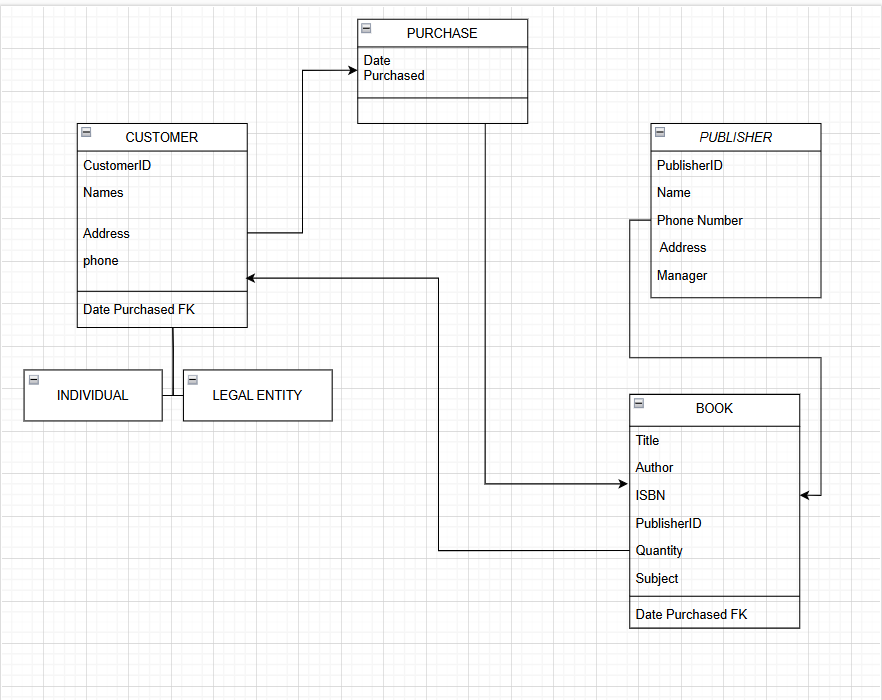
CRISTIAN NKOKO

MYSQL EXERCISE

BOOKSTORE ER-DIADRAM



The relationships would be

* Customer places many purchases
* Purchase is for one book
* Book is from one publisher
* Book is sold to more than one customer
* Bookstore purchases books from publishers

Logical Model would be:

* Customer, Publisher and Book become tables
* Purchase becomes a separate table with Fk’s to link customer & Book

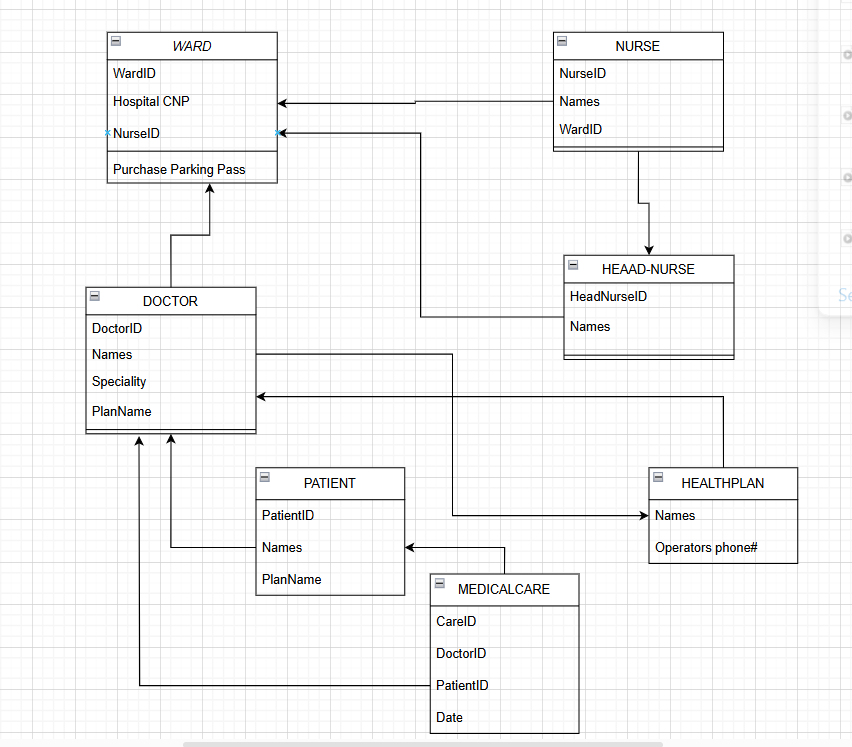
For Normalization to 3NF

Publisher table is already in 3NF

Customer and Book tables we can split them into two by having separate Address table

Purchase is already in 3NF since it does not have many attributes

HOSPITAL ER-DIADRAM



The relationships would be

* Ward has one Head-Nurse
* Ward has many nurses
* Nurse works in one ward
* Nurse reports to one Head-Nurse
* Doctors treats many patients
* Doctor is assigned to many wards
* Medical care provided by one doctor to one patient
* Health Plan accredits many doctors

Logical Model would be:

