## Data Mart Design on TOP of an OLTP Database

The practice purpose is to build a data mart in R/MySQL on top of an OLTP system.

# Problem specification

A telephone company is interested in analyzing its own data to improve customer services. At present, the company has a database with call logs. For each call, the caller and receiver phone numbers, the duration, the type of charge (e.g., peak, off-peak rates), the start time (date, hour, minute, second) are known. The OLTP database of the telephone company is reported in Table 1.

The managers wish to obtain very fast the information about the telephone traffic on the company lines and the daily income based on the caller location, the day and the phone rate.

In particular, the managers want to analyze the following events:

1. Monthly income and number of calls for each caller city.
2. Monthly income and number of calls for each caller province.

**Tutorial Tasks**

1. Create a conceptual model (UML class Diagram) for the relational design in Table-1;
2. Create a conceptual model (UML class Diagram) for the data mart design in Table-2;
3. Compare the output conceptual designs of task-1 and task-2;
4. Identify the main Business Process (fact ) and its dimensions;
5. Create the corresponding data mart tables (fact and dimension tables, Table-2) in MySQL;
6. Unzip the given “DWHPhones.Zip” sample data mart data and import them to the data mart tables (fact and dimension tables) created already in task-3. You may advise the following link;

<https://dev.mysql.com/doc/workbench/en/wb-admin-export-import.html>

1. Connect to MySQL through R , revisit week’s one tutorial and import the data mart data to R;

|  |  |
| --- | --- |
| **Tables** | **Description** |
| **PHONERATES**  (  phoneRateType INT NOT NULL,  phoneRateName VARCHAR(20) NOT NULL, phoneRate\_CostPerSecond FLOAT NOT NULL, PRIMARY KEY(phoneRateType)  ); | Different phone rates  8 rows |
| **PLACES**  (  Places\_ID INT NOT NULL,  City VARCHAR(20) NOT NULL,  Province VARCHAR(20) NOT NULL,  Region VARCHAR(20) NOT NULL, PRIMARY KEY(Places\_ID)  ); | Places: 2000 rows |
| **CALLS**  (  CallerPhoneNumber VARCHAR(20) NOT NULL,  ReceiverPhoneNumber VARCHAR(20) NOT NULL,  CallerLocation INT NOT NULL,  ReceiverLocation INT NOT NULL,  FullDate DATE NOT NULL,  StartTimeHour INT NOT NULL,  StartTimeMinute INT NOT NULL,  StartTimeSecond INT NOT NULL,  CallDuration FLOAT NOT NULL,  phoneRateType INT NOTNULL,  PRIMARY KEY(CallerPhoneNumber, ReceiverPhoneNumber, FullDate, StartTimeHour  , StartTimeMinute, StartTimeSecond),  FOREIGN KEY(phoneRateType) REFERENCES PhoneRates(phoneRateType) ON DELETE  CASCADE,  FOREIGN KEY(CallerLocation)REFERENCES Places(Places\_ID) ON DELETE CASCADE,  FOREIGN KEY(ReceiverLocation) REFERENCES Places(Places\_ID) ON DELETE CASCADE  ); | Calls in 2007 and 2008  ~ 1500000 rows |

## Table 1 – Source data base with single call information

|  |  |
| --- | --- |
| **Tables** | **Description** |
| **TIMEDIM**  (  ID\_time INT NOT NULL,  DayOfWeek CHAR(15) NOT NULL,  DateMonth INT NOT NULL,  DateYear INT NOT NULL,  PRIMARY KEY(ID\_time)  ); | Time dimension  10 rows |
| **PHONERATE**  (  ID\_phoneRate INTEGER NOT NULL,  phoneRateType VARCHAR(20) NOT NULL,  PRIMARY KEY(ID\_phoneRate)  ); | Phone rate dimension  7 rows |
| **LOCATION**  (  ID\_location INTEGER NOT NULL,  City VARCHAR(20) NOT NULL,  Province CHAR(20) NOT NULL,  Region CHAR(20) NOT NULL,  PRIMARY KEY(ID\_location)  ); | Place dimension  1500 rows |
| **PhoneCalls**  (  ID\_time INTEGER NOT NULL,  ID\_phoneRate INTEGER NOT NULL, ID\_location\_Caller INTEGER NOT NULL, ID\_location\_Receiver INTEGER NOT NULL,  Price FLOAT NOT NULL,  NumberOfCalls INTEGER NOT NULL,  PRIMARY KEY(ID\_time,ID\_phoneRate,ID\_location\_Caller,ID\_location\_Receiver),  FOREIGN KEY(ID\_time) REFERENCES timeDim(ID\_time),  FOREIGN KEY(ID\_phoneRate) REFERENCES phoneRate(ID\_phoneRate),  FOREIGN KEY(ID\_location\_Caller) REFERENCES location(ID\_location),  FOREIGN KEY(ID\_location\_Receiver) REFERENCES location(ID\_location)  ); | Fact table 7809 rows |

**Table 2 – Proposed solution - Data warehouse tables**