# MySQL Introduction - Exercises

This document defines a set of tasks to be done as a part of the MySQL Introduction lecture’s exercises.

Databases are the natural way of one application to just… work. They provide a way to store an information such as application’s users, interactions and other components and the relations between them.

# Design Blog Database

Create a database named “blog”.

## Create Users Table

A user does have a **unique** **identifier** which is auto incremented for each row; a username which is a textfield, so does the password and email. Also they are **born on particular date** and have a fullname. The user is **either active or inactive.**

## Create Categories Table

In a blog there are categories where topics are stored. A category has a name and a **unique identifier**, auto incremented for each row.

## Create Topics Table

A blog topic has a **unique identifier** which is auto incremented for each row; a textual title limited to some symbols and an unlimited text as body.

# Manipulate Data

## Register Users

Insert user rows in order to achieve this result



## Add Categories

Insert two categories, namely **“**Programming**”** and **“**Life**”**

## Create Topics

Create some topics, test the length of both textual columns.

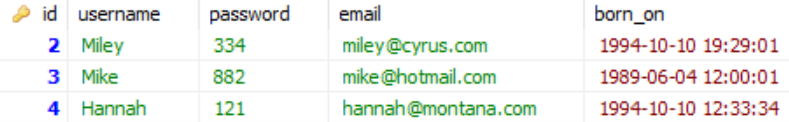
## Rename a User

Rename user ‘John’ to become **‘Johnathan’** and change its email to be **‘johnathan@mail.com’**



## Extract Users in Years Range

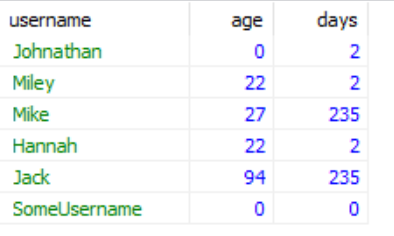
Extract all users that are born between year 1980 and 1995



You may want to use the [YEAR()](https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html#function_year) function

## Extract Users Age and Birthday

Extract users age and calculate how much days are left to their birthday



Look at the [other date/time functions](https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html) such as **DAY(), MONTH(), DATEDIFF()**

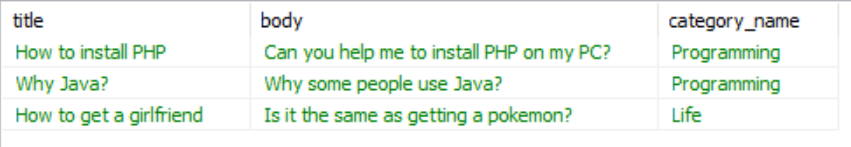
# Table Relations

## Relate Topics to Categories

Create an integer column in the topics table called category\_idwhich is referenced as a foreign key to the id column in category table

## Extract Topics Along with Categories

Extract the topics title**,** body and the category’s **name** in the resultset. Name the category’s name column “category\_name”

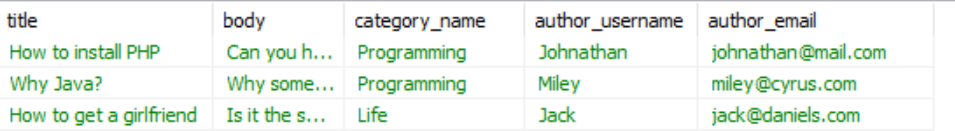


## Relate Topics to Users

Each topic should have an author. The author is one of the existing users.

## Extract Topics Along with Categories and Authors

Extract the topic’s title, body, category’s name, author’s username and author’s email.



## Count Category’s Topics

Count how many topics each category has. As a result, output two columns: category\_name and topics\_count



You might want to use [GROUP BY](http://dev.mysql.com/doc/refman/5.7/en/group-by-functions.html) statement.

## Count Category’s Topics Filtered

Count how many topics each category has, but output only the categories that has more than 1 topic. As a result, output two columns: category\_name and topics\_count

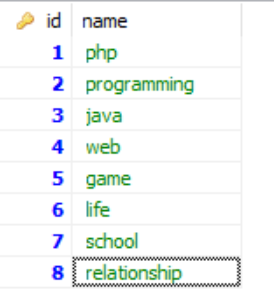
****

**WHERE** clause will not work. Use google to find the appropriate operation.

## Topic Tags

Create a new table called tags which has an id and name**.** Relate the topics to the tags, so each topic might have arbitrary tags and vice versa. Extract a resultset which outputs the topic\_titleand tag\_name

**Tags table:**



**The desired resultset:**

****

You might want to learn about [Many-to-Many relationship](https://en.wikipedia.org/wiki/Many-to-many_(data_model))