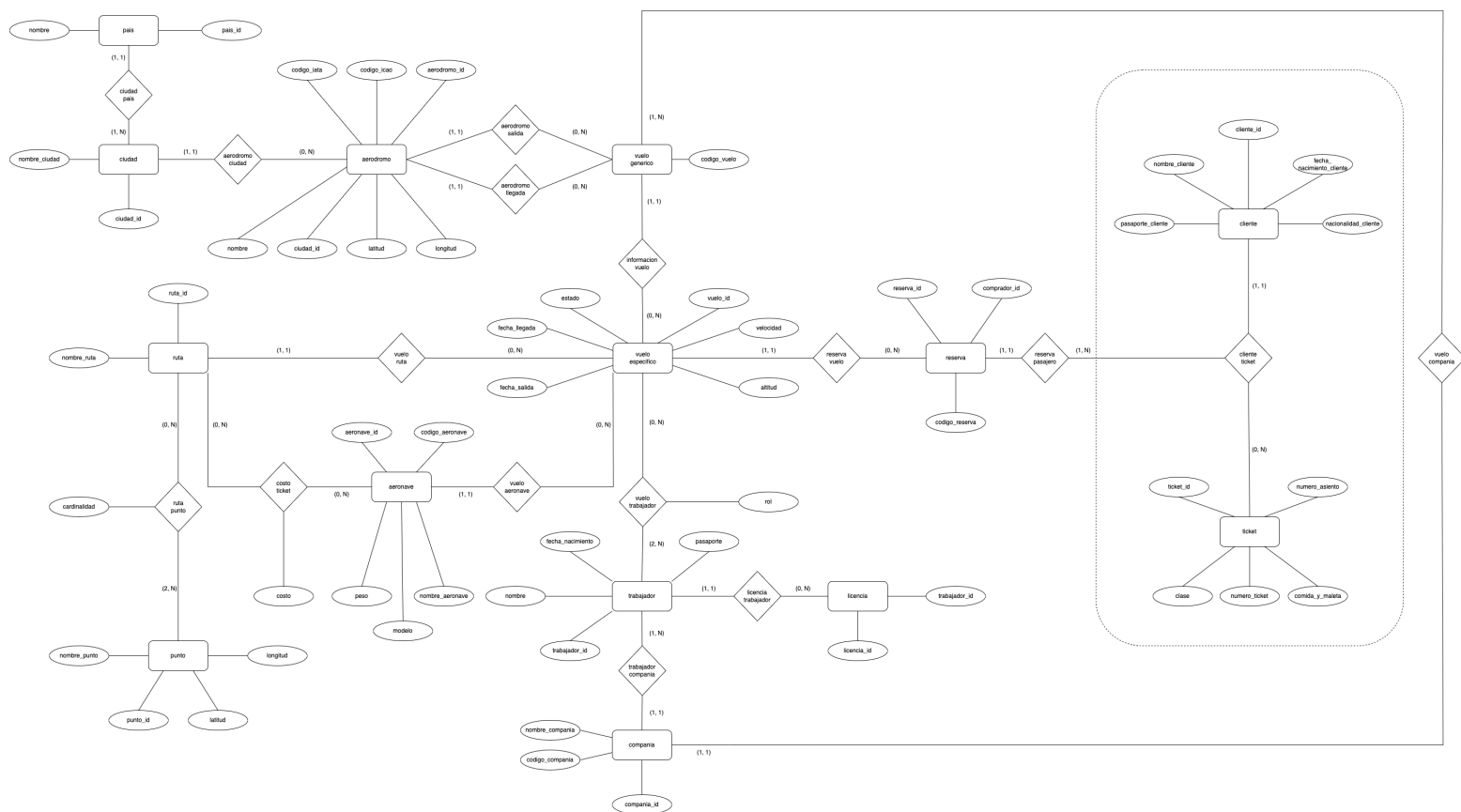




## Entrega 2— Diseño de una aplicación

### Modelo Relacional

### Modelo Entidad/Relación



## Esquema relacional

- aerodromo(aerodromo\_id: int, nombre: varchar(60), codigo\_icao: char(4), codigo\_iata: char(3), latitud: float, longitud: float, ciudad.ciudad\_id: int)
- ciudad(ciudad\_id: int, nombre\_ciudad: varchar(20), pais.pais\_id: int )
- pais(pais\_id: int, nombre\_pais: varchar(20))
- reserva(reserva\_id: int, codigo\_reserva: varchar(15), vueloespecifico.vuelo\_id: int, cliente.cliente\_id: int)
- cliente(cliente\_id: int, pasaporte\_cliente: char(9), nombre\_cliente: varchar(25), nacionalidad\_cliente: varchar(20), fecha\_nacimiento\_cliente: date)
- reservapasajero(reserva\_id: int, cliente.cliente\_id: int, ticket.ticket\_id: int)
- ticket(ticket\_id: int, numero\_ticket: int, numero\_asiento: int, clase: varchar(15), comida\_y\_maleta: varchar(10) )
- ruta(ruta\_id: int, nombre\_ruta: char(6) )
- rutapunto(ruta.ruta\_id: int, punto.punto\_id: int, cardinalidad: int)
- punto(punto\_id: int, nombre\_punto: varchar(8), latitud: float, longitud: float)
- trabajador(trabajador\_id: int, nombre: varchar(25), fecha\_nacimiento: date, pasaporte: char(9))
- trabajadorcompania(trabajador.trabajador\_id: int, compania.compania\_id: int)
- licencia(licencia\_actual\_id: int, trabajador.trabajador\_id: int)
- vuelotrabajador(trabajador.trabajador\_id: int, vuelo.vuelo\_id: int, rol: varchar(20))
- vueloespecifico(vuelo\_id: int, ruta.ruta\_id: int, vuelogenerico.codigo\_vuelo: char(7), aeronave.aeronave\_id: int, fecha\_salida: date, fecha\_llegada: date, velocidad: float, altitud: float, estado: varchar(10))
- compania(compania\_id: int, nombre\_compania: varchar(30), codigo\_compania: char(3))
- vuelogenerico(codigo\_vuelo: char(7), aerodromo\_salida\_id: int, aerodromo\_llegada\_id: int, compania.compania\_id: int)
