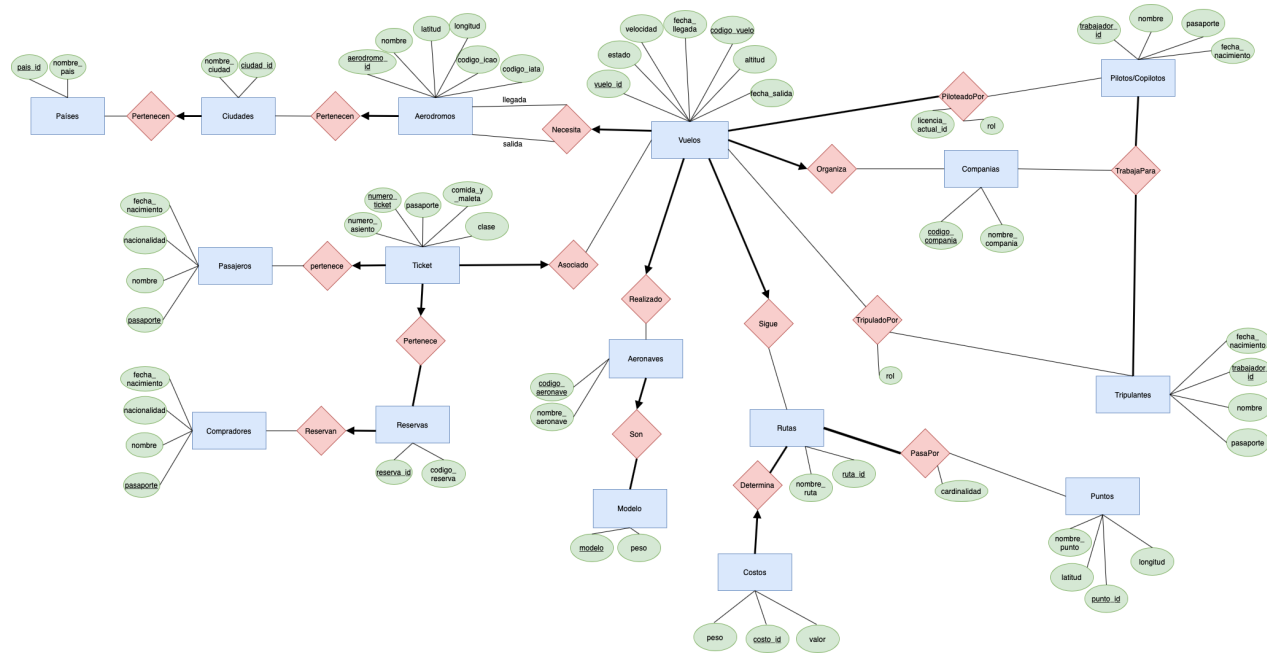


Reporte Entrega 2

Aníbal Contreras - Roberto Orrego

IIC2413

1 Diagrama E/R



2 Esquema Relacional

Respecto al esquema relacional, las *primary key* están subrayadas y son el primer atributo de cada una de las relaciones y las *foreign key* se explican implícitamente por el nombre de cada uno de los atributos. Ejemplo: En la realación Aeródromos, la llave primaria es aerodromo_id que está subrayado y la llave foránea es ciudad_id, pues tiene el nombre de la relacion del nombre Ciudades

- Países(pais_id:string, nombre_pais:string)
- Ciudades(ciudad_id:integer, nombre_ciudad:string, pais_id:integer)
- Aerodromos(aerodromo_id:integer, nombre:string, codigo_icao:string, codigo_iata:string, ciudad_id:integer, latitud:float, longitud:float)
- Compradores(pasaporte_comprador:string, nombre_comprador: string, nacionalidad_comprador: string, fecha_nacimiento_comprador:date)
- Reservas(reserva_id:integer, codigo_reserva:string, pasaporte_comprador:string)
- Pasajeros(pasaporte_pasajero:string, nombre_pasajero:string, nacionalidad_pasajero:string, fecha_nacimiento_pasajero:date)
- Tickets(reserva_id:integer, numero_ticket:integer, numero_asiento:integer, clase:string, comida_y_maleta:boolean, pasaporte_pasajero:string)
- Tripulantes(trabajador_id:integer, nombre:string, fecha_nacimiento:date, pasaporte:string)
- Pilotos(trabajador_id:integer, nombre:string, fecha_nacimiento:date, pasaporte:string)
- Companias(codigo_compania:string, nombre_compania:string)
- Puntos(punto_id:integer, nombre_punto: string latitud:float, longitud:float)
- Rutas(ruta_id:integer, nombre_ruta:string)
- Aeronaves(codigo_aeronave:string, nombre_aeronave:string, modelo:string)
- Modelos(modelo:string, peso:float)
- Costos(costo_id:integer, ruta_id:integer, peso:float, valor:integer)
- Vuelos(vuelo_id:integer,codigo_vuelo:string, aerodromo_salida_id:integer, aerodromo_llegada_id:integer, fecha_salida:date, fecha_llegada:date, estado:string, velocidad:float, altitud:float, codigo_aeronave:redstring, codigo_compania:string, ruta_id:integer)
- Tripulado_por(vuelo_id:integer, trabajador_id:integer, rol:string)
- Piloteado_por(vuelo_id:integer, trabajador_id:integer, rol:string, licencia_actual_id:integer)
- Pasa_por(ruta_id:integer, punto_id:integer, cardinalidad:integer)

