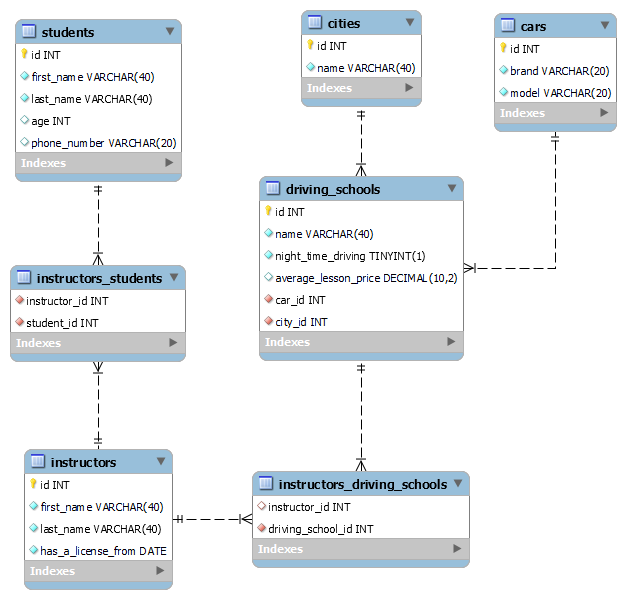
# MySQL Exam Preparation – 04 Oct 2024 Go Roadie - Driving schools in the UK

[Link: Database Basics MySQL Exam](https://judge.softuni.org/Contests/4532/Database-Basics-MySQL-Retake-Exam-7-April-2024)

*Driving schools play a crucial role in preparing individuals to become safe and responsible drivers. By offering comprehensive training programs, professional instruction, and valuable resources, these schools empower students to develop the knowledge, skills, and confidence to navigate the roadways safely and successfully obtain their driver's license.*

## Section 0: Database Overview

You have been given an Entity / Relationship Diagram of the Database:



The **go\_roadie** **Database** needs to hold information about **cities**, **driving\_schools**, **cars**, **instructors** **and students**.

Your task is to create a database called **go\_roadie**. Then you will have to create several **tables**.

* cities – contains information about the **cities**.
* driving\_schools – contains information about the driving\_schools.
  + Each driving\_school has a city, a car and instructors.
* cars – contains information about the **cars**.
* instructors – contains information about the **instructors**.
  + Each instructor has **students**.
* students – contains information about the **students**.
  + Each student has an instructor.
* instructors\_driving\_schools – a **many** to **many** **mapping** table between the **instructors** and the driving\_schools.
* instructors\_students - a **many** to **many** **mapping** table between the **instructors** and the **students**.

## Section 1: Data Definition Language (DDL) – 40 pts

Make sure you implement the whole database correctly on your local machine so that you can work with it.

The instructions you'll be given will be the minimum needed to implement the database.

### Table Design

You have been tasked to create the tables in the database by the following models:

#### cities

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer,** from **1** to **2,147,483,647.** | **Primary Key AUTO\_INCREMENT** |
| name | A **string** containing a maximum of **40 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**. UNIQUE** values. |

#### cars

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer,** from **1** to **2,147,483,647.** | **Primary Key AUTO\_INCREMENT** |
| brand | A **string** containing a maximum of **20 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**.** |
| model | A **string** containing a maximum of **20 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**. UNIQUE** values. |

#### instructors

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer,** from **1** to **2,147,483,647.** | **Primary Key AUTO\_INCREMENT** |
| first\_name | A **string** containing a maximum of **40 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted. |
| last\_name | A **string** containing a maximum of **40 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**. UNIQUE** values. |
| has\_a\_license\_from | The **DATE** when the license was acquired. | **NULL** is **NOT** permitted**.** |

#### driving\_schools

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer,** from **1** to **2,147,483,647.** | **Primary Key AUTO\_INCREMENT** |
| name | A **string** containing a maximum of **40 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**. UNIQUE** values. |
| night\_time\_driving | It can be **true** or **false**. | **NULL** is **NOT** permitted**.** |
| average\_lesson\_price | **DECIMAL**, up to **10 digits**, **2** of which are after the **decimal point**. | **NULL** is permitted**.** |
| car\_id | **Integer,** from **1** to **2,147,483,647.** | Relationship with table cars.  **NULL** is **NOT** permitted**.** |
| city\_id | **Integer,** from **1** to **2,147,483,647.** | Relationship with table cities **NULL** is **NOT** permitted**.** |

