In [ ]:
```
%load_ext sql
%sql sqlite:///olap.db
```

OLAP and Cubes Activity

1 Data and Motivation

You're given a bunch of data on search queries by users. (We can pretend that these users are Google search users and you are an engineer on the Google Web Search team). You want to analyze the number of search queries made, who they are made by, and how successful your search engine is at returning a result that people want to click on. A particular user can only make one query at a time.

Below is a table called raw_search_log containing details of search queries.

raw_search_log

Field	Type	Description
user_id	INTEGER	ID of the user that made the search.
timestamp	TIMESTAMP	Time at which the search occurred. A particular user_id can only make a single query at a given timestamp.
query	VARCHAR(500)	Search query (text typed into the search bar).
rank	INTEGER	The rank of the search result they clicked on (how high it appears in the results). This is NULL if they never clicked on a result.
click_url	VARCHAR(200)	The URL of the search result they clicked on. This is NULL if they never clicked on a result.
city	VARCHAR(50)	The location of the user (is constant for a given user).
age	INTEGER	The age of the user (is constant for a given user).

2 OLAP Database Design

We will split up the attributes of the raw data schema into a star schema with 2 dimension tables (users_dim, dates_dim), and a fact table (searches_fact). Design the star schema.

users_dim

</table>

User information.

searches_fact

user_id	INTEGER
city	VARCHAR(50)
age	INTEGER

</table>

Search queries that were performed (who, when, what).

dates_dim

user_id	INTEGER
timestamp	TIMESTAMP
query	VARCHAR(500)
rank	INTEGER
click_url	VARCHAR(200)

</table>

Date/time information on search queries.

1. Write the CREATE TABLE statement for users_dim. Don't forget about the primary key!

```
In [ ]:
```

2. Write the CREATE TABLE statement for dates_dim. This table contains attributes not found in the raw schema because we want to be able to do more detailed analysis on the data. Don't forget about the primary key!

```
In [ ]:
```

3. Write the CREATE TABLE statement for searches_fact. Don't forget about foreign keys into the dimension tables and the primary key. (If you forgot the FOREIGN KEY syntax, take a look at Lecture 5: Design Theory3 from the course website).

```
In [ ]:
```

Discussion Why did we design it this way? What kind of queries might we be able to do with this schema design?

3 Populating Tables

After designing the tables, we need to populate them with data from raw_search_log in order to do our analysis!

Example. Populate the users_dim table with an INSERT...SELECT statement. [18250 rows]

In [ ]:

1. Populate the dates_dim table with an INSERT...SELECT statement. [28138 rows] </p>
    - You must extract the date (using the DATE function on the timestamp field) and hour of day (using the EXTRACT function) from timestamp. The documentation for EXTRACT is here4 and the syntax is EXTRACT(FROM timestamp).
    - You must figure out whether or not the date is a weekend. Look at the EXTRACT function again. You will want to use a CASE statement.

In [ ]:

2. Write an INSERT...SELECT statement to populate the searches_fact table. [28195 rows]

In [ ]:

Discussion Think about how much work was required to set up the data for analysis. Compare this to NoSQL/Redis. How much design work did you have to do before analyzing the data?

What if you wanted to add a field source to denote where the user queried from (desktop, Android browser, Safari, etc.)? What would you have to do in SQL versus what you do in Redis? What if source were unknown? How would this be denoted in SQL versus Redis?

4 OLAP Queries

Now, we want to use our dimension and fact tables to analyze the data. We don't want to use raw_search_log as that would defeat the purpose of this exercise. NATURAL JOINs will come in handy and are easy to use by design of the schema.

Example. Find the number of queries performed by people between the ages of 18 and 25, ordered by age.

In [ ]:

Problems

1. How many characters long are the search queries done by people at Stanford? Find the top 10 longest queries and their length for queries done by users in Stanford. LENGTH (documentation here6) and LIMIT (documentation here7) will be helpful. [Longest query is 255 characters]

In [ ]:

2. We want to get summary statistics on queries in each city. For each city, get the number of unique users and number of queries made. The result's schema should be (city, num_users, num_queries).

In [ ]:

3. Find the maximum length of search queries each day (not timestamp!) that returned a click_url (i.e. searches that resulted in a user clicking on a search result). We're interested in seeing if the search engine is getting better at returning results for queries, or if people are better at searching with shorter queries. The result should be of the form (date, max_query_length).

In [ ]:

4. Find the bad kids! Write a query that returns the user_ids of users under the age of 18 who've made searches between the hours of 2 am and 7 am (hour values of 2 and 7), inclusive. [176 rows]

In [ ]:

5 Discussion

Think about these questions below and we will discuss it together as a class.

- Installation and setup-wise, what was easier to do in Redis?
- What kind of analysis was easier to do in SQL?
- What are some of the disadvantages or limitations of using SQL rather than Redis?

6 Bonus Problems (Extra Practice)

Do these if you have extra time! This is not required as part of the activity.

1. Report number of queries and clicked websites viewed by each user in Palo Alto. The result should be of the form (user_id, num_queried, num_clicked). [First row of results is (281371, 2, 2)]

In [ ]:

2. For dates between '2006-03-04' and '2006-03-07', find the number of queries. Your result should be of the form (date, num_queries). Remember, some queries could have occurred at the same time! [335, 400, 348, 369]

In [ ]:

3. Are older people less effective at making a good search? Get the number of queries that did not return a click_url from users 50 or older who don't live in Oldsville. [2430]

In [ ]:

4. Super-Bonus! It would be interesting to see if the average queries per hour is different between days during the week and days during the weekend. Do people make more searches during the weekend? Do they make more at night? Write a query that computes the number of queries per hour, averaged over the weekdays and over the weekends. The result should have rows and columns like this:

timestamp	TIMESTAMP
date	DATE
hour	INTEGER
is_weekend	BOOLEAN

hour	avg_weekday_queries	avg_weekend_queries
0	11.89	10.69
1	7.58	9.27
...	...	...

avg_weekday_queries contains the number of queries per hour, averaged over all weekdays (the is_weekend field in dates_dim will be helpful), and avg_weekend_queries contains the number of queries per hour, averaged over all weekend days.

Hint: Break it apart into two queries (one for the weekend, one for the weekday) and join them together at the end.

In [ ]:
```
%%sql
```