-- Author: Makenzie Roberts

-- Last Edited: Oct 07, 2022

-------------------------------------------------------------------------------------------

-- NOTE: Sorry the schema name isn't relevant to the data - the autocomplete feature is pretty

-- slow for me and I didn't want to manually write a long name over and over again.

---------------------------------> SETTING UP THE TABLES <---------------------------------

-- First I'll create tables for our 4 main entities (with the suggested attributes):

-- Cities (id, name, state, population)

-- Passengers (id, first\_name, last\_name, phone\_number)

-- Airports (id, name, code)

-- Aircraft (id, type, airline\_name, number\_of\_passengers)

-- Now I'll take a quick look at my empty 'cities' table...

**SELECT \* FROM qap.cities;**

-- ... and insert some data into the 'cities' table.

**INSERT INTO qap.cities VALUES (1, 'phoenix', 'az', 1658000);**

**INSERT INTO qap.cities VALUES (2, 'denver', 'co', 715878);**

**INSERT INTO qap.cities VALUES (3, 'chicago', 'il', 2699000);**

**INSERT INTO qap.cities VALUES (4, 'atlanta', 'ga', 497642);**

**INSERT INTO qap.cities VALUES (5, 'orlando', 'fl', 284817);**

-- Now I'll take a look at my empty 'passengers' table...

**SELECT \* FROM qap.passengers;**

-- ... and insert some data into the 'passengers' table.

**INSERT INTO qap.passengers VALUES (1, 'samantha', 'jane', '480-200-4952', 1);**

**INSERT INTO qap.passengers VALUES (2, 'john', 'doe', '480-202-9966', 1);**

**INSERT INTO qap.passengers VALUES (3, 'brandon', 'smith', '480-204-2317', 1);**

**INSERT INTO qap.passengers VALUES (4, 'david', 'park', '303-200-3988', 2);**

**INSERT INTO qap.passengers VALUES (5, 'anthony', 'lewis', '303-201-7106', 2);**

**INSERT INTO qap.passengers VALUES (6, 'james', 'powell', '217-300-3333', 3);**

**INSERT INTO qap.passengers VALUES (7, 'fay', 'smith', '217-300-2673', 3);**

**INSERT INTO qap.passengers VALUES (8, 'ian', 'stone', '404-200-2132', 4);**

**INSERT INTO qap.passengers VALUES (9, 'sarah', 'shaw', '404-201-3350', 4);**

**INSERT INTO qap.passengers VALUES (10, 'phillip', 'hollis', '321-200-9366', 5);**

**INSERT INTO qap.passengers VALUES (11, 'chris', 'hunter', '321-201-5441', 5);**

**INSERT INTO qap.passengers VALUES (12, 'betty', 'holmes', '321-206-8516', 5);**

-- Now I'll take a look at my empty 'airports' table...

**SELECT \* FROM qap.airports;**

-- ... and insert some data into the 'airports' table. We have 5 cities in our

-- cities table, and I'd like to assign at least one airport to each city.

-- Since this table references the 'cities' table via city\_id, I'll be putting

-- appropriate names and codes for each city in our 'cities' table. For

-- instance - Because the city with an id of 1 has the name 'phoenix', I'll be

-- adding an airport from Phoenix, Arizona. In this case I'll choose "Phoenix Sky

-- Harbor Airport" (PHX). And since cities can have more than one airport, I'll

-- demonstrate that by adding another Phoenix airport - "Deer Valley Airport" (DVT)

-- NOTE: It's probably a bit redundant to include "airport" in every name but I did

-- just in case. In my mind either could be preferred depending on how the

-- data is going to be used.

