**Week 2 Assignment - Data Acquisition**

Group: Group 6 (Michael Adriel Darmawan & James Kumala) **Week 1 Revision (Conceptual Design)**

Diagram

Description automatically generated

1. What are the best (review, useful, funny & cool) 10 restaurants of the last 3 years (2014, 2015 & 2016) in Last Vegas?  
     
   For this question, we identified 4 points. They are the reviews, the restaurants, the time, and the city. So, for the conceptual data warehouse design, first we will create a category dimension table. This dimension table will be used to store the category information, as well as the business\_id as foreign key. The second dimension table is business table, which stores the information about the city, as well as the business name. The last dimension table is date dimension table, which stores the date time information for each review. The fact table in this solution is review table. This will contain the useful field, cool field, and funny field (used to determine the useful, funny, cool assessment). This table will also contain the foreign keys for business table and date table. In the end, this resulted in a snowflake schema.

Diagram

Description automatically generated  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Which businesses has the most reviews with the least users in Edinburgh?  
     
   In question number 2, we identify 4 points, which is the businesses, the reviews, the users, and the city. We decided to create a data warehouse schema with 2 dimension tables (business table and user table) and 1 fact table (review table). For the business table, we will use it to store the name and city information. Whereas the review table will be used to store the business\_id foreign key and review count to know the number of users for that business. As for the user table, it will store the foreign key of business id and the user count for that business id

Diagram

Description automatically generated\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. What kind of weather (first selected type) was it on the top 10 days of most reviews in Las Vegas?  
     
   In this question, we can identify the weather, the time (top 10 days), the reviews, and the city. First, we need the date dimension tables. This date dimension table will be used to get the top 10 days where the most reviews took place. Then we have the review fact table to store the weather id and date id as foreign keys, as well as the review count. For the weather dimension table, we will need it to know the weather type and the city.

Diagram

Description automatically generated  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. What is the current balance of each business of the top 10 most tip counts (in 2012) in Edinburgh?  
     
   In this question, we can identify balance, business, tip, and city. Therefore, we will need finance and tip tables. For the finance table, we will use it to store the balance of a business, which is linked through the finance id foreign key in business table. In tip fact table, it will be used to store the business id and date id foreign keys, as well as the tip counts. Last but not least, we also need the date dimension table.

Diagram

Description automatically generated

**Week 2 (Technical Design)**

**Data warehouse schema**

Diagram

Description automatically generated

**Technical Design Elements**

1. **Calculating the dimensions**

User Dimension Table:

Graphical user interface, application

Description automatically generated

Equal to the number of businesses, because we only need to count the number of users per business = 174567 rows

Date Dimension Table:  
  
  
 365 rows for 1 year

Weather Dimension Table:  
  
 To know the number of rows in weather table requires the calculation of both city and date data. We can get the number of city from business table in yelp database.

Graphical user interface, application

Description automatically generated

date = 365 rows in 1 year  
 city = 1044 rows  
  
 Thus,   
  
 dates\_in\_a\_year \* number\_of\_city = 365 \* 1044 = 381060 rows

Category Dimension Table:

Graphical user interface, text, application

Description automatically generated

667527 rows

Business Dimension Table:

Graphical user interface, application

Description automatically generated

174567 rows

Finance Dimension Table:

Equal with the number of businesses = 174567 rows

1. **Calculating the base level of fact records**

Review Score Fact Table

Graphical user interface, application

Description automatically generated

Number of Reviews = 5261669 rows  
Number of Years = 14 rows  
Number of Businesses = 174565 rows  
  
Number of Reviews per Business in a Year = Number of Reviews / (Number of Businesses \* Number of Years)  
 = 5261669 / 174565 \* 14  
 = 2 reviews per business in a year

In a year, this table will grow:  
  
Yearly Growth = 2 \* Number of Businesses = 2 \* 174565 = 349134 rows  
Growth after 5 Years = 5 \* Yearly Growth = 1745670 rows

Review Count Fact Table  
  
Equal to the Number of businesses = 174567 rows  
  
  
Tip Fact Table  
  
Equal to the Number of businesses = 174567 rows  
  
  
Review Weather Fact Table  
  
Equal to Number of Days in a Year, because this table is used to count the amount of review per day.  
365 rows