3 Justificación

A continuación se presentan las dependencias funcionales no triviales de nuestro modelo.

- Países(pais_id:string, nombre_pais:string)
 $pais_id \rightarrow nombre_pais$
- Ciudades(ciudad_id:integer, nombre_ciudad:string, pais_id:integer)
 $ciudad_id \rightarrow nombre_ciudad, pais_id$
- Aerodromos(aerodromo_id:integer, nombre:string, codigo_icao:string, codigo_iata:string, ciudad_id:integer, latitud:float, longitud:float)
 $aerodromo_id \rightarrow nombre, codigo_icao, codigo_iata, ciudad_id, latitud, longitud$
- Compradores(pasaporte_comprador:string, nombre_comprador:string, nacionalidad_comprador:string, fecha_nacimiento_comprador:date)
 $pasaporte_comprador \rightarrow nombre_comprador, nacionalidad_comprador, fecha_nacimiento_comprador$
- Reservas(reserva_id:integer, codigo_reserva:string, pasaporte_comprador:string, vuelo_id:integer)
 $reserva_id \rightarrow codigo_reserva, pasaporte_comprador, vuelo_id$
- Pasajeros(pasaporte_pasajero:string, nombre_pasajero:string, nacionalidad_pasajero:string, fecha_nacimiento_pasajero:date)
 $pasaporte_pasajero \rightarrow nombre_pasajero, nacionalidad_pasajero, fecha_nacimiento_pasajero$
- Tickets(reserva_id:integer, numero_ticket:integer, numero_asiento:integer, clase:string, comida_y_maleta:boolean, pasaporte_pasajero:string)
 $reserva_id \rightarrow numero_ticket, numero_asiento, clase, comida_y_maleta, pasaporte_pasajero$
- Tripulantes(trabajador_id:integer, nombre:string, fecha_nacimiento:date, pasaporte:string)
 $trabajador_id \rightarrow nombre, fecha_nacimiento, pasaporte$
- Pilotos(trabajador_id:integer, nombre:string, fecha_nacimiento:date, pasaporte:string)
 $trabajador_id \rightarrow nombre, fecha_nacimiento, pasaporte$
- Companias(codigo_compania:string(3), nombre_compania:string)
 $codigo_compania \rightarrow nombre_compania$
- Puntos(punto_id:integer, nombre_punto:string, latitud:float, longitud:float)
 $nombre_punto \rightarrow latitud, longitud$
- Rutas(ruta_id:integer, nombre_ruta:string(6))
 $ruta_id \rightarrow nombre_ruta$
- Aeronaves(codigo_aeronave:string(7), nombre_aeronave:string, modelo:string)
 $codigo_aeronave \rightarrow nombre_aeronave, modelo$

- Modelos(modelo:string(7), peso:float)
modelo → *peso*
- Costos(costo_id:integer, ruta_id:integer, peso:float, valor:integer)
costo_id → *ruta_id, peso, valor*
- Vuelos(vuelo_id:integer, codigo_vuelo:string(7), aerodromo_salida_id:integer, aerodromo_llegada_id:integer, fecha_salida:date, fecha_llegada:date, estado:string, velocidad:float, altitud:float)
vuelo_id → *codigo_vuelo, aerodromo_salida, aerodromo_llegada_id, fecha_salida, fecha_llegada, estado, altitud*
- Tripulado_por(vuelo_id:integer, trabajador_id:integer, rol:string)
vuelo_id, trabajador_id → *rol*
- Piloteado_por(vuelo_id:integer, trabajador_id:integer, rol:string, licencia_actual_id:integer)
vuelo_id, trabajador_id → *rol, licencia_actual_id*

Recordemos que las tablas entregadas en un inicio no estaban normalizadas. A través de la descomposición de estas se logró crear el modelo recién propuesto. Mediante "joins" de estas tablas se pueden obtener nuevamente las tablas originales sin ninguna pérdida de información.