* Ward, Doctor, Patient, Nurse and Medical-Care become tables
* Heard-Nurse becomes a separate table to avoid redundancy

For Normalization to 3NF

Ward, Head-Nurse and Health Plan are already in 3NF

Nurse needs WardID and Head-NurseID removed because they are dependencies

Patient and MedicalCare are also in 3NF

Doctor needs PlanName removed because it’s a dependency

NORMALIZATION EXERCISE 1

Person

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RG | Names | {Adress} | Phone | {Skill} |

P\_Address

|  |  |  |  |
| --- | --- | --- | --- |
| RG | Address | Names | Phone |

P\_Skill

|  |  |  |  |
| --- | --- | --- | --- |
| RG | Skill | Name | Phone |

NORMALIZATION EXERCISE 2

Relation R

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G | H |

Functional Dependencies

{A, B} → {C, D, E, F}

{B} → {G, H}

Identify Candidate Keys: {A, B}, B

Now we check if the Functional Dependencies that violate normal forms:

Relation R1

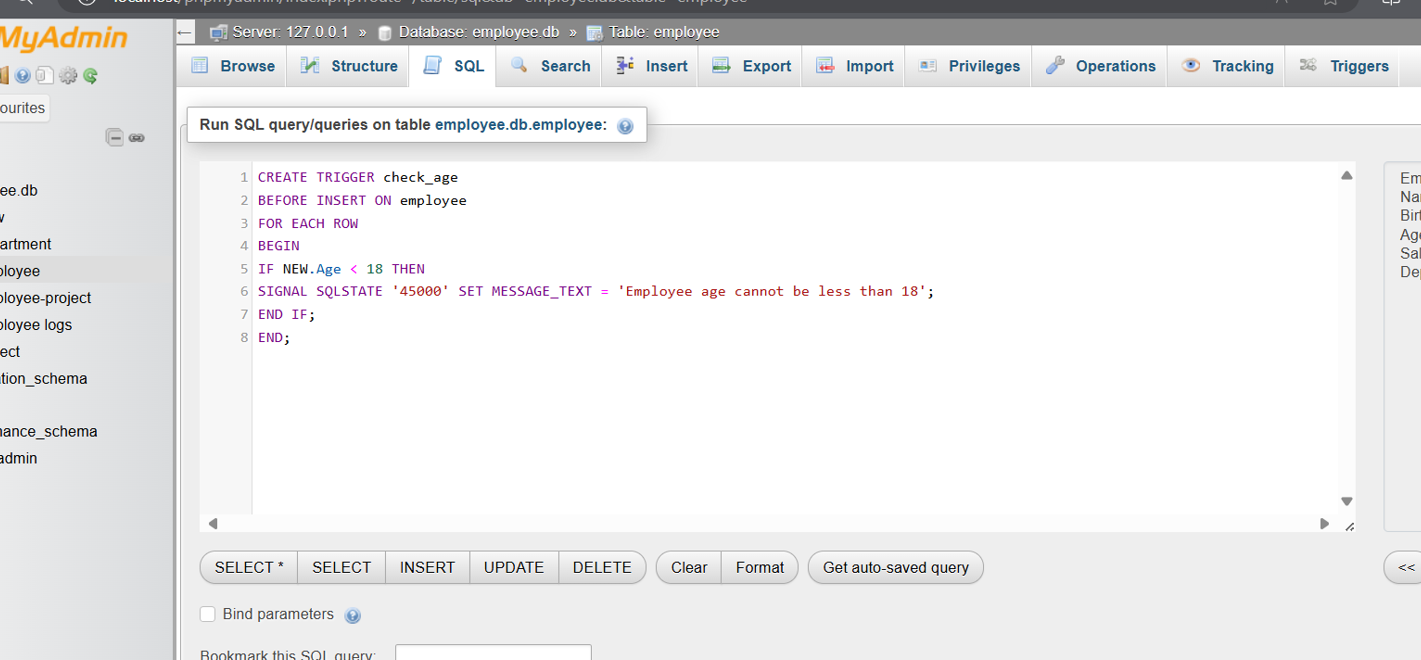
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F |

