# *Assignment #1*

1. [Big Data concept] (10) Give one example of Big Data application you know. Use the detailed example to explain each of the five Big V's. If you are required to design a database system for this application, what are the best data models (relational, semi-structured, among others) you would use to represent the data and why?

2. [Relational Data Model] (30) As of January 2017, the OpenFlights Airports Database (https://openflights.org/data.html) contains over 10,000 airports, train stations and ferry terminals spanning the globe. Each entry in the Airport table contains the following:

---

**Airport ID** Unique OpenFlights identifier for this airport.
**Name** Name of airport. May or may not contain the City name.
**City** Main city served by airport. May be spelled differently from Name.
**Country** Country or territory where airport is located. See countries.dat to cross-reference to ISO 3166-1 codes.
**IATA** 3-letter IATA code. Null if not assigned/unknown.
**ICAO** 4-letter ICAO code.
**Latitude** Decimal degrees, usually to six significant digits. Negative is South, positive is North.
**Longitude** Decimal degrees, usually to six significant digits. Negative is West, positive is East.
**Altitude** In feet.
**Timezone** Hours offset from UTC. Fractional hours are expressed as decimals, eg. India is 5.5.
**DST** Daylight savings time. One of E (Europe), A (US/Canada), S (South America), O (Australia), Z (New Zealand), N (None) or U (Unknown). See also: Help: Time
**Tz database time zone** Timezone in "tz" (Olson) format, eg. "America/Los_Angeles".
**Type** Type of the airport. Value "airport" for air terminals, "station" for train stations, "port" for ferry terminals and "unknown" if not known. In airports.csv, only type=airport is included.
**Source** Source of this data. "OurAirports" for data sourced from OurAirports, "Legacy" for old data not matched to OurAirports (mostly DAFIF), "User" for unverified user contributions. In airports.csv, only source=OurAirports is included.

---

(a) (5) Consider the following terms: *relation schema, attribute, attribute domain, relation instance.* Explain these terms using the above Airport database. Give one small (4-5 tuples) instance of the Airport table.

(b) (10) There are three databases in the OpenFlight dataset: Airport, Airline, and Route. Give the schema of these three databases and mark the primary keys, foreign keys and provide examples of functional dependencies you identified over the three tables. [You may draw a diagram to illustrate the schema, PKs, FKs and FDs]

(c) [FD inferencing] (10)

**Recall Armstrong's axioms.**

1. **Reflexivity rule: if Y ⊆ X then X → Y**
2. **Augmentation rule: if X → Y then XZ → YZ**
3. **Transitivity rule: if X → Y and Y → Z then X → Z**

Give two examples for using Armstrong's inference rules to induce new FDs from the set of FDs you designed in question 2 (b).

(d) [Normalization] (5) Given a relation R($A_1$, $A_2$, $A_3$, $A_4$), with the following functional dependencies FDs $A_2 \rightarrow A_4$ ; $A_3 \rightarrow A_1$; $A_3 \rightarrow A_2$; $A_3 \rightarrow A_4$ Provide the 3NF form of the schema and explain why.

3. [Relational Algebra] (20) Consider the following database schema:
   **Movies** (Title, Director, Actor);
   **Location** (Theater, Address, Phone number);
   **Schedule** (Theater, Title, Time).

   Express the following queries in relational algebra (select σ, project ∏ , Cartesian product X, join (theta-join))

   -Q1: which theaters feature "Zootopia"?

   -Q2: List the names and address of theaters featuring a film directed by Steven Spielberg.

   -Q3: What are the address and phone number of the Le Champo theater?

   -Q4: List pairs of actors that acted in the same movie. (* you want to use renaming on Movies and join the Movies with its copy Movie').

4. **[Join Operators]** (40) This sets of questions test the understanding of basic database search operators. Consider a join $\bowtie_{R.A=S.B}$. We ignore the cost of output the result, and measure the cost with the number of I/Os. Given the information about relations to be joined below:

   Relation $S$ contains 20,000 tuples and has 10 tuples per block. Relation $R$ contains 100,000 tuples and has 10 tuples per block. Attribute $B$ is the primary key of $S$. In total, 52 blocks are available in memory. Assume neither relation has any index.

   a. (10) Describe a block nested join algorithm, Give the cost of joining $R$ and $S$ with a block nested loops join.
   b. (15) Describe a sort-merge join algorithm. Give the cost of joining $R$ and $S$ with a sort-merge join.
   c. (15) Describe a hash-join algorithm. Give the cost of joining $R$ and $S$ with a hash join.