Lab Session 10: Data Warehouses

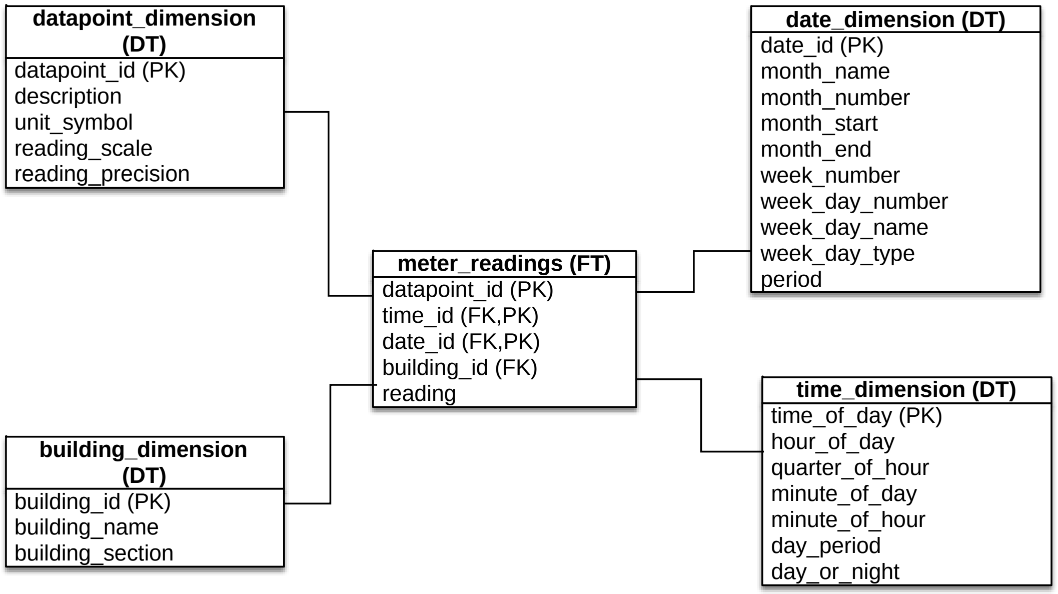
In this lab we will be working with a data warehouse of energy consumption in a university campus.

The university campus has 7 buildings (the building names are: A, B, C, D, E, F, G). Each building has multiple sections:

- Building A has 3 sections (A.1, A.2, A.3)
- Building B has 3 sections (B.1, B.2, B.3)
- Building C has 3 sections (C1, C2, C3)
- Building D has 1 section (Main)
- Building E has 2 sections (E.1, E.2)
- Building F has 1 section (Main)
- Building G has 3 sections (G.1, G.2, G.3)

There are 16 sections overall. Each section has 3 energy meters installed at different locations. These are called *data points*. There are 16 × 3 = 48 data points of interest.

Each data point provides a new reading every 15 minutes. A reading is an instantaneous measurement of energy consumption in mWh. The readings are stored in a data warehouse with the following star schema:



In this star schema, the fact table (FT) is *meter_readings* and the dimension tables (DT) are: *building_dimension*, *datapoint_dimension*, *date_dimension*, and *time_dimension*.

Note that in the *building_dimension* table, the primary key is *building_id*. This is a numeric value that uniquely identifies each building section in the university campus.

In a similar way, in the *datapoint_dimension* table, the primary key is *datapoint_id*. This is a numeric value that uniquely identifies each energy meter in the university campus.

Loading the data warehouse

The file **energy_data.sql** contains the SQL instructions to create the tables above and to load them with data. (To keep the amount of data within a manageable size, in this lab we will be working with the readings collected during the month of December 2013 only.)

1. Use an SFTP client (such as WinSCP or FileZilla) to transfer the **energy_data.sql** file to your home directory in **sigma.ist.utl.pt**.

2. Open an SSH connection to **sigma.ist.utl.pt**.
   On Windows, you can use PuTTY. On Mac OS or Linux, run the following command in a terminal: **ssh istxxxxxx@sigma.ist.utl.pt** where istxxxxxx is your username.

3. If you still remember your MySQL password, go to the next step. Otherwise, if you have forgotten your MySQL password, execute: **mysql_reset**

4. Connect to the MySQL server with the command:
   mysql -h db.ist.utl.pt -u istxxxxxx -p
   where istxxxxxx is your username.
   When prompted, enter your MySQL password (the password given by **mysql_reset**).