**INSERT INTO qap.airports VALUES (1, 'phoenix sky harbor airport', 'phx', 1);**

**INSERT INTO qap.airports VALUES (2, 'deer valley airport', 'dvt', 1);**

**INSERT INTO qap.airports VALUES (3, 'denver international airport', 'den', 2);**

-- NOTE: Had to use two ''s here to escape the single quote in "o'hare"

**INSERT INTO qap.airports VALUES (4, 'o''hare international airport.', 'ord', 3);**

**INSERT INTO qap.airports VALUES (5, 'hartsfield-jackson atlanta international airport', 'atl', 4);**

**INSERT INTO qap.airports VALUES (6, 'orlando international airport', 'mco', 5);**

-- Cool! Just for fun, I'll select the airport name and code columns, and the cities name and state columns,

-- This query will provide us the airport name, airport code, city name and city state columns of

-- every city that has been assigned an airport (AKA has the same id #). In this case, all of them will display.

**SELECT airports.name, airports.code, cities.name, cities.state**

**FROM qap.airports, qap.cities**

**WHERE cities.id = airports.city\_id**

**ORDER BY cities.name;**

-- Now I'll take a look at my empty 'aircraft' table...

-- NOTE: In the QAP document, 'number of passengers' was suggested as a column - I'm

-- assuming that is referring to the maximum capacity of the airplane.

**SELECT \* FROM qap.aircrafts;**

-- ... and insert some data into the 'aircraft' table.

**INSERT INTO qap.aircrafts VALUES (1, 'boeing 747-400', 'northwest airlines', 524);**

**INSERT INTO qap.aircrafts VALUES (2, 'boeing b757-200', 'northwest airlines', 182);**

**INSERT INTO qap.aircrafts VALUES (3, 'airbus a330-300', 'american airlines', 300);**

**INSERT INTO qap.aircrafts VALUES (4, 'boeing 777-300', 'american airlines', 550);**

**INSERT INTO qap.aircrafts VALUES (5, 'boeing 777-200', 'american airlines', 440);**

-- Awesome. So far I've created a one-to-one relationship:

-- [cities]---[passengers]

-- and a one-to-many relationships:

-- [cities]--<[airports]

-- So now, I'll create a couple many-to-many relatonships using bridge tables:

-- [passengers]--<[aircraft\_passengers]>---[aircrafts]

-- and

-- [airports]--<[airport\_aircrafts]>--[aircrafts]

-- Notice that I intend to create a bridge table for 'airports' and 'aircrafts' called 'airport\_aircrafts'

-- To me, the relationship between the other tables is clear, but for 'airports' and 'aircrafts' we've

-- been given this sentence under 'Key Relationship Information': "Aircraft can land/take off from many Airports"

-- In my mind this also means that an airport can house many different aircraft - AKA a many-to-many relationship.

-- My logic is based on this statement from the docs, under "Many-to-Many" (https://launchschool.com/books/sql\_first\_edition/read/multi\_tables):

-- "Example: A user has many books checked out or may have checked them out in the past. A book has many users

-- that have checked a book out."

-- So, if I convert that logic to work with our air travel case study:

-- An aircraft can take off/land from many airports or may have landed/taken off from them in the past. An airport houses many aircraft that may have landed/taken off.

-- or...

-- An airport can house many aircraft or may have housed them in the past. An aircraft can have many airports that

-- it lands at/takes off from.

-- So, following the steps laid out in Lecture #5 (2022-09-21)...

-- I'll create the bridge table for 'airports' and 'aircrafts' named 'airport\_aircrafts'

-- Now I'll take a look at the empty table to confirm everything is set up as I intended...

**SELECT \* FROM qap.airport\_aircrafts;**

-- Now I'll add some records to the table, effectively 'assigning' at least one aircraft to each airport.

