**4.2 a & b**

a) Compare Snowflake schema, fact constellation, and starnet query model: Snowflake and Star schemas are similar. They both consist of a large fact table and then sets of smaller dimensions tables. In the Star schema, each dimension is represented by a single table. In the Snowflake schema, some dimensions tables are normalized, splitting the data into additional tables. This reduces redundancies, but may also decrease efficiency by increasing the number of joins necessary to execute queries.

Fact constellation (or galaxy schema) is the most elaborate multi-dimensional model of the three. In this model, there are multiple fact tables (as opposed to one in the star and snowflake schemas) which are joined together by dimensions tables. Because fact constellations are able to model many distinct but related subjects, they are more likely to be used in data warehouses than the other models.

A star-net query model is a conceptual model. It is not implemented to actually organize data like schema. In a star-net query model, there is a central point from which “arms” radiate. Each arm represents a concept hierarchy for one of the dimensions.

b) Compare Data cleaning, data transformation, refresh: data cleaning, transformation, and refreshing are back end tools and utilities that are used to keep data warehouses up to date with data from its sources (operational databases across the organization). These processes occur in the order above (clean, transform, refresh) and are repeated often. Data cleaning refers to the process of correcting errors in the data collected from operational databases and other sources. Data transformation refers to the process of integrating data from operational databases and other sources so that all data is in the format used by the data warehouse. Refresh refers to the process of propagating the data from the sources to the data warehouse.

**4.3 a, b, & c**

Suppose that a data warehouse consists of the three dimensions time, doctor, and patient,

and the two measures count and charge, where charge is the fee that a doctor charges a

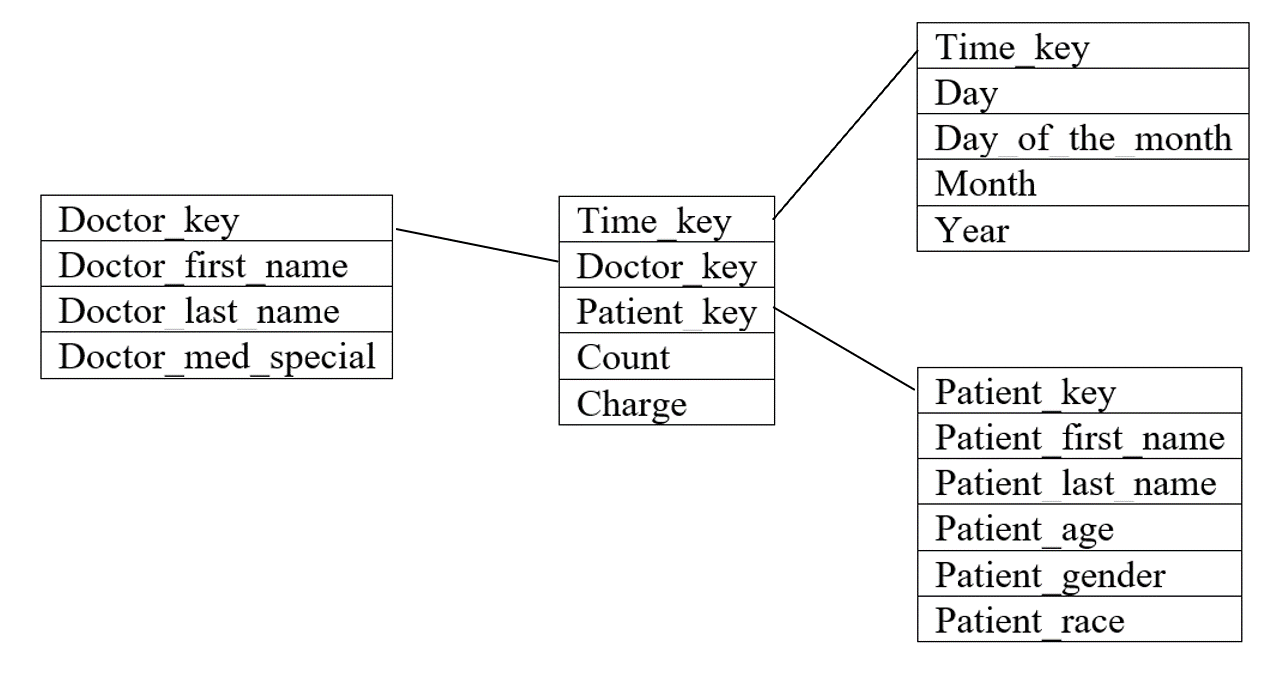
patient for a visit.