Relation R2

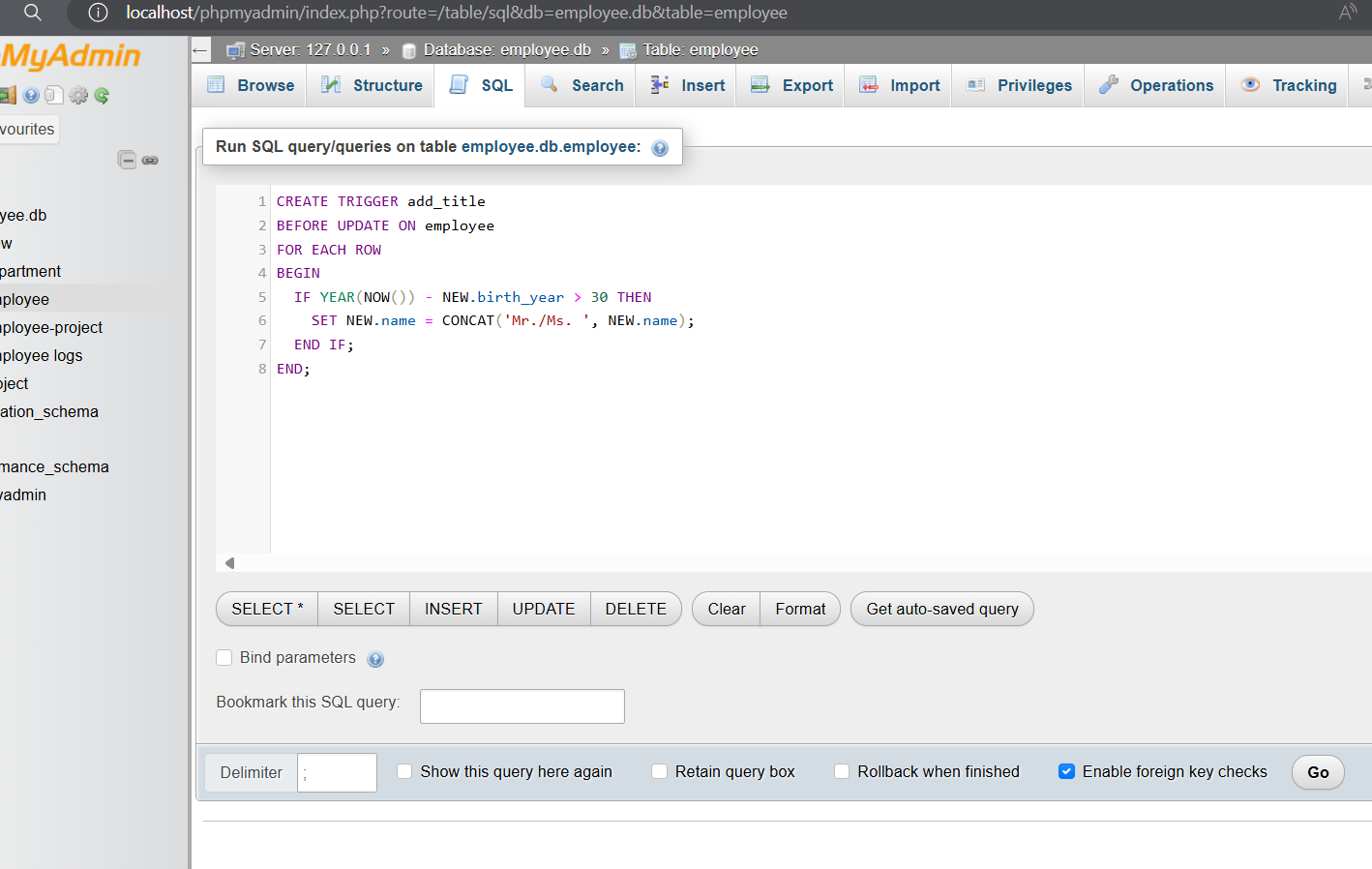
|  |  |  |
| --- | --- | --- |
| B | G | H |

