MySQL and PHP test task

1. Preparing the environment.

In order to successfully complete the exercises, the access to working **MySQL** and **PHP** environment is required. We recommend installing **WAMP** (Windows) or **LAMP** (Linux), which includes needed programs for completing the test task.

WAMP can be acquired from here: http://www.wampserver.com/en/

2. Exercise 1 - preparing MySQL database and tables.

2.1 Create a MySQL database for the test task to create the tables.

2.2 Create the following tables:

a) Table named „**patient**“ with following column names and types:

„_id“, int with length of 10, unsigned, not null, auto increment,
„pn“ varchar with field length of 11, default null;
„first“ varcahr with field length of 15, default null;
„last“ varchar with field length of 25, default null;
„dob“ with type date, default null;

b) Table named „**insurance**“ with following column names and types:
„_id“, int with length of 10, unsigned, not null, auto increment;
„patient_id“ int with length of 10, unsigned, not null, foreign key referenced to patient table „_id“ column;
„iname“, varchar with field length of 40, default null;
„from_date“ with type date, default null;
„to_date“ with type date, default null;

2.3 Populate tables with sample data – at least **5** records for patient table and **2** insurance records for each patient.

2.4 Solution file has to contain the **table creation** and **data population** MySQL queries.

3. Exercise 2 – PHP scripting.

3.1 Create a PHP script, which can be ran from **command console**.

3.2 The script has to have the following functionality:

a) Display the following columns for each patient to the console:

Patient number (**pn**), Patient last name (**last**), Patient first name (**first**), Insurance name, Insurance from date (**from_date**) in **US short date format (MM-DD-YY)** and Insurance to date (**to_date**) in US short date format (MM-DD-YY), ordered by Insurance from date starting from earliest and then patient last name.

Example output:

```
000000002, Smith, John, Medicare, 01-01-09, 01-01-10
000000002, Smith, John, Blue Cross, 06-01-09, 01-01-10
000000001, Doe, John, Medicaid, 01-01-10, 01-01-11
000000001, Doe, John, Blue Shield, 01-01-11, 01-01-12
```

NB! Ordering must handled by script (or query) ie. it can not depend on the record insertion queue.

b) Create statistics about how many times (in count and in percentages with **two decimal points** from total) each letter occurs in first and last names. This has to be considered as **case insensitive** and only considering alphabetic characters. Do not output letters, which do not occur in the strings. Output has to be sorted alphabetically ascending order.

Example output for names “ John Smith“ and „John Doe“:

```
D	1	6,25 %
E	1	6,25 %
H	3	18,75 %
I	1	6,25 %
J	2	12,5 %
M	1	6,25 %
N	2	12,5 %
O	3	18,75 %
S	1	6,25 %
T	1	6,25 %
```

NB! Separate each field of the row with tab character (like show in example above).

4. Exercise 3 – Object-oriented PHP.

4.1 Create a PHP interface called „**PatientRecord**“, which has following methods:

1) A method which declares a method for returning implementing record „_id“ property.

2) A method which declares a method for returning implementing record’s associated patient number.

4.2 Create PHP class called „**Patient**“, which implements interface called „**PatientRecord**“ and has all properties to represent the fields of the according database table plus one extra property, which represents an array of Insurance records (see section 4.3) associated with the patient and has following methods:

1) Class constructor to instantiate Patient record by given pn parameter and fill out all class properties.

2) A class method which returns Patient record „_id“ property.

3) A class method which returns Patient record „pn“ property.

4) A class method which returns Patient name in format „First Last“.

5) A class method which returns an array of patient Insurance record instances (see section 4.3).

6) A class method which accepts a date argument in **US short date format** and prints a table with following output:

„Patient Number, First Last, Insurance name, Is Valid“

Example output, when date argument is „06-02-09“:

```
000000002, John Smith, Medicare, No
000000002, John Smith, Blue Cross, Yes
```

4.3 Create PHP class called „**Insurance**“, which implements interface „**PatientRecord**“ and has following methods:

1) Class constructor to instantiate Insurance record by given id parameter and fill out all class properties.

2) A class method which returns Insurance record „_id“ property.

3) A class method which returns Insurance record „pn“ property.

4) A class method which accepts a date argument in **US short date format** and returns **true** or **false** whether the given date falls in the Insurance record **from_date** and **to_date** or not.

Example:

from_date = 2010-01-01, to_date = 2011-01-01, compare_date = 12-31-10 (format) returns true.

This functions has also deal with the situatuon if **to_date** is not defined. In that case the insurance is effective infinitely.

4.4 Create a test script to test the features of Patient and Insurance classes.

Print out all patients and their insurances, which effectiveness is compared to **today** (a date, which is the same as date when the script is ran) , having following format and ordered by patient number ascending.

Example output:

```
000000001, Doe, John, Medicaid, No
000000001, Doe, John, Blue Shield, Yes
000000002, John Smith, Medicare, No
000000002, John Smith, Blue Cross, No
```