CPSC 321: Homework 1 *Due: Wednesday, Sept. 10, 2025*

Instructions. Write your answers to the following questions and submit them on canvas by the end of the day on the due date (please note that canvas will not accept the submission after 11:59pm). Your answers must be organized by question number and the question clearly marked. Also include your first and last name, the course (CPSC-321), the section, and the assignment number at the top of your write up. Your answers must be submitted as a *single* PDF file called `hw-1.pdf`. The homework assignment is worth 25 points: up to 15 points for your submission and up to 10 points for the in-class assignment check-in.

Part A: Historical Context. For this part, read part 1 of "*50 Years of Queries*" (Chamberlin, 2024), and then answer the following questions.

1. List 3 things that you found particularly suprising and/or interesting from the article and why. Additionally, list at least one item that you considered including in your top-3 but that didn't make the cut (and why).

2. Give a brief timeline of the major events described in the article. For each event you include, provide a brief explanation of why you included it (i.e., why you felt it was important to include in the timeline). You are free to write down the timeline as you think best, but be sure to make the order clear (and include years where possible).

3. What were the author's initial impressions of Codd's 1970 paper? Why did his views change? Note that in many fields of computer science, most new and/or transformational products created in industry come from research advances. According to the article, what are some ways this continued after Codd? Give at least two other similar advances in computer science where research and theory work led to high-profile industry products.

Part B: Warm Up. Answer the following question related to the scope of database systems within products today.

4. Pick a desktop, web, or mobile (Android/iOS) app that you frequently use and that clearly persists data. (By the way, can you think of a product that doesn't?) Assume the app uses a relational database for storing transactional data. List the kinds of relations you think might be used and the general attributes you think they contain. You do not need to specify details (e.g., domains, keys, foreign keys, and other constraints). Instead, think of this as brainstorming all of the kinds of information (as relations and basic attributes) that might be needed by the application.

5. In your own words, describe the main difference between OLAP and OLTP. Using your answer in 4 describe how each type of system might play a part in the application.

Part C: Relational Schemas

6. Create relation schemas that can be used to store the following information. Your schemas must include each relation's name, attributes, primary keys, and foreign keys (as appropriate). In addition, for each relation provide at least three example rows.

(a). US airports consisting of a unique three-letter FAA identifier (e.g., SEA, LAX, PDX, etc.), a name (e.g., "Seattle-Tacoma International", "Los Angeles International"), the city and the state it is located in or nearest to, and its elevation (in feet). Assume every airport has a different three-letter identifier (assigned by the FAA), but that there can be cases where two different airports have the same name (e.g., there are multiple airports named "John F. Kennedy").

(b). Airlines consisting of a unique two-character airline code (e.g., AS, WN, UA), an airline name (e.g., "Alaska Airlines", "Southwest Airlines"), a three-letter FAA airport identifier (see (a) above) that represents the airline's primary "hub" airport, and the year the airline was founded.

(c). Flight information consisting of the airline code (e.g., AA), a flight number (e.g., 1), the departure airport, the arrival airport, and the number of flights per week. Note that a flight is identified by combining its airline code and flight number (i.e., no two flights will have the same combination). While flights can have many different "segments", i.e., stops between the departure and arrival airports, the flight relation only consists of the first departure airport and the final arrival airport of each flight.

(d). Flight segment information that specifies stops along a flight (if the flight is not non-stop). A flight segment consists of the airline and flight number, an airport code for where the segment starts (departs), an airport code for where the segment ends (arrives), and a segment "offset", i.e., an integer denoting whether it is the flight's first, second, third, and so on, stop. Note that any particular flight could have zero, one, or more segments (stops). Also, offset values are only unique with respect to a particular flight.

7. Draw a single table (schema) diagram that contains each of your relation definitions in 5(a) through 5(d) above. Be sure to look back at the lecture notes for details on what a table (schema) diagram is.

8. Create relational schemas (one schema for each part) that can be used to store the following information. Your relational schemas must include for each relation the relation name, attributes, primary keys, and foreign keys (as appropriate). In addition, for each relation provide at least three example tuples.

(a). Employees consisting of a unique employee id, a first name, a last name, employment start date, and the name of the department the employee works in (e.g., "Human Resources" or "Software Development").

(b). Projects consisting of a unique project number, the name of the department that sponsors the project, the employee that is the project lead, the project start date, and the project annual budget.

(c). Manager information specifying the employees that manage other employees, along with the date the employee started managing the employee. Note that each employee can have at most one manager (but possibly none).