**Assignment-2 Data Warehouse**

**1.** Suppose that a data warehouse for a Hospital chain consists of the five dimensions’ ***date,***

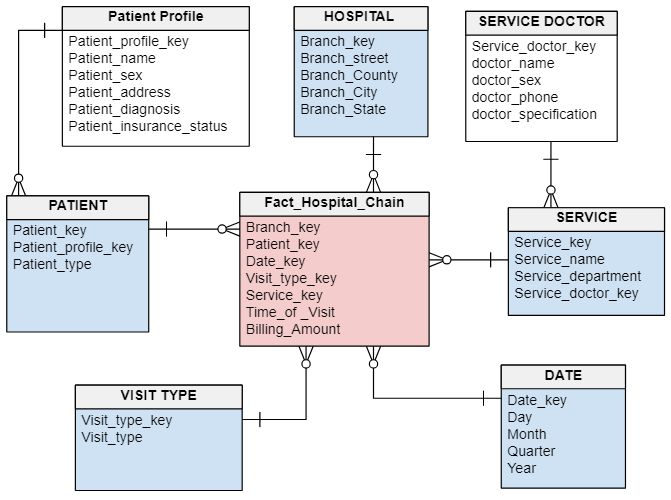
***Hospital branch, visit type, services*, *patient*,** and the two measures ***time of visit* and *billing***

***amount***, where billing amount is the amount that a patient pays for a particular visit starting

on a given admission date. Patients may be children, adults or elderly. Hospitals are in US and

the visit type includes Emergency, Consulting or Admission

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



b) Starting with the base cuboid, what specific OLAP operations should one perform in order

to list the total billing amount of children patients in the Charlotte branch in 2015 for all

Emergency type of visits.

The 5-D cuboid which holds the lowest level of summarization is called the base cuboid. Starting with base cuboid [***date, Hospital branch, visit type, services*, *patient***].

1. Dice on ***date, Hospital branch, visit type,* *patient***
2. Roll up on ***Hospital branch*** from branch\_key to branch \_city
3. Roll up on ***date*** from date\_key to year
4. Roll up on ***patient*** from patient \_key to patient\_type
5. Roll up on ***visit type*** from visit\_key to visit\_type
6. Dice for (branch\_city=” Charlotte”, patient\_type=” Children”, visit\_type=” Emergency”, year=”2015”)

c) If each dimension has four levels (excluding all), how many cuboids will this contain

(including the base and apex cuboids)?

****

**The cube will contain = (4+1+1) (4+1+1) (4+1+1) (4+1+1) (4+1+1) =7776 cuboids**

**2.** Suppose that a data warehouse contains 20 dimensions, each with about five levels

of granularity.

a) Users are mainly interested in four particular dimensions, each having three frequently

accessed levels for rolling up and drilling down. How would you design a data cube?

structure to efficiently support this preference?

**ANSWER:** To efficiently support that we use partial materialization of data cubes that offers an interesting trade-off between storage space and response time for OLAP. Instead of computing the full cube, we can compute only a subset of the data cube’s cuboids, or sub cubes consisting of subsets of cells from the various cuboids. When answering a query, instead of fetching the data from the base cuboid and performing aggregation on it, the cuboid corresponding to the query can be calculated from the closest materialized superset cuboid. Therefore, the subset of cuboids to materialize is picked so as to minimize the time needed for the expected query workload, while requiring no more than a given amount of storage. Thus, thin cube shell materialization is a partial materialization where only the base cuboid and certain low-dimensional (most highly aggregated) cuboids are stored. This gives users the freedom to dynamically focus on any subset of interesting dimensions. Users can interactively drill down or roll up to varying abstraction levels to find classification models.

b) At times, a user may want to drill through the cube, down to the raw data for one or two

particular dimensions. How would you support this feature?

**ANSWER:** If the user wants to drill through the cube, down to the raw for one or two

particular dimensions. This can be done by computing the required cuboids on the fly. In case this feature frequently implemented required time to calculate aggregates on that one or two dimensions on the fly can be rational. Another common strategy is to materialize a *shell cube*. This involves precomputing the cuboids for only a small number of dimensions (such as 3 to 5) of a data cube.