1. Calculate the maximum fact table size growth:

Review Score Fact Table  
Size of a single row:  
   
 Number of foreign keys = 2 (business\_id INT & date INT)  
 Number of degenerate dimensions = 3 (useful INT, cool INT, and funny INT)  
 Number of measures = 3 (INT)  
  
With the assumption that an INTEGER column takes 4 bytes, the size of a single row is:  
(2 + 3 + 3) \* 4 = 32 bytes  
  
  
Review Count Fact Table  
Size of a single row:  
  
 Number of foreign keys = 1 (INT)  
 Number of degenerate dimensions = 1 (INT)  
 Number of measures = 1 (INT)  
  
With the assumption that an INTEGER column takes 4 bytes, the size of a single row is:  
(1 + 1 + 1) \* 4 = 12 bytes

Tip Fact Table  
Size of a single row:  
  
 Number of foreign keys = 2 (INT)  
 Number of degenerate dimensions = 1 (INT)

Number of measures = 1 (INT)  
  
With the assumption that an INTEGER column takes 4 bytes, the size of a single row is:  
(2 + 1 + 1) \* 4 = 16 bytes  
  
  
Review Weather Fact Table  
Size of a single row:  
  
 Number of foreign keys = 2 (INT)  
 Number of degenerate dimensions = 1 (INT)

Number of measures = 1 (INT)  
  
With the assumption that an INTEGER column takes 4 bytes, the size of a single row is:  
(2 + 1 + 1) \* 4 = 16 bytes  
  
  
Estimated Growth Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table name | Current number of records | Current size | Estimated growth per year in rows | Estimated growth per year in size |
| user | 174567 | 169.9 MB | 6750 | 176.469 MB |
| date | 5,261,669 | 3.6 GB | 365 | 3.62 GB |
| weather | 5,261,669 | 3.6 GB | 5150 | 3.603 GB |
| category | 665181 | 41.6 MB | 8220 | 42.11 MB |
| business | 174567 | 23.6 MB | 4635 | 24.22 MB |
| finance | 174567 | 23.6 MB | 4635 | 24.22 MB |
| review\_score | 5,261,669 | 3.6 GB | 378150 | 3.85 GB |
| review\_count | 5,261,669 | 3.6 GB | 378150 | 3.85 GB |
| review\_weather | 5,261,669 | 3.6 GB | 378150 | 3.85 GB |
| tip | 174567 | 23.6 MB | 2102 | 23.8 MB |
| Total | 27671794 | 18.28 GB | 1166307 | 19.06 GB |

1. **Page Size**

Diagram

Description automatically generated

1. **Hardware and software**

Calculating amount of CPU Cores  
  
-> Amount of data (in MB) is retrieved by an average query = 5.3 MB  
-> How many concurrent users = 500  
-> What is the required response time for a query = 1 second  
  
Amount of cores = ((5.3 MB / 200 MB/s) \* 500) / 1  
 = 13.25   
 = 13 cores (for data warehouse that could be used for at least 5 years)

Table

Description automatically generated

1. **Performance features**

* Clustered Indexes vs Non-Clustered Indexes

For this problem, a clustered indexes suit better than non-clustered indexes. This is because using clustered indexes is generally much faster in accessing the data compared to that of non-clustered indexes. Clustered indexes also require less memory for operations.

* Row store vs Column store

For this problem, a row store index will suit better. This is because we don’t have to do any aggregations operations anymore. We also have very small number of rows for each table. Lastly, we might also need to access the complete record of the data, so row store is better than column store for this problem.

* Compression

For this problem, we would not opt for compression, because we want to have a quick access of the data, while compression requires more CPU usage.

* Aggregation tables

For this problem, we will create aggregation tables, like review\_count fact table. Because in running a query that can produce the review\_count result requires a “group by” statement, we will perform them only at night so that it doesn’t disturb the performance during the day.

* Vertical partitioning vs Horizontal Partitioning

For this problem, we choose vertical partitioning. This is because by dividing bigger tables into smaller tables, the rarely used columns will not slow down the performance, hence increasing the speed of frequently used queries.