- aeronave(aeronave\_id: int, codigo\_aeronave: char(7), nombre\_aeronave: varchar(20), modelo: varchar(15), peso: float)
- costoticket(ruta.ruta\_id: int, aeronave.aeronave\_id: float, costo: int)

## Justificación modelo

\* Durante toda la justificación se debe tomar en cuenta que un modelo esta en 3NF si para toda dependencia funcional no trivial  $X \rightarrow Y$ ,  $X$  es una superllave o  $Y$  es una llave minimal.  $Y$  es una llave minimal si no existe llave  $Z'$  tal que  $Z' \subset Z$ .  $X$  es una superllave si todas las relaciones están asignadas a  $X$  en el esquema relacional.

### Aeródromos

Se tiene la entidad inicial dada:

**aerodromos**(aerodromo\_id, nombre, codigo\_icao, codigo\_iata, latitud, longitud, nombre\_ciudad, nombre\_pais)

Esta es normalizada, obteniendo las siguientes entidades:

- **aerodromo**(aerodromo\_id, nombre, codigo\_icao, codigo\_iata, latitud, longitud, ciudad\_id)
- **ciudad**(ciudad\_id, nombre\_ciudad, pais\_id)
- **pais**(pais\_id, nombre\_pais)

Dentro de las entidades creadas existen las siguientes dependencias:

1. pais: {pais\_id}  $\rightarrow$  {nombre\_pais}
2. ciudad: {ciudad\_id}  $\rightarrow$  {nombre\_ciudad, pais\_id}
3. aerodromo: {aerodromo\_id}  $\rightarrow$  {nombre, codigo\_icao, codigo\_iata, latitud, longitud, ciudad\_id}

Ahora, con las entidades ya normalizadas se justifica que cada una se encuentra en 3NF.

**países:** pais\_id es la superllave, esta se creo al normalizar los datos y de este se puede determinar su nombre.

**ciudades:** ciudad\_id es la superllave, creada al normalizar los datos y de el id de la ciudad se puede determinar su nombre y el país al cual pertenece.

**aerodromos:** El aerodromo\_id es la superllave, esta es obtenida de la entidad inicial aerodromos. Del aerodromo\_id se puede obtener el nombre, el codigo IATA e ICAO, latitud, longitud y el ciudad\_id. El nombre del aerodromo, los códigos y la tupla longitud, latitud son atributos propios del aerodromo\_id y no se repiten.