**INSERT INTO qap.airport\_aircrafts VALUES (1, 5);**

**INSERT INTO qap.airport\_aircrafts VALUES (2, 4);**

**INSERT INTO qap.airport\_aircrafts VALUES (2, 5);**

**INSERT INTO qap.airport\_aircrafts VALUES (3, 3);**

**INSERT INTO qap.airport\_aircrafts VALUES (4, 1);**

**INSERT INTO qap.airport\_aircrafts VALUES (4, 2);**

**INSERT INTO qap.airport\_aircrafts VALUES (6, 3);**

**INSERT INTO qap.airport\_aircrafts VALUES (6, 4);**

**INSERT INTO qap.airport\_aircrafts VALUES (5, 1);**

**INSERT INTO qap.airport\_aircrafts VALUES (5, 3);**

-- Awesome! To check that everything is as I intended, I'll query for the airport name, aircraft airline name,

-- and aircraft type columns for every airport/aircraft that has been assigned to eachother.

**SELECT airports.name, aircrafts.airline\_name, aircrafts.type**

**FROM qap.airports, qap.aircrafts, qap.airport\_aircrafts**

**WHERE airports.id = airport\_aircrafts.airport\_id**

**AND aircrafts.id = airport\_aircrafts.aircraft\_id**

**ORDER BY airports.name;**

-- Great! From what I can tell it returned exactly what I expected.

-- I'll continue on and create the other bridge table, 'aircraft\_passengers'...

-- NOTE: I wasn't sure whether or not to add a composite primary key to this table

-- because depending on how this data would theoretically be used, it could

-- contain multiple records due to a passenger taking the same flight more

-- than once, and maybe someone would want to add a timestamp to that data

-- and archive it - but because I'm still new to this I want to closely follow

-- the instructions laid out in class.

-- Check out the empty table to make sure everything looks fine...

**SELECT \* FROM qap.aircraft\_passengers;**

-- ... and insert some data into the 'aircraft\_passengers' table.

**INSERT INTO qap.aircraft\_passengers VALUES (1, 1);**

**INSERT INTO qap.aircraft\_passengers VALUES (1, 2);**

**INSERT INTO qap.aircraft\_passengers VALUES (1, 7);**

**INSERT INTO qap.aircraft\_passengers VALUES (1, 9);**

**INSERT INTO qap.aircraft\_passengers VALUES (2, 6);**

**INSERT INTO qap.aircraft\_passengers VALUES (2, 3);**

**INSERT INTO qap.aircraft\_passengers VALUES (3, 2);**

**INSERT INTO qap.aircraft\_passengers VALUES (3, 4);**

**INSERT INTO qap.aircraft\_passengers VALUES (3, 5);**

**INSERT INTO qap.aircraft\_passengers VALUES (3, 8);**

**INSERT INTO qap.aircraft\_passengers VALUES (4, 1);**

**INSERT INTO qap.aircraft\_passengers VALUES (4, 2);**

**INSERT INTO qap.aircraft\_passengers VALUES (4, 5);**

**INSERT INTO qap.aircraft\_passengers VALUES (4, 10);**

**INSERT INTO qap.aircraft\_passengers VALUES (4, 11);**

**INSERT INTO qap.aircraft\_passengers VALUES (4, 12);**

**INSERT INTO qap.aircraft\_passengers VALUES (5, 3);**

**INSERT INTO qap.aircraft\_passengers VALUES (5, 4);**

-- Now to make sure everything looks right, I'll query passengers names and their aircraft information

**SELECT passengers.first\_name, aircrafts.airline\_name, aircrafts.type**

**FROM qap.passengers, qap.aircrafts, qap.aircraft\_passengers**

**WHERE aircrafts.id = aircraft\_passengers.aircraft\_id**

**AND passengers.id = aircraft\_passengers.passenger\_id**

**ORDER BY passengers.first\_name;**

-- Cool! Now I can do some fun things, like find out every city a passenger has flown to

-- based on the aircraft(s) they've travelled on and the corresponding airport(s) those aircrafts

-- have flown to.

-- (Hopefully I did this correctly, I'm just experimenting because it helps me grasp the concept!)