a) Enumerate three classes of schemas that are popularly used for modeling data warehouses.

Snowflake, star, and fact constellations/galaxy.

b) Draw a schema diagram for the above data warehouse using one of the schema classes listed in (a).

See star schema below.



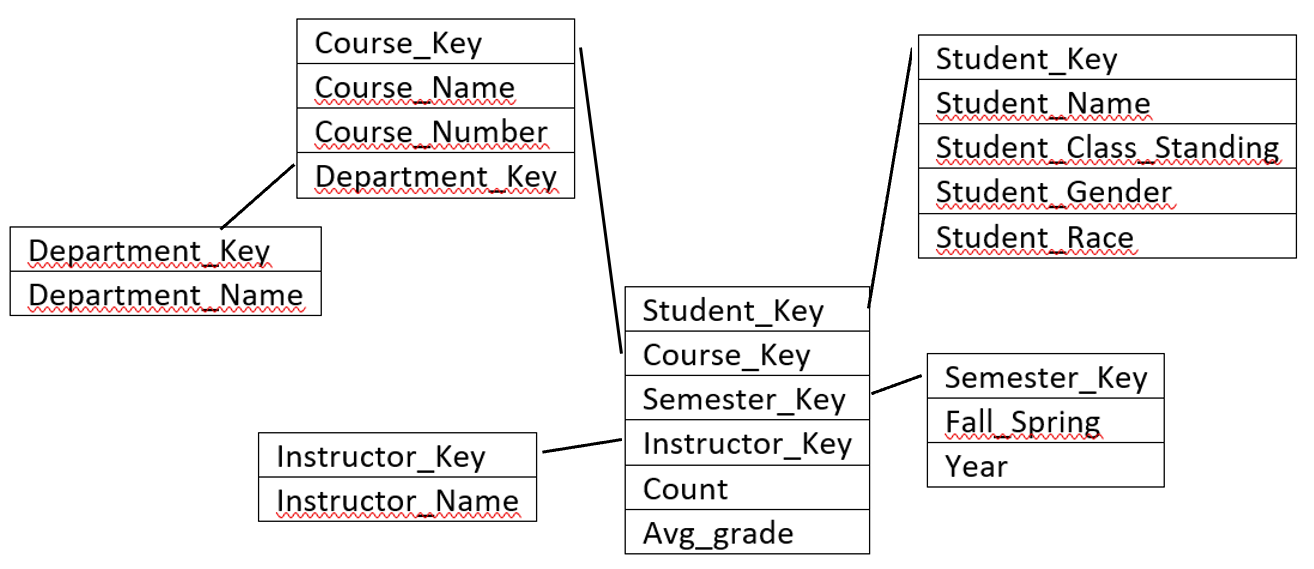
c) Starting with the base cuboid [day, doctor, patient], what specific OLAP operations should be performed in order to list the total fee collected by each doctor in 2010?

You should perform a roll up procedure. You would “roll up” on time, aggregating days to a year. You would also be performing a slice on time, cutting out one specific year.

**4.4**

Suppose that a data warehouse for Big University consists of the four dimensions student, course, semester, and instructor, and two measures count and avg grade. At the lowest conceptual level (e.g., for a given student, course, semester, and instructor combination), the avg grade measure stores the actual course grade of the student. At higher conceptual levels, avg grade stores the average grade for the given combination.

