Relational Schema and Update/Delete Behavior

By: Yuv Rout, Aditya Gutha, Vignesh Minjur, Rishi Sukumar
Team Number: 72
Title: *Airport Database Management System*
Date: Feb 28, 2025

Relational Schema:

Person = (personID, first, last, locationID[FK1])
FK1: locationID → **location**(locID)
On update cascade – If a location changes, the person's recorded location should be updated accordingly.
On delete set null – If a location is removed, the person should not be deleted but instead have a NULL location.

Pilot = (personID [FK2], taxID, experience)
FK2: personID → **person**(personID)
unique attributes: {taxID}
On update restrict – The primary key for a person should not change; restricting updates ensures data integrity.
On delete cascade – If a person is deleted, their role as a pilot should also be removed.

Pilot_License = (personID [FK3], license)
FK3: personID → **pilot**(personID)
On update restrict – A pilot's identity should remain fixed.
On delete cascade – If a pilot is deleted, their associated licenses should also be removed.

Passenger = (personID [FK4], funds, miles)
FK4: personID → **person**(personID)
On update restrict – Ensures that a passenger's identity remains unchanged.
On delete cascade – If a passenger is deleted, their financial details (funds, miles) should also be removed.

Passenger_Vacation = (personID [FK5], destination, sequence)
FK5: personID → **passenger**(personID)
On update restrict – Ensures that a passenger's vacation records remain tied to their identity.
On delete cascade – If a passenger is removed, their vacation records should also be removed.

Airline = (airlineID, revenue)

Airplane = (airlineID [FK6], tail_num, speed, seat_cap, locationID [FK7])
FK6: airlineID → **airline**(airlineID)
On update cascade – If an airline ID changes, the associated airplanes should reflect that change.
On delete restrict – An airline should not be deleted if it still owns airplanes.
FK7: locationID → **location**(locID)
On update cascade – If a location changes, the airplane's recorded location should be updated accordingly.
On delete set null – If a location is removed, airplanes should not be deleted but instead have a NULL location.

Boeing = (airlineID [FK8], tail_num [FK9], maintained, model)
FK8: airlineID → **airplane**(airlineID)
On update cascade – If an airplane's airline changes, Boeing aircraft should reflect this.
On delete cascade – If an airplane is deleted, its Boeing classification should also be removed.

FK9: tail_num → **airplane**(tail_num)
On update cascade – If an airplane's tail number is updated, its Boeing record should also update.
On delete cascade – If an airplane is removed, its Boeing record should be deleted.

Airbus = (airlineID [FK10], tail_num [FK11], variant)
FK10: airlineID → **airplane**(airlineID)
On update cascade – If an airline changes, its Airbus fleet should be updated.
On delete cascade – If an airplane is removed, any associated Airbus record should also be deleted.
FK11: tail_num → **airplane**(tail_num)
On update cascade – If an airplane's tail number is updated, its Airbus record should reflect this.
On delete cascade – If an airplane is removed, any associated Airbus record should also be deleted.

Airport = (airportID, name, city, state, country, locationID [FK12])
FK12: locationID → **location**(locID)
On update cascade – If a location changes, the corresponding airport should update.
On delete restrict – Airports should not be deleted automatically if a location is removed.

Leg = (legID, distance, fromairport [FK13], toairport [FK14])
FK13: fromairport → **airport**(airportID)
FK14: toairport → **airport**(airportID)
On update cascade – If an airport ID is updated, the route should reflect this change.
On delete restrict – Routes should not be deleted if the airport still exists.

Route = (routeID)

Contains = (routeID [FK15], legID [FK16], sequence)
FK15: routeID → **route**(routeID)
FK16: legID → **leg**(legID)
On update cascade – If a route ID or leg ID changes, their associations should update.
On delete cascade – If a route or leg is removed, its connections should also be removed.

Flight = (flightID, cost, routeID [FK17], pilotID [FK18])
FK17: routeID → **route**(routeID)
On update cascade – If a route ID is updated, flights should also reflect this.
On delete cascade – If a route is removed, associated flights should also be deleted.
FK18: pilotID → **pilot**(taxID)
On update cascade – If a pilot ID changes, flights should reflect this.
On delete set null – If a pilot is deleted, flights should retain their data but show NULL for the pilot.

Supports = (flightID [FK19], airlineID [FK20], tail_num [FK21], progress, status, next_time)
FK19: flightID → **flight**(flightID)
FK20: airlineID → **airplane**(airlineID)
FK21: tail_num → **airplane**(tail_num)
On update cascade – If any of these foreign keys change, the respective records should be updated.
On delete cascade – If a flight, airline, or airplane is removed, the "supports" relationship should also be removed.

Location = (locID)