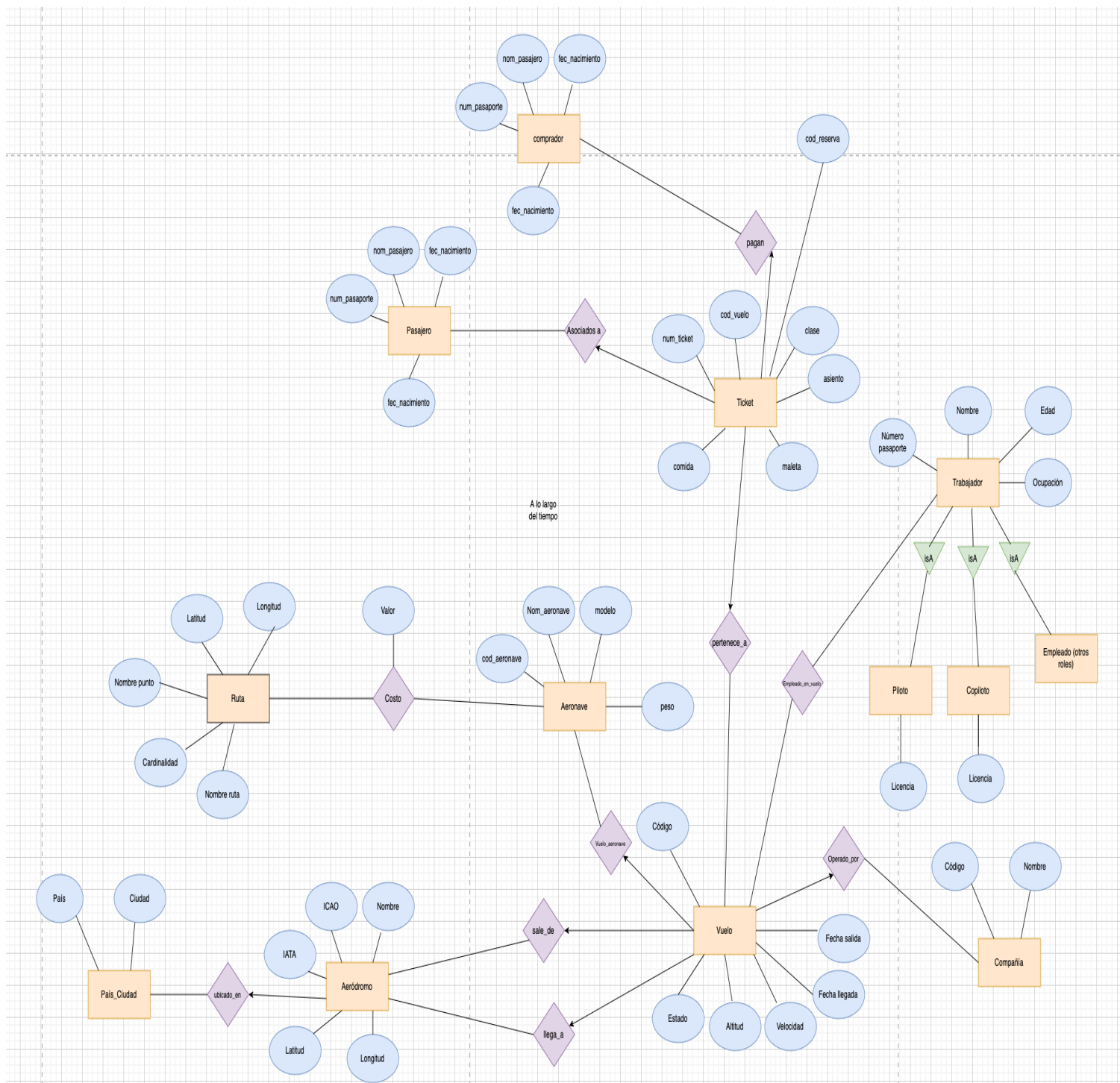


Diagrama E/R

Dejamos el link, ya que no todo se alcanza a distinguir pegando la imagen

<https://drive.google.com/file/d/1IKDHF7wkW8RoqhxclFkjhTHZmJr-co3t/view?usp=sharing>



Dependencias funcionales

1. Tabla compañía
Codigo_compania → nombre_compania
Está en BCNF, ya que se encuentra en su mínima descomposición
2. Tabla piloto
trabajador_id → pasaporte, nombre, fecha_nacimiento, rol, licencia_actual_id
Se encuentra en 3NF, ya que se podría hacer una descomposición más creando la relación: pasaporte → nombre, fecha_nacimiento
3. Tabla copiloto
trabajador_id → pasaporte, nombre, fecha_nacimiento, rol, licencia_actual_id
Se encuentra en 3NF, ya que se podría hacer una descomposición más creando la relación: pasaporte → nombre, fecha_de_nacimiento
4. Tabla empleado
Trabajador_id → pasaporte, nombre, fecha_nacimiento, rol
Se encuentra en 3NF, ya que se podría hacer una descomposición más creando la relación: pasaporte → nombre, fecha_de_nacimiento
5. Tabla vuelo
vuelo_id → codigo_vuelo, aeródromo_salida_id, aeródromo_llegada_id, código_aeronave, código_compania, fecha_salida, fecha_llegada, velocidad, altitud, estado, ruta_id
Se encuentra en 3NF, tiene más descomposiciones como: código_vuelo → aeródromo_salida_id, aeródromo_llegada_id
6. Tabla aeródromo
Aeródromo_id → nombre, código_ICAO, código_IATA, latitud, longitud, nombre_ciudad
Se encuentra en 3NF, ya que para estar en BCNF debería tener más descomposiciones como, por ejemplo: código_IATA → nombre
7. Tabla aeronave
Código_aeronave → nombre_aeronave, modelo, peso
No puede ser descompuesta otra vez, se encuentra en BCNF
8. Tabla pasajero
Pasaporte_pasajero → nombre_pasajero, nacionalidad_pasajero, fecha_nacimiento_pasajero
No puede ser descompuesta otra vez, se encuentra en BCNF
9. Tabla comprador
Pasaporte_comprador → nombre_comprador, nacionalidad_comprador, fecha_nacimiento_comprador
No puede ser descompuesta otra vez, se encuentra en BCNF