TRIGGERS

1. Create a TRIGGER that prevents an employee under the age of 18 from being inserted.

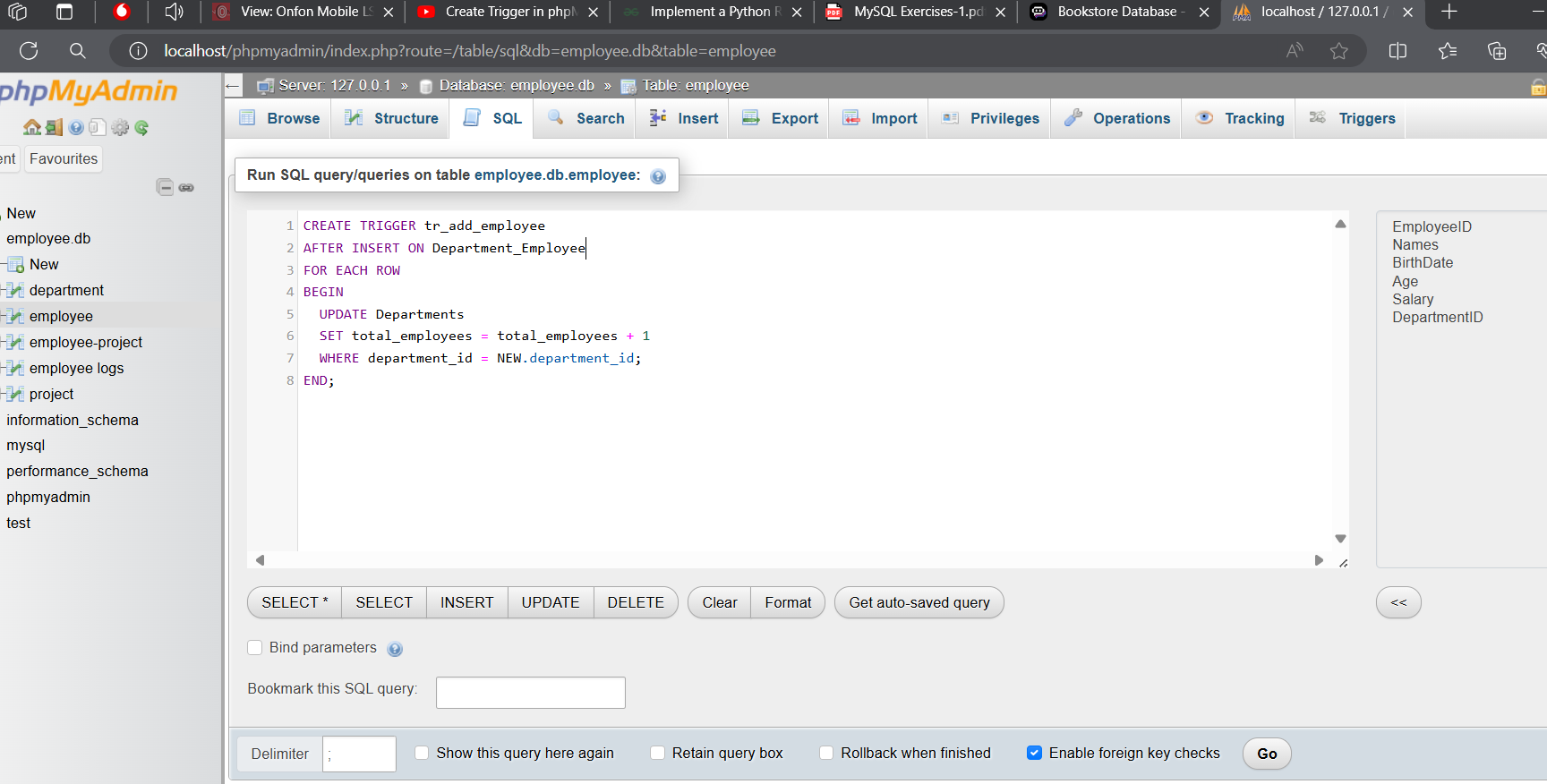


1. Create a TRIGGER to add the word "Mr. / Ms." to the name of PEOPLE who were born more than 30 years ago.