(a) Draw a snowflake schema diagram for the data warehouse.



(b) Starting with the base cuboid [student, course, semester, instructor], what specific OLAP operations (e.g., roll-up from semester to year) should you perform in order to list the average grade of CS courses for each Big University student.

Slice for CS courses on course dimension (only include CS). Collapse semester and instructor keys.

(c) If each dimension has five levels (including all), such as “student < major < status < university < all”, how many cuboids will this cube contain (including the base and apex cuboids)?

If we were dealing with a 4-dimensional data cube whose dimensions had no hierarchies, we could use 2^4 = 16 (the power law) to find this answer.  
 However, our dimensions have five levels each, so we use this formula to calculate the number of cuboids:



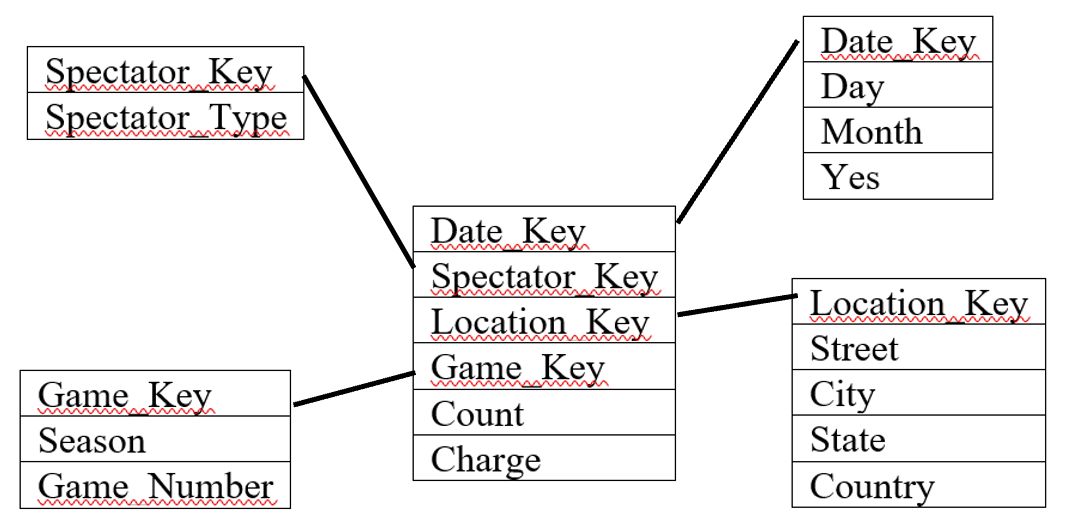
In our case, we have five levels, so L­i = 5 for each dimension we have (and we have 4 dimensions, so n=4).

(5 + 1) (5 + 1) (5 + 1) (5 + 1) = 1,296

**4.5**

Suppose that a data warehouse consists of the four dimensions date, spectator, location, and game, and the two measures count and charge, where charge is the fare that a spectator pays when watching a game on a given date. Spectators may be students, adults, or seniors, with each category having its own charge rate.

1. Draw a star schema diagram for the data warehouse.



(b) Starting with the base cuboid [date, spectator, location, game], what specific OLAP operations should you perform in order to list the total charge paid by student spectators at GM Place in 2010?

Roll up date to year. Dice to include only students on spectator dimension, only 2010 on date dimension, and only GM Place on location dimension. Collapse game dimension.

(c) Bitmap indexing is useful in data warehousing. Taking this cube as an example, briefly discuss advantages and problems of using a bitmap index structure.

Bitmap indexing creates a bit vector of zeros and ones for each distinct value on each dimension. Bitmap indexes save space compared to the size of the original table. Bitmap indexes are also useful when running queries with multiple dimension specifications, such as above where we specify that we only want students at GM Place in 2010. Bitmap indexing is not the most efficient for domains with many distinct values. If this data cube has many distinct values, then bitmap indexing is not suitable.