Finalmente podemos concluir que este modelo se encuentra en 3NF. Todos los atributos que no son llave dependen directamente de la llave primaria de la tabla a la cual pertenecen. Listando todas las dependencias funcionales se puede notar que este modelo se encuentra normalizado.

4 Consultas SQL

1.

```
1 SELECT codigo_vuelo, estado
2 FROM vuelos
3 WHERE estado = 'pendiente'
```
2.

```
1 SELECT vuelos.vuelo_id, aerodromo_salida_id, aerodromo_llegada_id, estado, nombre_compania
2 FROM aerodromos, grupo105e2.public.companias, grupo105e2.public.vuelos
3 WHERE aerodromos.aerodromo_id = vuelos.aerodromo_llegada_id
4 AND vuelos.codigo_compania = companias.codigo_compania
5 AND codigo_ICAO LIKE '%$icao%'
6 AND nombre_compania LIKE '%$var%'
7 AND vuelos.estado LIKE 'aceptado'
```
3.

```
1 SELECT numero_ticket, pasaporte_pasajero, costos.valor, codigo_reserva
2 FROM reservas, grupo105e2.public.tickets, vuelos, grupo105e2.public.aeronaves, grupo105e2.public.rutas, costos
3 WHERE tickets.reserva_id = reservas.reserva_id
4 AND tickets.vuelo_id = vuelos.vuelo_id
5 AND vuelos.codigo_aeronave = aeronaves.codigo_aeronave
6 AND vuelos.ruta_id = rutas.ruta_id
7 AND costos.ruta_id = rutas.ruta_id
8 AND costos.peso = aeronaves.peso
9 AND codigo_reserva LIKE '%$codigo_reserva%'
```
4.

```
1 SELECT maximosCompradores.nombre_compania, nombre_comprador, cantidadTickets
2 FROM (SELECT DISTINCT nombre_compania, compradores.nombre_comprador, count(numero_ticket) as cantidadTickets
3 FROM grupo105e2.public.companias, grupo105e2.public.vuelos, grupo105e2.public.tickets, reservas, compradores
4 WHERE companias.codigo_compania = vuelos.codigo_compania
5 AND vuelos.vuelo_id = tickets.vuelo_id
6 AND tickets.reserva_id = reservas.reserva_id
7 AND reservas.pasaporte_comprador = compradores.pasaporte_comprador
8 GROUP BY nombre_compania, compradores.nombre_comprador
9 ORDER BY nombre_compania) as TicketsCompradores, (SELECT DISTINCT nombre_compania, max(cantidadTickets)
10 FROM (SELECT DISTINCT nombre_compania, compradores.nombre_comprador, count(numero_ticket) as cantidadTickets
11 FROM grupo105e2.public.companias, grupo105e2.public.vuelos, grupo105e2.public.tickets, reservas, compradores
12 WHERE companias.codigo_compania = vuelos.codigo_compania
13 AND vuelos.vuelo_id = tickets.vuelo_id
14 AND tickets.reserva_id = reservas.reserva_id
15 AND reservas.pasaporte_comprador = compradores.pasaporte_comprador
16 GROUP BY nombre_compania, compradores.nombre_comprador
17 ORDER BY nombre_compania) as compradoresTickets
18 GROUP BY nombre_compania) as maximosCompradores
19 WHERE maximosCompradores.nombre_compania = TicketsCompradores.nombre_compania
20 AND TicketsCompradores.cantidadTickets = maximosCompradores.max
```
5.

```
1 SELECT DISTINCT estado, COUNT(estado) as cantidadVuelos
2 FROM grupo105e2.public.vuelos, grupo105e2.public.companias
3 WHERE vuelos.codigo_compania = companias.codigo_compania
4 AND nombre_compania LIKE '%$aerolinea%'
5 GROUP BY estado
```
6.

```
1 SELECT codigo_compania, nombre_compania, porcentaje
2 FROM (SELECT max(porcentaje) as maximo_porcentaje
3 FROM (SELECT tablaEstadosTotales.codigo_compania, tablaEstadosTotales.nombre_compania, (estado_aceptado/estadosTotales) as porcentaje
4 FROM (SELECT estadosAceptados.codigo_compania, estadosAceptados.nombre_compania, sum(estadosAceptados.estado_aceptado + estadosNoAceptados.estado_no_aceptado)
5 as estadosTotales
6 FROM (SELECT vuelos.codigo_compania, companias.nombre_compania, count(nombre_compania) as estado_aceptado
