HW1CSDS234

1 Data and Attributes Types

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

1. Flavors of Coca Cola (such as Classic, Cherry, Zero...)

Ans

Nominal, flavors are categorical and can not be ordered

2. Income of computer science students in US in 2024

Ans

Ratio, income can be categorical, orderable, addable, and multiplication (meaningful 0)

3. Weights of lions in a zoo, in pounds

Ans

Ratio, weight can be categorical, orderable, addable, and multiplication (meaningful 0)

4. Rating of a hotel in {Excellent, Above Average, Average, Below Average, Poor}

Ans

Ordinal, ratings of a hotel are categorical, orderable, but can not be added

5. Blood Type of Human

Ans

Nominal, blood types are categorical and can not be ordered

6. GINI coefficient of Asian countries

Ans

Ratio, GINI coefficient has meaningful 0

7. Memory cost measured in MB of a computer program

Ans

Ratio, memory costs can be categorical, oderable, additive, and multiplication (meaningful 0).

8. Processing temperatures of alloys of Aluminum in Celsius

Ans

Interval, celsius can be categorical, oderable, additive but not multiplicative

9. The probability a person wins a lottery

Ans

Ratio, probabilities are categorical, oderable, additive, and multiplicative (meaningful 0)

2 Data Models and Types of Data

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

i. Sensor data from a weather station that records windspeed, irradiance, and temperatures

Ans

Ordered data, sensor data is data over time so Ordered data

ii. Google map data, for searching a shortest path from location A to location B

Ans

Semi-structured, map data is akin to graphs, which are semi-structured

iii. A set of Tweets discussing “” with their posted time

Ans

Unstructured data, text data is considered unstructured.

iv. A fraction of genome sequences

Ans

Ordered data, genetic sequences are ordered data (given in slides)

v. A sequence of images sampled from a movie when it is played

Ans

Unstructured, image data is considered unstructured

vi. A list of image features defined by attributes “{image name, pixel number, topic, color-depth, height, length}”

Ans

Structured, this would create a data matrix which is considered structurerd data.

vii. Webpages in HTML code

Ans

Semi-structured, HTML code is considered semi-structured according to slides

viii. Real-time Stock price feed

Ans

Ordered data, stock prices are ordered over time (also shown in slides)

3 Relational Data and Models

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.

(10) Identify a primary key, and a candidate key for each relation.

Ans

Relation	PK	CK
Ship	ship_number	ship_name
Cruise	cruise_number	ship_number, start_date
Port	port_name, country	port_name, country
Visit	cruise_number, port_name, country, arrival_date	cruise_number, port_name, country, departure_date
Passenger	passenger_number	SSN
Voyage	passenger_number, cruise_number	passenger_number, cruise_number, stateroom_number

(10) Identify the foreign keys of each relation, given your choice of primary keys.

Ans

Relation	FK's
Ship	
Cruise	ship_number
Port	
Visit	cruise_number, (port_name, country)
Passenger	

Relation	FK's
Voyage	passenger_number, cruise_number

(10) Identify the foreign keys that are also a part of the primary keys of the same schema they are defined on.

Ans

Relation	FK's in PK
Ship	
Cruise	
Port	
Visit	cruise_number, (port_name, country)
Passenger	
Voyage	passenger_number, cruise_number

(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 a schema as necessary to store this information.

Ans

We will need the following relations:

- Voyage
- Cruise
- Visit

We will create the following schema
R(passenger_number, cruise_number, port_name, country, ship_number, arrival_date)

(10) Give three examples of functional dependencies that may likely hold on the schema, and explain why you think they will hold.

Ans

ship_number -> ship_name, ship_builder, gross weight
a ship number uniquely determine its name, builder, and weight. A ship number can not have more than one name / builder / weight.

cruise_number -> start_date, end_date, director, ship_number
a cruise number can not have more than one start date / end date / director / ship number

passenger_number -> passenger_name, SSN, Address, Phone
a passenger_number can not have more than one name / SSN / Address / Phone

4 Data Constraints

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 propose.

Check if FK1 is part of the PK of R1.
If FK1 $\in$ PK:

- Look for Nulls
- If there is a null: return "violates"

for each fk $\in$ FK1:
if fk $\notin$ PK2:
return "violates"

return "no foreign key violation"

If key lookups are constant time, runtime is $O(n)$
If key lookups are not constant time, runtime is $O(n^2)$