vuelo) as cantidad_aprobada
            FROM Vuelo
            WHERE estado = 'aceptado'
            GROUP BY codigo_aerolinea
        ) as VuelosAprobados, (
            SELECT codigo_aerolinea, count(id_vuelo) as cantidad
            FROM Vuelo
            GROUP BY codigo_aerolinea
        ) as VuelosTotales
        WHERE CompaniaAerea.codigo_aerolinea = VuelosTotales.codigo_aerolinea
            AND CompaniaAerea.codigo_aerolinea
            = VuelosAprobados.codigo_aerolinea
            AND VuelosTotales.cantidad > 0
    ) as PorcentajesAprobados
    ORDER BY porcentaje DESC
    LIMIT 1
) as PorcentajeMayor, (
    SELECT CompaniaAerea.nombre_aerolinea,
        100 * VuelosAprobados.cantidad_aprobada/VuelosTotales.cantidad
```

```
        as porcentaje
    FROM CompaniaAerea, (
        SELECT codigo_aerolinea, count(id_vuelo) as cantidad_aprobada
        FROM Vuelo
        WHERE estado = 'aceptado'
        GROUP BY codigo_aerolinea
    ) as VuelosAprobados, (
        SELECT codigo_aerolinea, count(id_vuelo) as cantidad
        FROM Vuelo
        GROUP BY codigo_aerolinea
    ) as VuelosTotales
    WHERE CompaniaAerea.codigo_aerolinea = VuelosTotales.codigo_aerolinea
        AND CompaniaAerea.codigo_aerolinea
        = VuelosAprobados.codigo_aerolinea
        AND VuelosTotales.cantidad > 0
) as PorcentajesAprobados
WHERE PorcentajesAprobados.porcentaje = PorcentajeMayor.porcentaje;
```

4. Página Web

Con el siguiente link se puede ingresar a la página web que implementa las consultas:

https://codd.ing.puc.cl/~grupo19/index.php?