### Reservas

Se tiene la entidad inicial dada:

**reservas**(reserva\_id, codigo\_reserva, numero\_ticket, vuelo\_id, pasaporte\_comprador, nombre\_comprador, nacionalidad\_comprador, fecha\_nacimiento\_comprador, numero\_asiento, clase, comida\_y\_maleta, pasaporte\_pasajero, nombre\_pasajero, nacionalidad\_pasajero, fecha\_nacimiento\_pasajero)

La cual se normaliza obteniendo las siguientes entidades:

- **reserva**(reserva\_id, codigo\_reserva, vuelo\_id, cliente\_id)
- **cliente**(cliente\_id, pasaporte\_cliente, nombre\_cliente, nacionalidad\_cliente, fecha\_nacimiento\_cliente)
- **reservapasajero**(reserva\_id, cliente\_id, ticket\_id)
- **ticket**(ticket\_id, numero\_ticket, numero\_asiento, clase, comida\_y\_maleta)

Dentro de las entidades existen las siguientes dependencias.

1. reserva: {reserva\_id}  $\rightarrow$  {codigo\_reserva, vuelo\_id, cliente\_id}
2. cliente: {cliente\_id}  $\rightarrow$  {pasaporte\_cliente, nacionalidad\_cliente, fecha\_nacimiento\_cliente}
3. reservapasajero: {reserva\_id, cliente\_id}  $\rightarrow$  {ticket\_id}
4. ticket: {ticket\_id}  $\rightarrow$  {numero\_ticket, numero\_asiento, clase, comida\_y\_maleta}

Ahora, con las entidades ya normalizadas se justifica que cada una se encuentra en 3NF.

**ticket:** ticket\_id es la superllave de ticket, la cual es creada al normalizar los datos. Con el id se puede

determinar las características propias del ticket, tales como el número de ticket, número de asiento, clase y comida y maleta

**cliente:** El atributo `cliente_id` es la superllave de la entidad cliente, también es creado al normalizar los datos. De este se obtiene la información personal del cliente, tal como pasaporte, que es único, nacionalidad y fecha de nacimiento.

**reserva:** La superllave de reserva es `reserva_id`, la cual viene con los datos iniciales. Con esta se obtiene el código de reserva, que es único, `vuelo_id` y `cliente_id`.

**reservapasajero:** la superllave es `reserva_id` y `cliente_id`, y estos dos atributos determinan el `ticket_id`, que es único.

## Rutas

Se tiene la entidad inicial dada:

**rutas**(`ruta_id`, `nombre_ruta`, `cardinalidad`, `nombre_punto`, `latitud`, `longitud`)

La cual se normaliza obteniendo las siguientes entidades:

- **ruta**(`ruta_id`, `nombre_ruta`)
- **rutapunto**(`ruta_id`, `punto_id`, `cardinalidad`)
- **punto**(`punto_id`, `nombre_punto`, `latitud`, `longitud`)

Dentro de las entidades existen las siguientes dependencias.

1. ruta:  $\{ruta\_id\} \rightarrow \{nombre\_ruta\}$
2. rutapunto:  $\{ruta\_id, punto\_id\} \rightarrow \{cardinalidad\}$
3. punto:  $\{punto\_id\} \rightarrow \{nombre\_punto, latitud, longitud\}$

Ahora, con las entidades ya normalizadas se justifica que cada una se encuentra en 3NF.

**punto:** La superllave de esta entidad es `punto_id`, esta se crea al normalizar los datos y determina el nombre del punto, que es único, la longitud y la latitud, que como tupla son únicos.

**ruta:** La superllave de esta entidad es `punto_id`, que es un atributo dado. Con el `punto_id` es posible obtener el nombre de dicho punto.

**rutapunto:** La superllave de rutapunto es `ruta_id` y `punto_id`, las cuales determinan la cardinalidad de una ruta.

## Trabajadores

Se tiene la entidad inicial dada:

**trabajadores**(`trabajador_id`, `nombre`, `fecha_nacimiento`, `pasaporte`, `nombre_compania`, `codigo_compania`, `licencia_actual_id`, `vuelo_id`, `rol`)

La cual se normaliza obteniendo las siguientes entidades:

- **trabajador**(`trabajador_id`, `nombre`, `fecha_nacimiento`, `pasaporte`)
- **trabajadorcompania**(`trabajador_id`, `compania_id`)
- **licenciatrabajador**(`licencia_actual_id`, `trabajador_id`)
- **vuelotrabajador**(`trabajador_id`, `id_vuelo`, `rol`)

Dentro de las entidades existen las siguientes dependencias.