#### students

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | **Integer,** from **1** to **2,147,483,647.** | **Primary Key AUTO\_INCREMENT** |
| first\_name | A **string** containing a maximum of **40 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted. |
| last\_name | A **string** containing a maximum of **40 characters**. Unicode is **NOT** needed. | **NULL** is **NOT** permitted**. UNIQUE** values. |
| age | **Integer,** from **1** to **2,147,483,647.** | **NULL** is permitted**.** |
| phone\_number | A **string** containing a maximum of **20 characters**. Unicode is **NOT** needed. | **NULL** is permitted**.**  **UNIQUE** values. |

#### instructors\_driving\_schools

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| instructor\_id | **Integer**, from **1** to **2,147,483,647.** | Relationship with table instructors.  **NULL** is permitted. |
| driving\_school\_id | **Integer**, from **1** to **2,147,483,647.** | Relationship with table driving\_schools.  **NULL** is **NOT** permitted |

#### instructors\_students

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| instructor\_id | **Integer**, from **1** to **2,147,483,647.** | Relationship with table instructors.  **NULL** is **NOT** permitted. |
| student\_id | **Integer**, from **1** to **2,147,483,647.** | Relationship with table students.  **NULL** is **NOT** permitted |

Submit your solutions in Judge on the first task. Submit **all** SQL table creation statements.

You will also be given a data.sql file. It will contain a **dataset** with random data which you will need to **store** in your **local database**. This data will be given to you so you will not have to think of data and lose essential time in the process. The data is in the form of **INSERT** statement queries.

## Section 2: Data Manipulation Language (DML) – 30 pts

Here we need to do several manipulations in the database, like changing data, adding data, etc.

### Insert

You will have to **insert** records of data into the **students** table, based on the **students** table.

For all **students** which age is **under** **20** years (**age** < **20**), **insert data** in the **students** tablewith the **following values**:

• **first\_name** – set it to the first name of the student but **reversed** and **lowercase**.

• **last\_name** – set it to the last name of the student but **reversed** and **lowercase**.

• **age** – set it to the **age** of the **student** plus the **first digit** of his/her **phone\_number**.

• **phone\_number** – set it to **"1+"** and **phone\_number.**

### Update

Because of the inconvenience for the instructors, **increase** the **average\_lesson\_price** of each **driving** **school**, which is located in **London** and **offers** night driving,by **30**.

### Delete

**Delete** all **driving schools**, which **do** **not** offer night driving.

## Section 3: Querying – 50 pts

And now we need to do some data extraction. **Note** that the **example results** from **this section** use a **fresh database**. It is **highly recommended** that you **clear** the **database** that has been **manipulated** by the **previous problems** from the **DML** **section** and **insert again** the **dataset** you've been given, to ensure **maximum consistency** with the **examples** given in this section.

### Youngest students

Extract from the **go\_roadie** database, info (**full\_name** and **age**) about the youngest **students** whose **first** name contains the letter **'a'**.

**Order** the results by **id** of the student in **ascending** order.

#### Required Columns

* full\_name (first\_name + " " + last\_name)
* age

#### Example

|  |  |
| --- | --- |
| **full\_name** | **age** |
| Michael Pinch | 19 |
| Ashley Hill | 19 |
| Jessica White | 19 |

### Driving schools without instructors

Write a query that returns: **id, name** and **brand** (of the offered **car** for driving) from table **driving\_schools**. **Filter** **only** the driving schools which are **without** instructors. Show only the first **5 (five)** results.

