**CSDS 234 Structured and Unstructured Data  
Department of Computer & Data Sciences, Fall 2024**

Assignment 1



1. [**Data & Attribute Types**] [20]

Determine the type of the following attributes of real-world objects (Nominal, Ordinal, Interval, Ratio). Justify your answers with brief explanations.

1. Colors of M&M candies
   * Nominal, the colors of the candies are attributes that have no ranking involved with them
2. Total annual income of computer science students in the 2018 class
   * Ratio, there is an absolute 0
3. Weights of blue whales in pounds
   * Interval, there is no 0 pound whale and the distance is meaningful
4. Rating of restaurants in {Excellent, Above Average, Average, Below Average, Poor}
   * Ordinal, the rating of the restaurants can be ranked, but they are also attributes
5. Human Blood Type
   * Nominal, the blood types are all attributes that have no inherent ranking with them
6. GINI coefficient of African countries
   * Ratio, the values range from 0-1 where 0 is perfect equality and 1 is perfect inequality. There is a meaningful fraction.
7. Memory cost measured in MB of a library of anti-virus software
   * Ratio, there are no values that are lower than 0 and as such it is a meaningful fraction of the amount of MB that can be used
8. Processing temperatures of alloys of Aluminum in Celsius
   * Interval, the distance is meaningful and is a temperature is in Celsius
9. Home addresses of customers
   * Nominal, the attributes are named but there is no order to it
10. The probability a person wins a lottery
    * Ratio, there is a 0% chance if no ticket is bought, 100% if only 1 ticket is bought, and everything in between. It is a meaningful fraction.
11. **[Data Models and Types of Data]** [30]
    1. (14) What is a data model? Describe the three perspectives of a data model.

A data model is a framework that defines the structure of the data or it’s relationship. It creates the map of how the data will be accessed and stored in the system. Data models ensure consistency for the data management process. The three perspectives of a data model is the conceptual perspective, logical perspective, and the physical perspective. The conceptual perspective is what the data stricture will look like at the highest level. It is the form of the data structure that can most easily be understood. The logical perspective delves deeper into what the structure of the data is and includes more constraints and relationships that might not be seen in the conceptual perspective. The physical perspective is the technical implementation of the database and is in a specific database system. This one is the most technical and describes how data can be accessed and how the storage should be made for the most efficiency.

* 1. (16) We have seen several examples of different types of data. Based on your understanding, determine the proper types of the following dataset (structured, semi-structured, unstructured, ordered data, time-series data), for a given context of application scenario. Briefly justify your answer. A dataset can be assigned with multiple types, whenever applicable.
     1. Sensor data from a weather station that records windspeed, irradiance, and temperatures
        + Structured, the data might be in a table with all of the records next to it and the sensor data from the weather station needs to be connected to the other records
     2. Google map data, for searching a shortest path from location A to location B
        + Semi-structured, it is a type of spatial data and was be represented as a graph rather than as a table, which leads it to be semi-structured
        + Ordered, the map data is spatial so it must be in an order
     3. A set of Tweets between May 2022 to August 2022 discussing “House of the Dragon”, yet all timestamps are missing
        + Unstructured, because it is words in a social media format
     4. A fraction of genome sequences
        + Unstructured Data, if the fraction of genome sequences are in a raw text file with no additional structure or labels.
     5. A sequence of images sampled from a movie when it is played
        + Unstructured, it is of images which is inherently unstructured
     6. A list of image features defined by attributes “{image name, pixel number, topic, color-depth, height, length}”
        + Structured Data, there is a consistent scheme and it can easily be kept as a data table with each of the attributes being in a separate column
     7. Webpages in HTML code
        + Semi structured, the webpages are in HTML which is inherently semi-structured
     8. Real-time Stock price feeds
        + Time Series Data, the data is affected by the time and time gives it meaning

1. **[Relational Data Model] [50]**

Consider the following schema defined for Lake Erie Cruises. The schema keeps track of ships, cruises, ports and passengers. A cruise refers to the sailing of a ship on a specific date.

Ship(ship\_number, ship\_name, ship\_builder, departure\_date, gross\_weight)

Cruise(cruise\_number, start\_date, end\_date, director, ship\_number)

Port(port\_name, country, dock\_number, port\_manager)

Visit(cruise\_number, port\_name, country, arrival\_date, departure\_date)

Passenger(passenger\_number, passenger\_name, SSN, Address, Phone)

Voyage(passenger\_number, cruise\_number, stateroom\_number, fare)

The following facts have been validated.

* Both ship number and ship name are unique in Ship relation.
* A ship goes on many cruises. A cruise is associated with a single ship.
* A port can be uniquely identified by a port name and country
* A cruise may visit multiple ports, and a port can be included as a stop by multiple cruises.
* A passenger has a unique passenger number, and a unique SSN.
* A person has a single passenger number that is used for all cruises she takes.
* A voyage indicates that a person can take many cruises. A cruise, as expected, can have multiple passengers.

Answer the following questions.

* (20) Identify a primary key, and a candidate key for each relation.
  + Ship:
    - Primary Key: ship\_number
    - Candidate Key: ship\_name
  + Cruise:
    - Primary Key: cruise\_number
    - Candidate Key: cruise\_number
  + Port:
    - Primary Key: port\_name
    - Candidate Key: country
  + Visit:
    - Primary Key: cruise\_number & departure\_date
    - Candidate Key: cruise\_number & arrival\_date
  + Passenger:
    - Primary Key: passenger\_number
    - Candidate Key: SSN
  + Voyage:
    - Primary Key: passenger\_number & cruise\_number
    - Candidate Key: : passenger\_number & cruise\_number
* (10) Identify the foreign keys of each relation, given your choice of primary keys.
  + Ship:
    - departure\_date (Visit)
  + Cruise:
    - Cruise\_number (Visit)
    - Ship\_number(Ship)
  + Port:
    - Port\_name (Visit)
  + Visit:
    - Port\_name (Port)
    - Cruise\_number (Cruise)
  + Passenger:
    - Passenger\_number (Voyage)
  + Voyage:
    - Passenger\_number (Passenger)
    - Cruise\_number (Visit)
* (10) Identify the foreign keys that are also a part of the primary keys of the same schema they are defined on.
  + For Voyage, passenger\_number is on the primary key but is also a foreign key
  + For Visit, cruise\_number is on the primary key but is also a foreign key
* (10) Happy Hour Lines wants to track which passengers visited which ports on which ships, and on which dates. In this case, which are the relations you will need? Design schemas as necessary to store this information.
  + Passenger (passenger\_number, passenger\_name)
  + Voyage (passenger\_number, cruise\_number)
  + Cruise (cruise\_number, ship\_number)
  + Ship (ship\_number, ship\_name)
  + Visit(cruise\_number, arrival\_date, departure\_date)

1. **[Data Constraints\*]** (10) This is a bonus question.

Given a referencing relation R1 with foreign key FK1, and a referenced relation R2 with primary key PK2 that FK1 refers to, and two states r(R1), and r(R2), describe an algorithm that checks if r(R1) and r(R2) violate the foreign key constraint. You may use pseudocode, or simply describe it. It is also encouraged that you provide a time cost analysis for the algorithm you proposed.

* Create a set that stores all the values of PK2 from r(R2)
* For each of the tuples in r(R1) check if the value of FK1 exists in the previous set
* If the value does not exist, then the foreign key constraint is no longer valid
  + If all values are found, then the foreign key constraint is not violated
* Time Cost: O(n+m)
  + Step 1: Iterate through all of r(R2) and add to a Set
  + Iterate through r(R1) and check it against the Set (which is a time complexity of O(1)