1. trabajador:  $\{trabajador\_id\} \rightarrow \{nombre, fecha\_nacimiento, pasaporte\}$
2. trabajadorcompania:  $\{trabajador\_id, compania\_id\}$
3. licencia:  $\{licencia\_actual\_id\} \rightarrow \{trabajador\_id\}$
4. vuelotrabajador:  $\{trabajador\_id, id\_vuelo\} \rightarrow \{rol\}$

Ahora, con las entidades ya normalizadas se justifica que cada una se encuentra en 3NF.

**trabajador:** trabajador\_id es la superllave, esta venía con los datos iniciales. De este atributo se puede determinar su nombre.

**trabajadorcompania:** trabajador\_id es la superllave, esta venía con los datos iniciales. De este atributo se puede determinar su nombre.

**licencia:** licencia\_actual\_id es la superllave, esta venía con los datos iniciales. De este atributo se puede determinar el id del trabajador al cual le pertenece esta licencia.

**vuelotrabajador:** El trabajador\_id y el vuelo\_id son las superllaves de vuelotrabajador. Con estas se puede determinar el rol.

## Vuelos

Se tiene la entidad inicial trabajador\_id es la superllave, esta venía con los datos iniciales. De este atributo se puede determinar su nombre.dada:

**vuelos**(vuelo\_id, aerodromo\_salida\_id, aerodromo\_llegada\_id, ruta\_id, codigo\_vuelo, codigo\_aeronave, fecha\_salida, fecha\_llegada, velocidad, altitud, estado, nombre\_aeronave, modelo, peso, valor, nombre\_compania)

La cual se normaliza obteniendo las siguientes entidades:

- **vueloespecifico**(vuelo\_id, ruta\_id, codigo\_vuelo, aeronave\_id, fecha\_salida, fecha\_llegada, velocidad, altitud, estado)
- **compania**(compania\_id, nombre\_compania, codigo\_compania)
- **vuelogenerico**(codigo\_vuelo, aerodromo\_salida\_id, aerodromo\_llegada\_id, compania\_id)
- **aeronave**(aeronave\_id, codigo\_aeronave, nombre\_aeronave, modelo, peso)
- **costoticket**(ruta\_id, aeronave\_id, costo)

Dentro de las entidades existen las siguientes dependencias.

1. vueloespecifico: {vuelo\_id} → {ruta\_id, codigo\_vuelo, aeronave\_id, fecha\_salida, fecha\_llegada, velocidad, altitud, estado}
2. compania: {compania\_id} → {nombre\_compania, codigo\_compania}
3. vuelogenerico: {codigo\_vuelo} → {aerodromo\_salida\_id, aerodromo\_llegada\_id, compania\_id}
4. aeronave: {aeronave\_id} → {codigo\_aeronave, nombre\_aeronave, modelo, peso}
5. costoticket: {ruta\_id, aeronave\_id} → {costo}

Ahora, con las entidades ya normalizadas se justifica que cada una se encuentra en 3NF.

**vueloespecifico:** Tiene como superllave vuelo\_id, que es un atributo que estaba inicialmente y es único. De este se determina la ruta, el código de vuelo, aeronave, fecha de salida y llegada, velocidad, altitud y estado. Todas estas con atributos que se pueden repetir por si solos.

**vuelogenerico:** La superllave es el codigo\_vuelo, y este determina el aerodromo de salida, llegada y la compañía del vuelo. Estos atributos no son únicos, se pueden repetir ya sea cada uno por separado, como en su conjunto.

**compania:** Se crea un atributo compania\_id, es único y compone a la superllave de la entidad. De esta se obtiene el nombre de la compañía y su código.

**aeronave:** Se crea el atributo único aeronave\_id, el cual es la superllave de la entidad aerodromos. De este atributo se logran obtener todos los atributos característicos de la aeronave.

**costoticket:** La tupla ruta\_id y aeronave\_id es la superllave de costoticket. Esta tupla única determina el costo asociado.

## Consultas en SQL

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

```
SELECT *
FROM vueloespecifico, vuelogenerico, compania, aeronave, costoticket
WHERE vueloespecifico.codigo_vuelo = vuelogenerico.codigo_vuelo AND
      vueloespecifico.aeronave_id = aeronave.aeronave_id AND
      vueloespecifico.ruta_id = costoticket.ruta_id AND
      vuelogenerico.compania_id = compania.compania_id AND
      costoticket.aeronave_id = aeronave.aeronave_id AND
      vueloespecifico.estado = 'pendiente';
```