**Order** the results **ascending** by **brand** and then by **driving\_school id**.

#### Required Columns

* id (driving\_school)
* name (driving\_school)
* brand (car brand)

#### Example

|  |  |  |
| --- | --- | --- |
| **id** | **name** | **brand** |
| 36 | Freedom Drive | Audi |
| 38 | GGO Driving School | Audi |
| 39 | GILL's Driving Academy | BMW |
| 42 | Tomcat Driving | Buick |
| 59 | OK 4U | Buick |

### Instructors with more than one student

Write a query that returns: the **first\_name, last\_name** and **count** of the studentsand the **city** (where their driving lessons take place) from **instructors** table. **Filter** only **instructors** who have **more than one** student.

**Order** by **students\_count** in **descending** order, then by **instructor's first\_name** **ascending**.

#### Required Columns

* first\_name (instructor)
* last\_name (instructor)
* students\_count (count of the students)
* name (city)

#### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **first\_name** | **last\_name** | **students\_count** | **city** |
| Christopher | White | 2 | Birmingham |
| John | Doe | 2 | Nottingham |
| Kyle | King | 2 | Peterhead |
| Melissa | Lewis | 2 | Leeds |
| Paul | Hogan | 2 | London |
| Robert | Thompson | 2 | Worcester |

### Instructor's count by city

Extract from the **go\_roadie** database the city **name,** and the **total** **count** of all instructors who **worked** in driving schools in each **city.** Only the cities where **there** **are** driving schools with **instructors** should be displayed. (*If the* ***count*** *of instructors in some cities is* ***0*** *(instructors\_count = 0) do not print this city****.****)*

**Order** the results **descending** byan **instructors\_count**.

#### Required Columns

* name (city)
* instructors\_count

#### Examples

|  |  |
| --- | --- |
| **name** | **instructors \_count** |
| Liverpool | 8 |
| Gosport | 7 |
| Aberdeen | 4 |
| . . . | . . . |
| Manchester | 1 |

### Instructor's experience level

From the database extract the **full name** and experience **level** of the **instructors**. If the **year** of their license acquisition is **1980** and later but before **1990** the user must see **"Specialist",** if it is **1990** and later but before **2000** it should display "**Advanced**". If it is **2000** and later but before **2008** it should display "**Experienced**", if it is **2008** and later but before **2015** -> "**Qualified**", if it is **2015** and later but before **2020** -> **"Provisional"** and finally if it is **2020** and later it should display "**Trainee**".

**Order** the results **ascending** by **year of license acquisition,** then by **first\_name** of the instructor.

#### Required Columns

* full\_name (first\_name + " " + last\_name)
* level (experience)

#### Example

|  |  |
| --- | --- |
| **full\_name** | **level** |
| John Doe | Specialist |
| William Jones | Specialist |
| Jane Smith | Specialist |
| Rebecca Baker | Specialist |
| . . . | . . . |
| Heather Allen | Trainee |

## Section 4: Programmability – 30 pts

The time has come for you to prove that you can be a little more dynamic on the database. So, you will have to write several procedures.

### Extract the average lesson price by city

Create a **user defined function** with the name **udf\_average\_lesson\_price\_by\_city (name VARCHAR(40))** that receives a **city name** and returns the average price of the lesson for all driving schools in a given city.

#### Required Columns

* name (city)
* average\_lesson\_price (average\_lesson\_price)

#### Example

|  |  |
| --- | --- |
| **Query** | |
| SELECT p.name, udf\_average\_lesson\_price\_by\_city (London) as average\_lesson\_price  FROM cities c  WHERE c.name = 'London' | |
| name | average\_lesson\_price |
| London | 51.60 |

### Find a driving school by the desired car brand

Create a stored procedure udp\_find\_school\_by\_car which accepts the following parameters:

* brand (VARCHAR(20))

Extract **name** and **average\_lesson\_price** from **driving\_schools** for all driving schools that offer the desired car **brand**. **Order** the results **descending** by **average\_lesson\_price**.

#### Result

|  |
| --- |
| **Query** |
| CALL find\_school\_by\_car (Mercedes-Benz); |
| This execution will return all the driving schools which offer Mercedes-Benz for driving. |

#### Result

|  |  |
| --- | --- |
| **name** | **average\_lesson\_price** |
| Learn2drive | 80.60 |
| Able2Drive SOM Notts | 66.99 |
| Evieonepasses | 60.00 |
| Roadstar Driving School | 54.00 |
| Roadsense | 45.00 |