**SELECT passengers.first\_name, passengers.last\_name, cities.name, airports.name, aircrafts.type**

**FROM qap.passengers, qap.cities, qap.airports, qap.aircrafts, qap.airport\_aircrafts, qap.aircraft\_passengers**

**WHERE cities.id = airports.city\_id**

**AND passengers.id = aircraft\_passengers.passenger\_id**

**AND aircrafts.id = aircraft\_passengers.aircraft\_id**

**AND airports.id = airport\_aircrafts.airport\_id**

**AND aircrafts.id = airport\_aircrafts.aircraft\_id**

**ORDER BY passengers.first\_name;**

-------------------------------------> QUESTIONS <--------------------------------------

-- Questions to answer with SQL select statements:

-- 1. What airports are in what cities?

-- 2. List all aircraft passengers have travelled on?

-- 3. Which airports can aircraft take off from and land at?

-- 4. What airports have passengers used?

-- 1. What airports are in what cities?

-- To find out what airports are in what cities, I need to \*select\* the city name and airport

-- name columns \*from\* the cities and airports tables, respectively - and then only select

-- records \*where\* the city id matches the city\_id key in the airports table. Lastly, I \*ordered\*

-- it by city name for readability.

**SELECT cities.name, airports.name**

**FROM qap.airports, qap.cities**

**WHERE cities.id = airports.city\_id**

**ORDER by cities.name;**

-- 2. List all aircraft passengers have travelled on?

-- To list all the aircrafts passengers have travelled on, I need to access one of the

-- bridge tables I made called 'aircraft\_passengers' - \*from\* that table, I'll \*select\* the

-- passenger first/last name and aircraft type/airline-name columns, and then only the records

-- \*where\* both the passenger id and aircraft id's match their foreign keys in the

-- 'aircraft\_passengers' table.

**SELECT passengers.first\_name, passengers.last\_name, aircrafts.airline\_name, aircrafts.type**

**FROM qap.passengers, qap.aircrafts, qap.aircraft\_passengers**

**WHERE passengers.id = aircraft\_passengers.passenger\_id**

**AND aircrafts.id = aircraft\_passengers.aircraft\_id**

**ORDER BY passengers.first\_name;**

-- 3. Which airports can aircraft take off from and land at?

-- Same process as before - I selected some columns containing relevant information

-- (aircraft airline name and type, airport code and name, city), and then only records

-- that have both the airport and aircraft id's as keys in the airport\_aircrafts table

-- which links the aircrafts to their airports.

-- NOTE: I was just messing around and noticed that if I add more than one expression to

-- the ORDER BY clause, it will subsort the list? So now it's sorted by airline

-- name and subsorted by aircraft type. Cool :)

**SELECT aircrafts.airline\_name, aircrafts.type, airports.code, airports.name, cities.name**

**FROM qap.aircrafts, qap.airports, qap.airport\_aircrafts, qap.cities**

**WHERE aircrafts.id = airport\_aircrafts.aircraft\_id**

**AND airports.id = airport\_aircrafts.airport\_id**

**AND cities.id = airports.city\_id**

**ORDER BY aircrafts.airline\_name, aircrafts.type;**

-- 4. What airports have passengers used?

-- Same process as above except this query pulled from both bridge tables, not just one.

-- To find airports a passenger has used, I first needed to match a passenger's id to

-- aircrafts they've flown in (from it's key contained in the aircraft\_passengers table), and

-- then take that aircraft id and match it with the airport(s) it's landed at/taken off from.

-- (from it's key contained in the airport\_aircrafts table). Lastly I added some other relevant

-- information, like the airport code and city name.

**SELECT passengers.first\_name, passengers.last\_name, airports.code, airports.name, cities.name**

**FROM qap.passengers, qap.cities, qap.airports, qap.aircrafts, qap.airport\_aircrafts, qap.aircraft\_passengers**

**WHERE cities.id = airports.city\_id**

**AND passengers.id = aircraft\_passengers.passenger\_id**

**AND aircrafts.id = aircraft\_passengers.aircraft\_id**

**AND airports.id = airport\_aircrafts.airport\_id**

**AND aircrafts.id = airport\_aircrafts.aircraft\_id**

**ORDER BY passengers.first\_name;**