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

```
SELECT vueloespecifico.*, aerodromo.codigo_icao
FROM vueloespecifico, vuelogenerico, compania, aerodromo
WHERE vueloespecifico.codigo_vuelo = vuelogenerico.codigo_vuelo AND
      vuelogenerico.compania_id = compania.compania_id AND
      vueloespecifico.estado = 'aceptado' AND
      compania.nombre_compania = a AND
      vuelogenerico.aerodromo_llegada_id = aerodromo.aerodromo_id AND
      aerodromo.codigo_icao = c;"
```

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

```
SELECT ticket.numero_ticket, cliente.nombre_cliente, cliente.pasaporte_cliente,
       cliente.nacionalidad_cliente, cliente.fecha_nacimiento_cliente,
       ticket.numero_asiento, ticket.clase,
       ticket.comida_y_maleta, costoticket.costo
FROM reservapasajero, reserva, vueloespecifico, costoticket, cliente, ticket
WHERE reserva.reserva_id = reservapasajero.reserva_id AND
      reserva.codigo_reserva = c AND
      reserva.vuelo_id = vueloespecifico.vuelo_id AND
      vueloespecifico.ruta_id = costoticket.ruta_id AND
      reservapasajero.cliente_id = cliente.cliente_id AND
      ticket.ticket_id = reservapasajero.ticket_id AND
      vueloespecifico.aeronave_id = costoticket.aeronave_id
;
```

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

```
SELECT compania.compania_id, compania.nombre_compania, cliente.cliente_id,
       cliente.nombre_cliente, a.compras
FROM cliente, compania, (
  SELECT a.compania_id as compania_id, MAX(a.compras) as max
  FROM cliente, (
    SELECT reserva.cliente_id as cliente_id,
           compania.compania_id as compania_id, COUNT(*) as compras
    FROM reserva, compania, vueloespecifico, vuelogenerico, reservapasajero
    WHERE reserva.vuelo_id = vueloespecifico.vuelo_id and
           vueloespecifico.codigo_vuelo = vuelogenerico.codigo_vuelo and
           vuelogenerico.compania_id = compania.compania_id AND
           reservapasajero.reserva_id = reserva.reserva_id
    GROUP BY compania.compania_id, reserva.reserva_id
```

```

) as a
WHERE a.cliente_id = cliente.cliente_id
GROUP BY a.compania_id) as b,
(SELECT reserva.cliente_id as cliente_id, compania.compania_id
as compania_id, COUNT(*) as compras
FROM reserva, compania, vueloespecifico, vuelogenerico, reservapasajero
WHERE reserva.vuelo_id = vueloespecifico.vuelo_id and
vueloespecifico.codigo_vuelo = vuelogenerico.codigo_vuelo and
vuelogenerico.compania_id = compania.compania_id AND
reservapasajero.reserva_id = reserva.reserva_id
GROUP BY compania.compania_id, reserva.reserva_id) as a
WHERE compania.compania_id = b.compania_id AND
a.compras = b.max AND a.compania_id = b.compania_id AND
a.cliente_id = cliente.cliente_id
ORDER BY compania.compania_id ;

```

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

```

SELECT vueloespecifico.estado, COUNT(*) as cantidad_de_vuelos
FROM vueloespecifico, vuelogenerico, compania
WHERE vueloespecifico.codigo_vuelo = vuelogenerico.codigo_vuelo AND
vuelogenerico.compania_id = compania.compania_id AND
compania.nombre_compania = a
GROUP BY vueloespecifico.estado
;

```

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

```

2) as porcentaje
SELECT t.compania_id , a.nombre_compania, ROUND(a.aceptados * 1.0 * 100/t.todos,
FROM (
SELECT compania.compania_id as compania_id, COUNT(*) as todos
FROM compania, vuelogenerico, vueloespecifico
WHERE vuelogenerico.codigo_vuelo = vueloespecifico.codigo_vuelo and
vuelogenerico.compania_id = compania.compania_id
GROUP BY compania.compania_id
) as t,
(
SELECT compania.nombre_compania as nombre_compania,
compania.compania_id as compania_id, COUNT(*) as aceptados
FROM compania, vuelogenerico, vueloespecifico
WHERE vuelogenerico.codigo_vuelo = vueloespecifico.codigo_vuelo and
vuelogenerico.compania_id = compania.compania_id and
vueloespecifico.estado = 'aceptado'
GROUP BY compania.compania_id
) as a
WHERE a.compania_id = t.compania_id
ORDER BY porcentaje DESC LIMIT 1
;

```

\*Supuestos y detalles para la corrección.

Se cambio el valor de codigo\_vuelo de la linea 68 de vuelos.csv de KAI2257 a KAI2258 debido a que estaba repetido y generaba problemas.