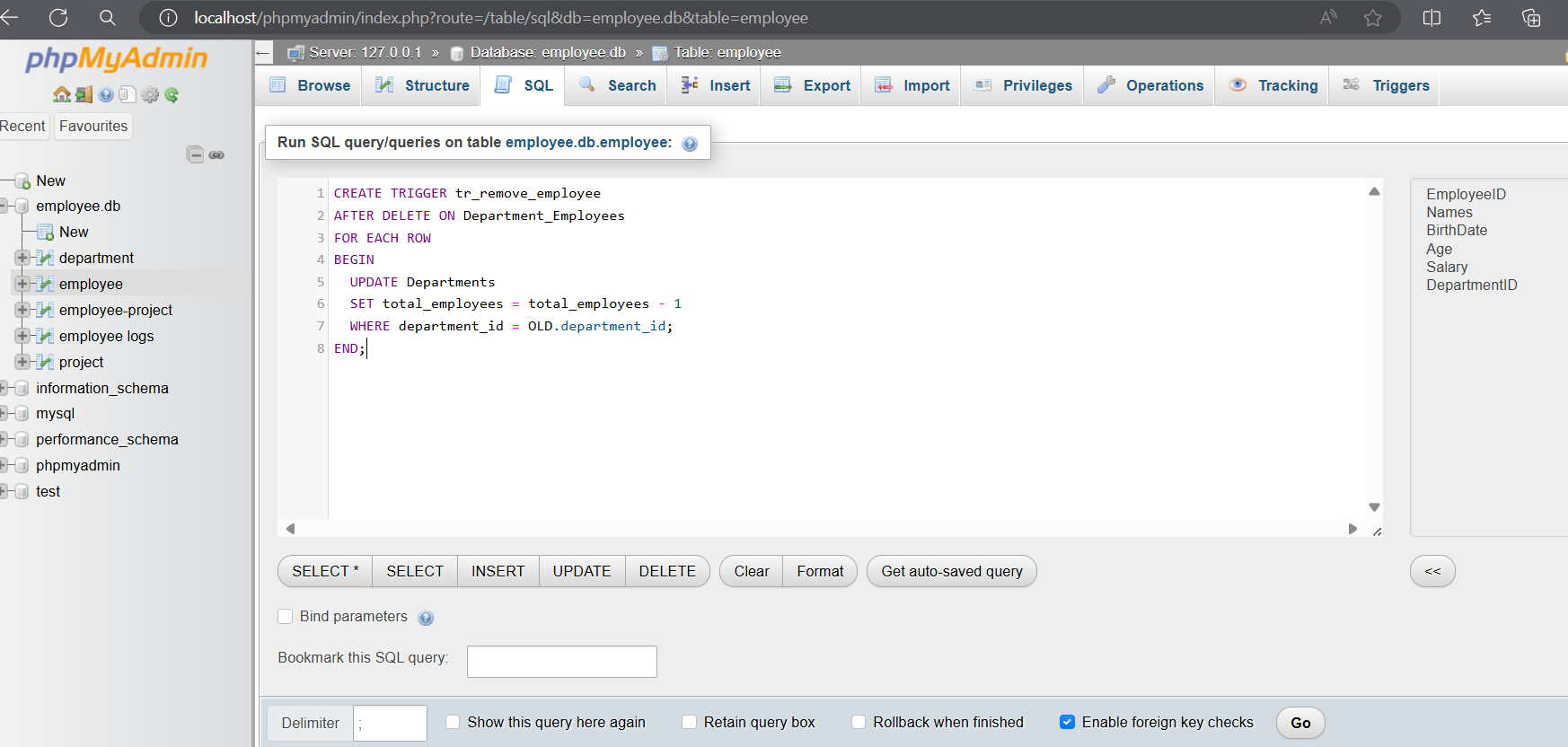


1. Add the attribute total\_employees and then keep the total number of employees in the Department table updated considering:

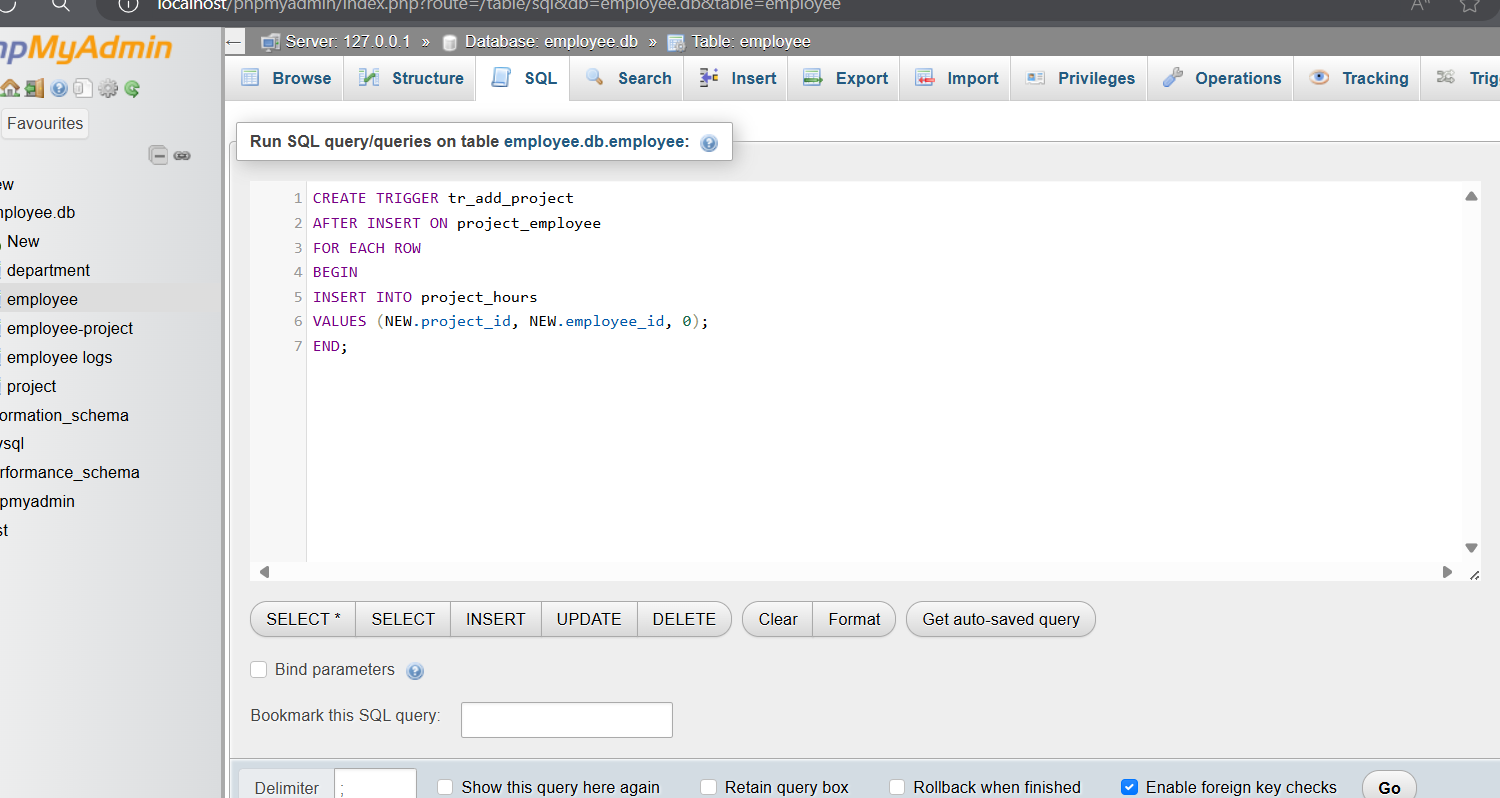
To Handle Adding Employees:



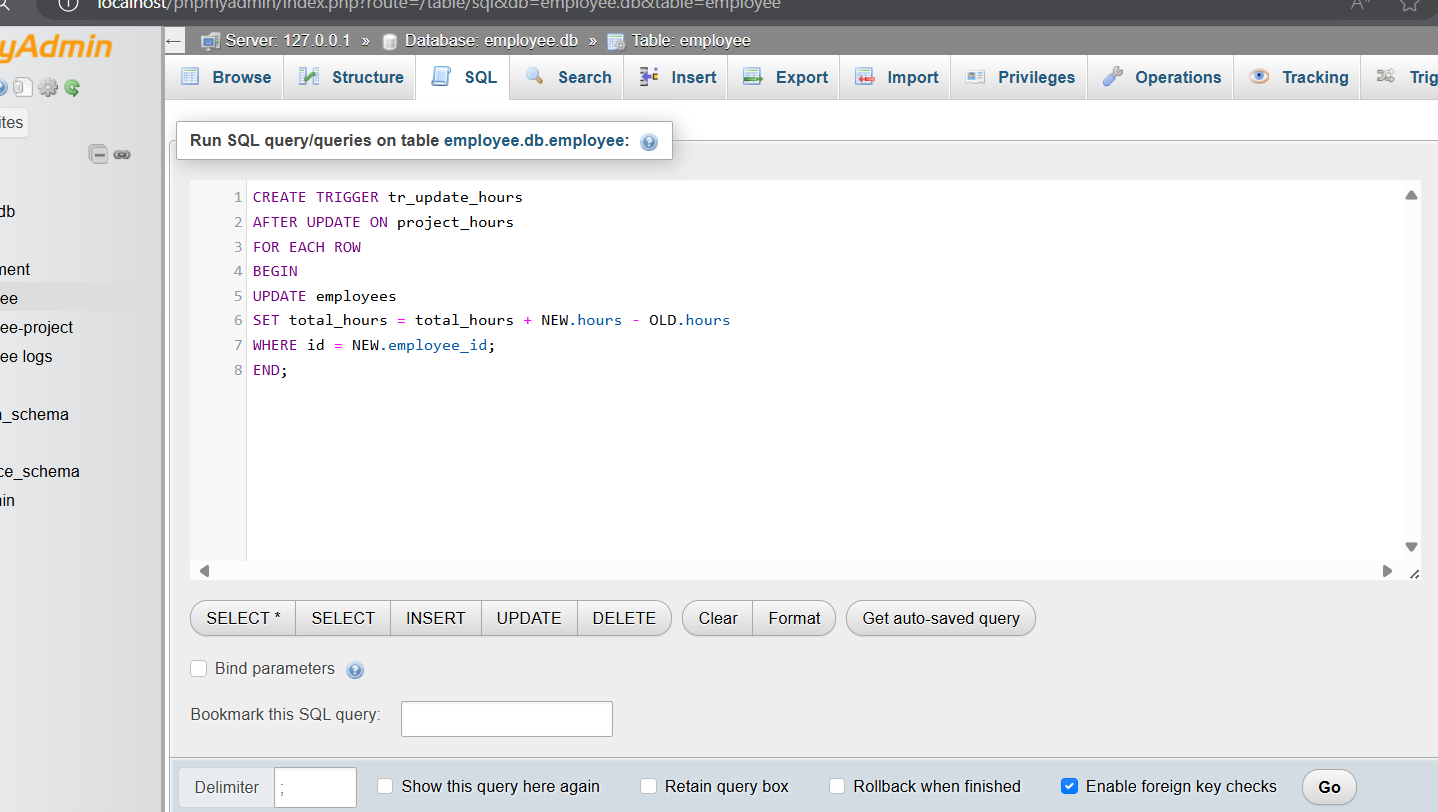
To Handle Removing Employees:



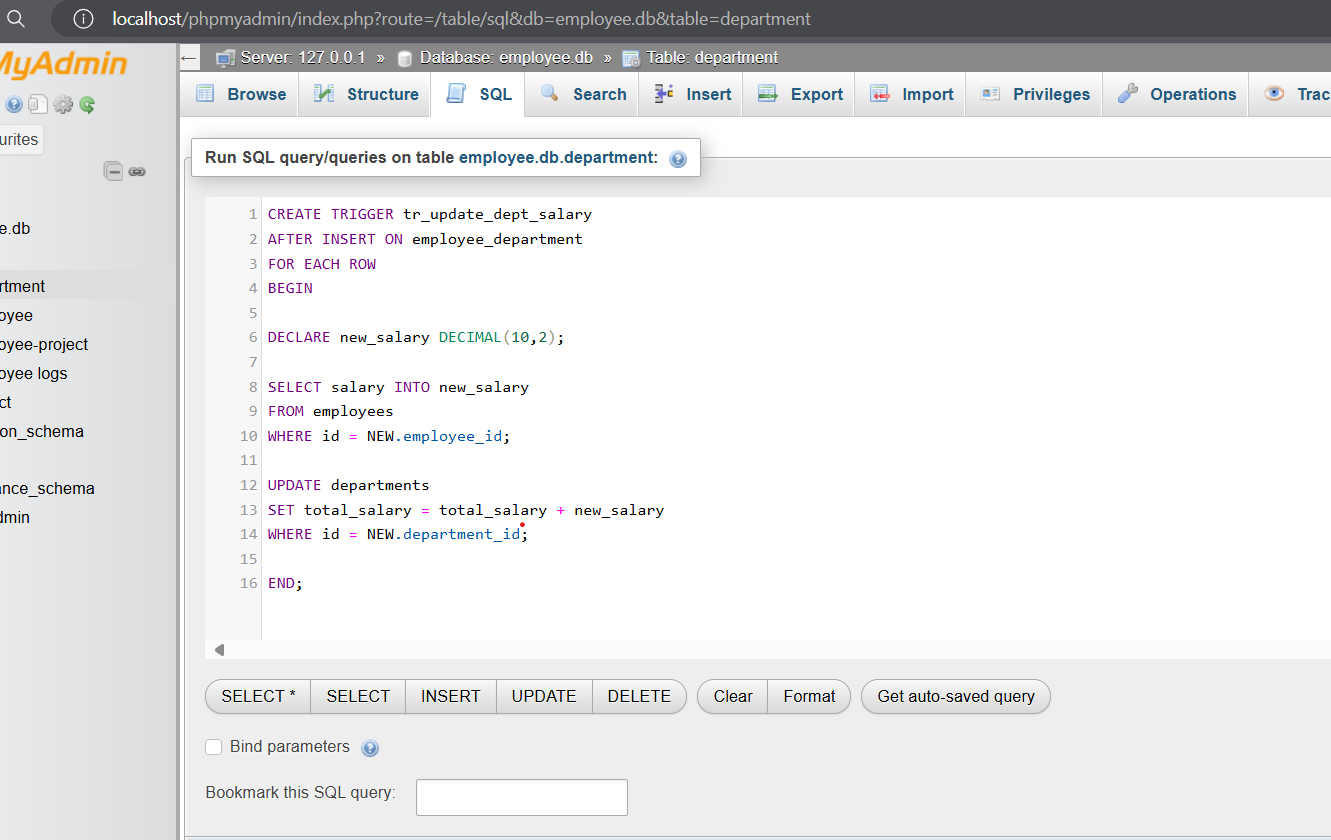
1. Trigger to initialize Hours on project assignment



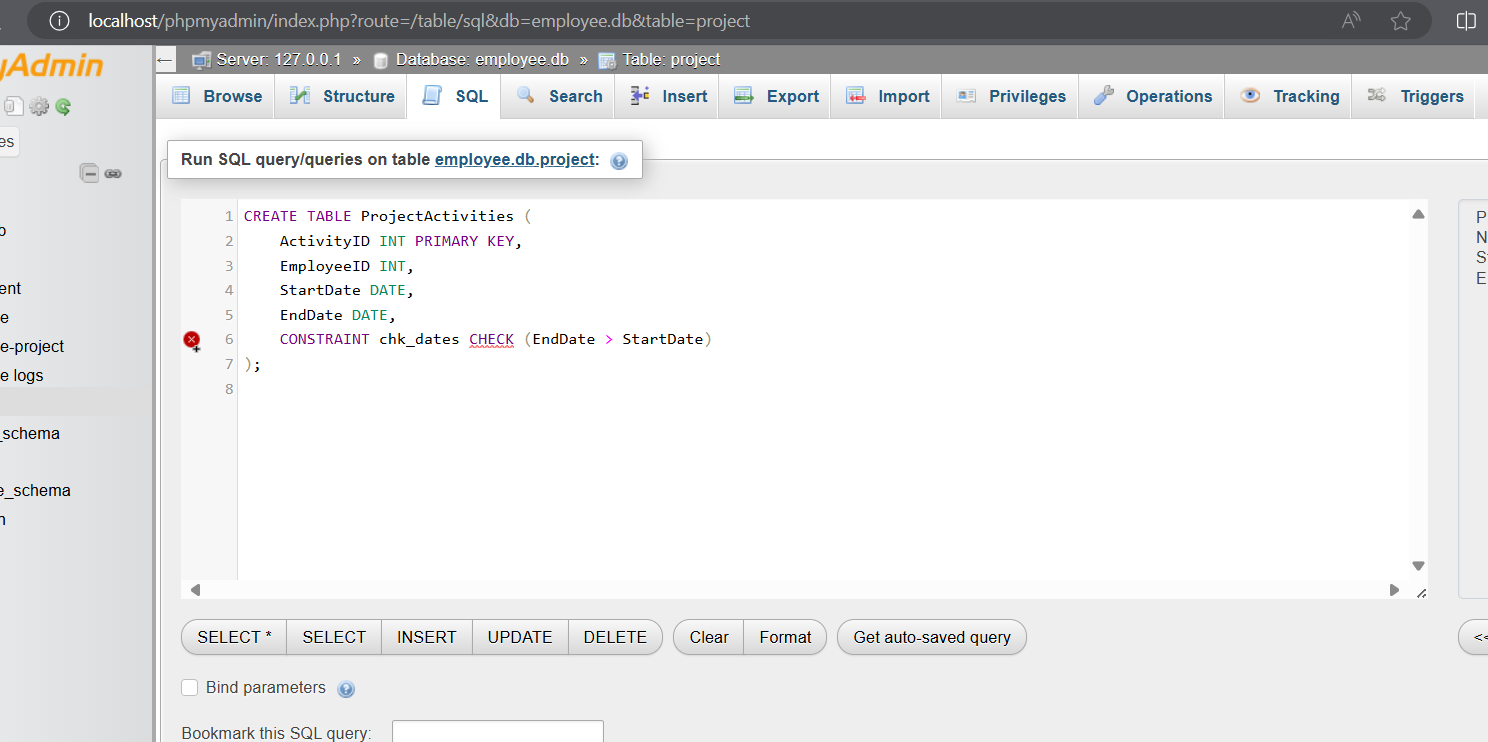
Trigger to recalculate totals on hours update