7 FROM grupo105e2.public.vuelos, grupo105e2.public.companias
8 WHERE vuelos.codigo_compania = companias.codigo_compania
9 AND estado = 'aceptado'
10 GROUP BY vuelos.codigo_compania, companias.nombre_compania) as estadosAceptados, (
11 SELECT vuelos.codigo_compania, companias.nombre_compania, count(nombre_compania) as estado_no_aceptado
12 FROM grupo105e2.public.vuelos, grupo105e2.public.companias
13 WHERE vuelos.codigo_compania = companias.codigo_compania
14 AND estado <> 'aceptado'
15 GROUP BY vuelos.codigo_compania, companias.nombre_compania) as estadosNoAceptados
16 WHERE estadosAceptados.codigo_compania = estadosNoAceptados.codigo_compania
17 GROUP BY estadosAceptados.codigo_compania, estadosAceptados.nombre_compania) as tablaEstadosTotales,
18 (SELECT vuelos.codigo_compania, companias.nombre_compania, count(nombre_compania) as estado_aceptado
19 FROM grupo105e2.public.vuelos, grupo105e2.public.companias
20 WHERE vuelos.codigo_compania = companias.codigo_compania
21 AND estado = 'aceptado'
22 GROUP BY vuelos.codigo_compania, companias.nombre_compania) as tablaEstadosAceptados
23 WHERE tablaEstadosTotales.codigo_compania = tablaEstadosAceptados.codigo_compania
24 AND tablaEstadosAceptados.nombre_compania = tablaEstadosTotales.nombre_compania) as tabla_porcentajes) as maximoValor, (
```

```

35 SELECT tablaEstadosTotales.codigo_compania, tablaEstadosTotales.nombre_compania, (estado_aceptado/estadosTotales) as porcentaje
36 FROM (SELECT estadosAceptados.codigo_compania, estadosAceptados.nombre_compania, sum(estadosAceptados.estado_aceptado + estadosNoAceptados.estado_no_aceptado)
37      as estadosTotales
38 FROM (SELECT vuelos.codigo_compania, companias.nombre_compania, count(nombre_compania) as estado_aceptado
39 FROM grupo105e2.public.vuelos, grupo105e2.public.companias
40 WHERE vuelos.codigo_compania = companias.codigo_compania
41 AND estado = 'aceptado'
42 GROUP BY vuelos.codigo_compania, companias.nombre_compania) as estadosAceptados, (
43 SELECT vuelos.codigo_compania, companias.nombre_compania, count(nombre_compania) as estado_no_aceptado
44 FROM grupo105e2.public.vuelos, grupo105e2.public.companias
45 WHERE vuelos.codigo_compania = companias.codigo_compania
46 AND estado <> 'aceptado'
47 GROUP BY vuelos.codigo_compania, companias.nombre_compania) as estadosNoAceptados
48 WHERE estadosAceptados.codigo_compania = estadosNoAceptados.codigo_compania
49 GROUP BY estadosAceptados.codigo_compania, estadosAceptados.nombre_compania) as tablaEstadosTotales,
50 (SELECT vuelos.codigo_compania, companias.nombre_compania, count(nombre_compania) as estado_aceptado
51 FROM grupo105e2.public.vuelos, grupo105e2.public.companias
52 WHERE vuelos.codigo_compania = companias.codigo_compania
53 AND estado = 'aceptado'
54 GROUP BY vuelos.codigo_compania, companias.nombre_compania) as tablaEstadosAceptados
55 WHERE tablaEstadosTotales.codigo_compania = tablaEstadosAceptados.codigo_compania
56 AND tablaEstadosAceptados.nombre_compania = tablaEstadosTotales.nombre_compania
57 ) as maximosValores
58 WHERE maximosValores.porcentaje = maximo_porcentaje

```

5 README

Las tablas se encuentran creadas en la base de datos grupo105e2. Las tablas que comienzan con "tabla_..." son las tablas que fueron entregadas en formato csv, con el *datatype* correspondiente. Luego de eso se crearon las tablas segun se muestra en el esquema relacional.

Para ingresar a la pagina web y realizar las consultas se debe ir al siguiente link:

- <https://codd.ing.puc.cl/~grupo105/index.php?>