10. Tabla ticket

numero_ticket → código_reserva, reserva_id, vuelo_id, pasaporte_comprador, pasaporte_pasajero, numero_asiento, clase, comida_y_maleta

Se encuentra en 3NF, puede seguir descomponiéndose en tablas como:

Código_reserva → numero_asiento, clase, comida_y_maleta

11. Tabla costo

Id_costo → ruta_id, peso, valor

Se encuentra en BCNF

Como no todas las relaciones se encuentran en BCNF, pero si en 3NF podemos decir que nuestro modelo cumple con las condiciones 3NF de la teoría. Pero no con las BCNF. Como grupo consideramos que esta sería la mejor manera de modelar esta situación, ya que con 3NF es muy difícil perder información y la redundancia sigue siendo mínima, no generando problemas durante el proceso de modelaje.

Consultas SQL

Consulta 1:

```
SELECT vuelo_id, codigo_vuelo FROM vuelo WHERE estado = 'pendiente'
```

Consulta 2:

```
SELECT distinct vuelo_id
FROM vuelo, compania, aerodromo
WHERE aerodromo_llegada_id = (select aerodromo.aerodromo_id from aerodromo
where aerodromo.codigo_ICAO = '$id_icao_llegada' LIMIT 1)
and aerodromo.aerodromo_id = vuelo.aerodromo_llegada_id
AND compania.codigo_compania = (select compania.codigo_compania from compania
where compania.nombre_compania = '$nombre_compania' LIMIT 1)
AND vuelo.codigo_compania = compania.codigo_compania
AND vuelo.estado = 'aceptado';
```

Consulta 3:

```
SELECT ticket.numero_ticket, pasajero.nombre_pasajero, costo.valor
FROM ticket, pasajero, aeronave, vuelo, costo
WHERE ticket.codigo_reserva = '$reserva'
AND ticket.pasaporte_pasajero = pasajero.pasaporte_pasajero
AND ticket.vuelo_id = vuelo.vuelo_id
AND vuelo.codigo_aeronave = aeronave.codigo_aeronave
AND vuelo.ruta_id = costo.ruta_id
AND aeronave.peso = costo.peso
```

Consulta 4:

```
WITH grouped as (SELECT comprador.nombre_comprador as nombre_comprador,
compania.nombre_compania as nombre_compania, count(*) as countX,
    count(*) OVER (PARTITION BY compania.nombre_compania) AS cat_cnt,
    ROW_NUMBER() OVER (PARTITION BY compania.nombre_compania ORDER BY COUNT(*)
DESC) AS rn
    FROM compania, ticket as t1 , comprador, vuelo
    WHERE t1.vuelo_id = vuelo.vuelo_id
    AND compania.codigo_compania = vuelo.codigo_compania
    AND t1.pasaporte_comprador = comprador.pasaporte_comprador

    GROUP BY comprador.nombre_comprador, compania.nombre_compania
    ORDER BY countX DESC
)
SELECT nombre_comprador, nombre_compania, countX
FROM grouped
WHERE cat_cnt >= 1
    AND rn = 1
```

Consulta 5:

```
SELECT vuelo.estado, count(estado)
    FROM vuelo, compania
    WHERE compania.codigo_compania = (SELECT codigo_compania from compania where
nombre_compania LIKE '%$nombre_compania%')
    AND vuelo.codigo_compania = compania.codigo_compania
    GROUP BY vuelo.estado";
$result = $db -> prepare($query);
$result -> execute();
$vuelos = $result -> fetchAll()
```

Consulta 6:

```
SELECT tabla1.nombre_compania, cast((cast(tabla1.aceptados as decimal) /
cast(tabla2.totales as decimal))*100 as integer) as porcentaje
    FROM (
        SELECT compania.nombre_compania, count(vuelo_id) as aceptados
        FROM vuelo, compania
        WHERE vuelo.estado = 'aceptado'
        and vuelo.codigo_compania = compania.codigo_compania
        GROUP BY nombre_compania
    ) as tabla1, (
        SELECT nombre_compania, count(vuelo_id) as totales
        FROM vuelo, compania
        WHERE vuelo.codigo_compania = compania.codigo_compania
```

```
GROUP BY nombre_compania
) as tabla2
WHERE tabla1.nombre_compania = tabla2.nombre_compania
order by porcentaje desc
FETCH FIRST 1 ROWS ONLY;
```

Supuestos:

- En la consulta 4, dice “Por cada aerolinea, muestre **al** cliente que ha comprado la mayor cantidad de tickets”. Dado que dice “al cliente”, y no “a los clientes”, decidimos mostrar solo a un cliente, aun cuando hayan habido dos personas que compraron la misma cantidad máxima de tickets.