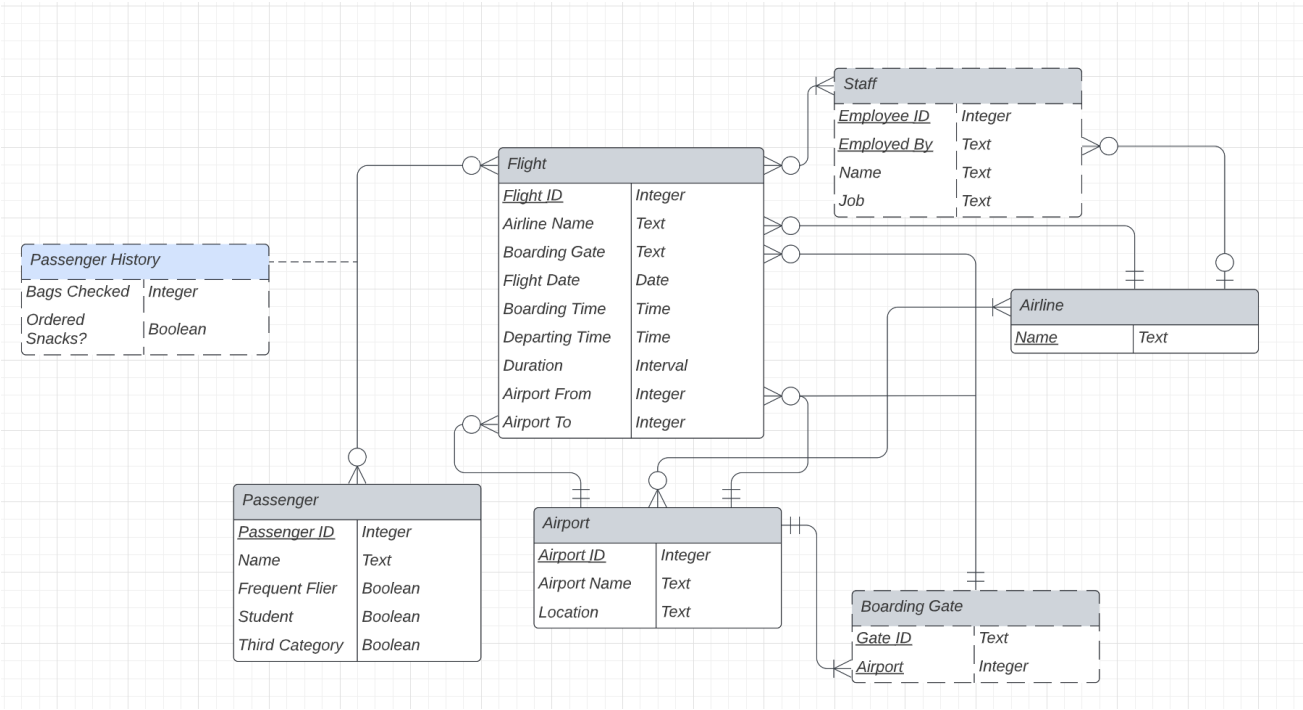
CSC460 Prog4 Writeup

Conceptual Database Design:

ER Diagram



Design Rationale

Passengers must have some kind of history, and that history has to be connected to what flights they've taken. Many passengers may be on one flight, one passenger may be on many flights. Flights must be connections between airports, one flight may go to many airports, and one flight may depart from many airports. This being the case, a flight might involve going to and from many boarding gates.
A given flight may switch out staff, staff may be on many flights, but staff generally belong to one airline. An airport may service many airlines, and an airline may service many airports.

There are constraints around boarding gates at airports (many flights may not use the same boarding gate at the same time), as well as flight date overlap (the same flight id may not occur multiple times on the same date with overlapping departure and arrival times), passenger overlap (the same passenger may not be on different flights simultaneously), and checked baggage (student and non-student baggage limits).

Logical Database Design:

Airline	(name)
Airport	(airport_id, airport_name, airport_code, location)
BoardingGate	(boarding_gate, airport_id)
Flight	(flight_id, airline, boarding_gate, flight_date, boarding_time, departing_time, duration, dest_from, dest_to)
Passenger	(passenger_id, name, frequent_flier, student, minor)
Staff	(employee_id, employed_by, name, job)
PassengerAirline	(passenger_id, airline)
PassengerFlight	(passenger_id, flight_id, bags_checked, ordered_snacks)
StaffFlight	(employee_id, flight_id)

Normalization Analysis:

Functional Dependencies:

Airport	airport_id	airport_name, airport_code, location
Flight	flight_id	airline, boarding_gate, flight_date, boarding_time, departing_time, duration, dest_from, dest_to
Passenger	passenger_id	name, frequent_flier, student, minor
Staff	employee_id	employed_by, name, job
PassengerFlight	passenger_id, flight_id	bags_checked, ordered_snacks

Justification:

This is in 1NF because all attributes are atomic and flattened.
This is in 2NF because all non-primary-key attributes are fully functionally dependent on the primary keys
This is in 3NF because every non-prime attribute of the tables is non-transitively dependent on every key of the tables.