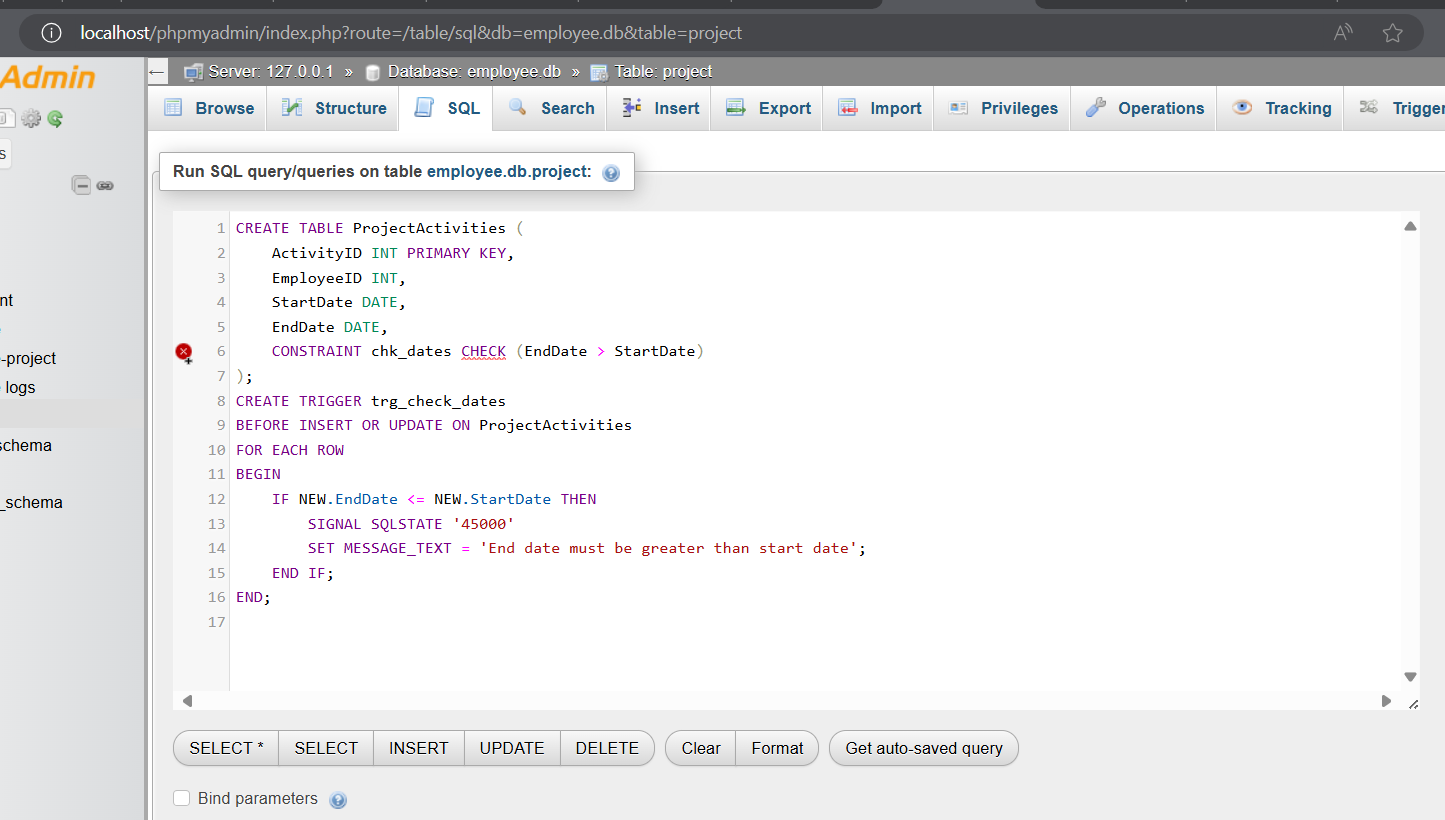
1. Create a trigger to update the total salary value of the department where a new employee is assigned.



1. a) Check type constraint



b) Trigger



1. Write a procedure that allocates the worker to a project