5. Once you are connected to MySQL, execute the command:
   source energy_data.sql
   to create the data warehouse.
   Note: since there is a lot of data, this may take a few minutes to complete.

Inspecting the data warehouse

The following queries are intended to get familiar with the data warehouse.

6. Execute the following query to see the university buildings and their sections:
   SELECT *

```
FROM building_dimension
ORDER BY building_id;
```

7. Execute the following query to see all the energy meters installed throughout the university campus:

   ```
   SELECT *
   FROM datapoint_dimension
   ORDER BY datapoint_id;
   ```

8. You will notice that there is a total of 108 data points, many more than the 48 data points that we are interested in. However, with the following query you can check that only the readings from 48 data points have been collected:

   ```
   SELECT DISTINCT datapoint_id
   FROM datapoint_dimension NATURAL JOIN meter_readings;
   ```

9. Execute the following command to see all the columns in the date dimension:

   ```
   DESCRIBE date_dimension;
   ```

10. Execute the following query to see the kind of data that is stored in that table:

    ```
    SELECT date_id, date_year, month_name, month_day_number, week_number,
    week_day_name, period
    FROM date_dimension;
    ```

11. You will see that the table has all the dates for 2013. However, with the following query you can confirm that only the readings for December 2013 have been collected:

    ```
    SELECT DISTINCT date_id FROM meter_readings;
    ```

12. Finally, check the time dimension:

    ```
    SELECT *
    FROM time_dimension
    LIMIT 200;
    ```

    This will show the only first 200 rows of the table. You will notice that there are some jumps in time, of about 15 minutes. This is because the meter readings are collected in intervals of 15 minutes.

13. Also, run the following query to see the kind of information that is stored in the *day_period* and *day_night* columns:

    ```
    SELECT DISTINCT day_period, day_night
    FROM time_dimension;
    ```

14. You can access this information for each reading by joining the tables *meter_readings* and *time_dimension*:

```
SELECT datapoint_id, time_id, date_id, building_id, reading, day_period, day_night
FROM meter_readings NATURAL JOIN time_dimension
LIMIT 1000;
```

Querying the data warehouse

We now get to the main goal of this lab: to explore the data stored in the data warehouse. For this purpose, have in mind the following principles:

- When we want to analyze the data according to multiple dimensions, we have to join the fact table (*meter_readings*) with the corresponding dimension tables. You can use **NATURAL JOIN** for this purpose.
- In most cases, we will be interested in average readings (e.g average energy consumption by building, or average energy consumption by day). For this purpose, you can use the **AVG** aggregate function.
- When using multiple dimensions to analyze the data (e.g. average reading per building *and* per day) you have to **GROUP BY** all those dimensions.
- In some queries we will be experimenting the option **WITH ROLLUP** to aggregate the overall results (MySQL only).

You are now asked to carry out the following analysis on your own:

15. Write a query to compute the average energy consumption by week day (use *week_day_name*). Based on these results, identify the weekdays of highest and lowest consumption.
    *Note: After the **GROUP BY** include **WITH ROLLUP** to aggregate the overall results.*

16. Write a query to determine the average consumption by week, but only during the last three weeks of the year (use *week_number*). Based on these results, you can confirm that energy consumption decreases in the last weeks of December.
    *Note: After the **GROUP BY** include **WITH ROLLUP** to aggregate the overall results.*

17. Write a query to determine the average consumption by building name and by week, but only during the last three weeks of the year.
    *Note: After the **GROUP BY** include **WITH ROLLUP** to aggregate the overall results.*

18. Write a query to determine the average consumption by building name. Use an **ORDER BY** in the query, so that the building that consumes the most energy appears at the top.

19. Write a query to determine the average consumption by building name and by day period (in the time dimension). The results should be sorted by building name in

alphabetical order and by average consumption in decreasing order. Is it true that all buildings have the highest consumption in the afternoon?

20. Change the previous query to include, besides the day period, also the hour of day.
    Hint: You have to change the query in two places.
    This is a drill-down into a more fine-grained level of detail.

21. Check the results of the previous query. It appears that building C now has its highest consumption in the morning. Why? Isn’t this inconsistent with the previous results?
    Hint: the results have been generated at different levels of detail.