ACIT 1630 - Relational Database Design and SQL

More Relationships in MySQL Server

Introduction:
The goal of this document is to serve as a walkthrough for creating Many-to-Many and Unary relationships in MySQL Server.

Remember that Many-to-Many Relationships are best implemented as two Many-to-One relationships. There is also a composite entity that serves as an intermediate table between the 2 entities (tables). The relationship will implement is the Person - Skill relationship from the previous ERD exercise.

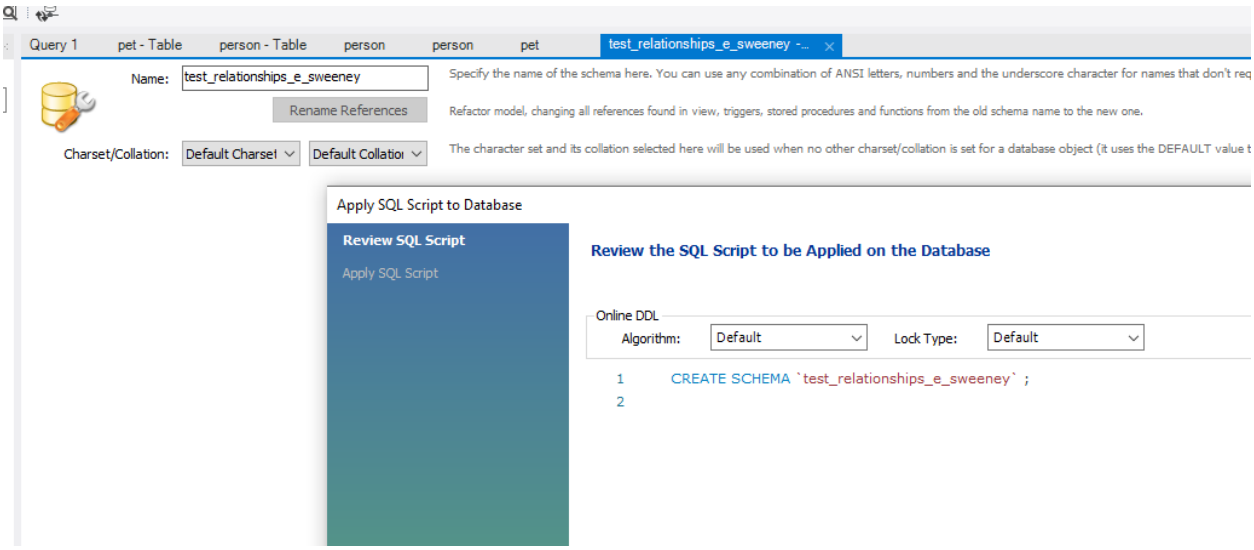
We will also be creating unary relationship between an Employee and their boss (which is also an Employee).

Step 1:
Open MySQL Workbench

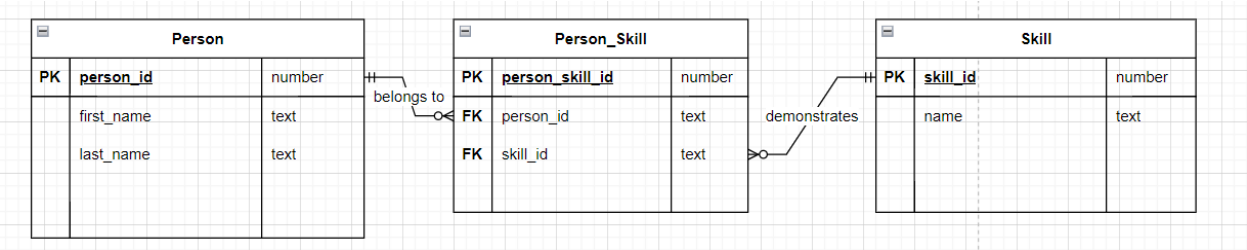
Step 2:
Create a new database in the following format:

Test_Relationships_ + Your First Initial + _ + Your Last Name

As an example, for me I would create the database as:
Test_Relationships_E_Sweeney



Step 3:
Using your ERD as a design, implement the Person, Person_Skill and Skill tables.

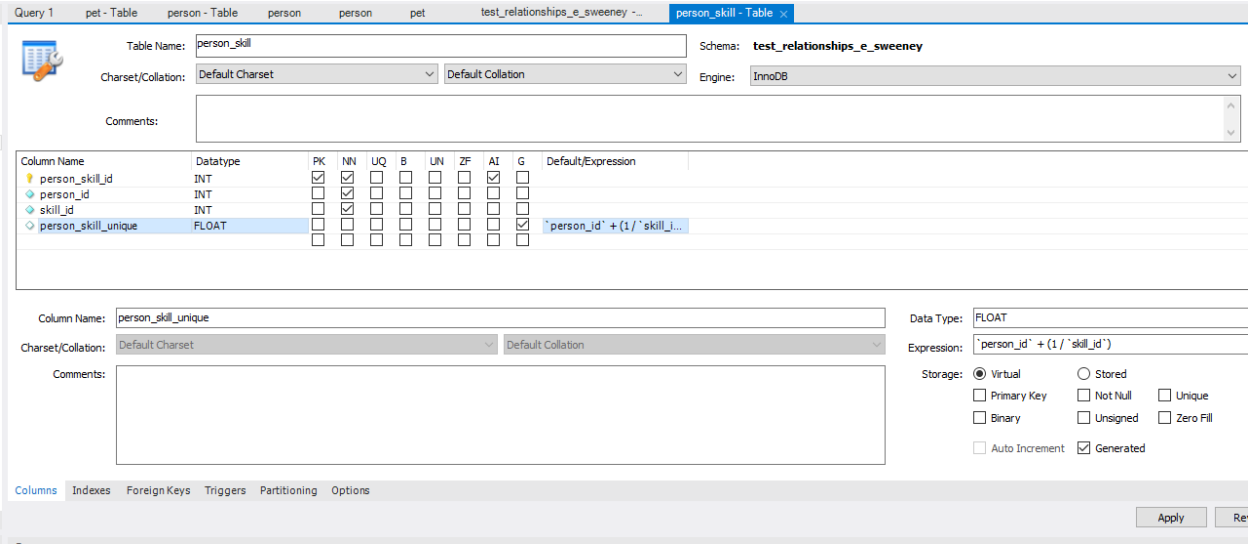


Add the appropriate primary keys, foreign keys, required attributes (Allow Null = No), optional attributes (Allow Null = Yes).

IMPORTANT: Make sure that all your primary keys have the AI (auto increment) field attribute (which we set in the alter table function) so that MySQL will generate index values for you automatically.

Which tables will need to be created first?
Hint: can a foreign key exist before the primary key it refers to?

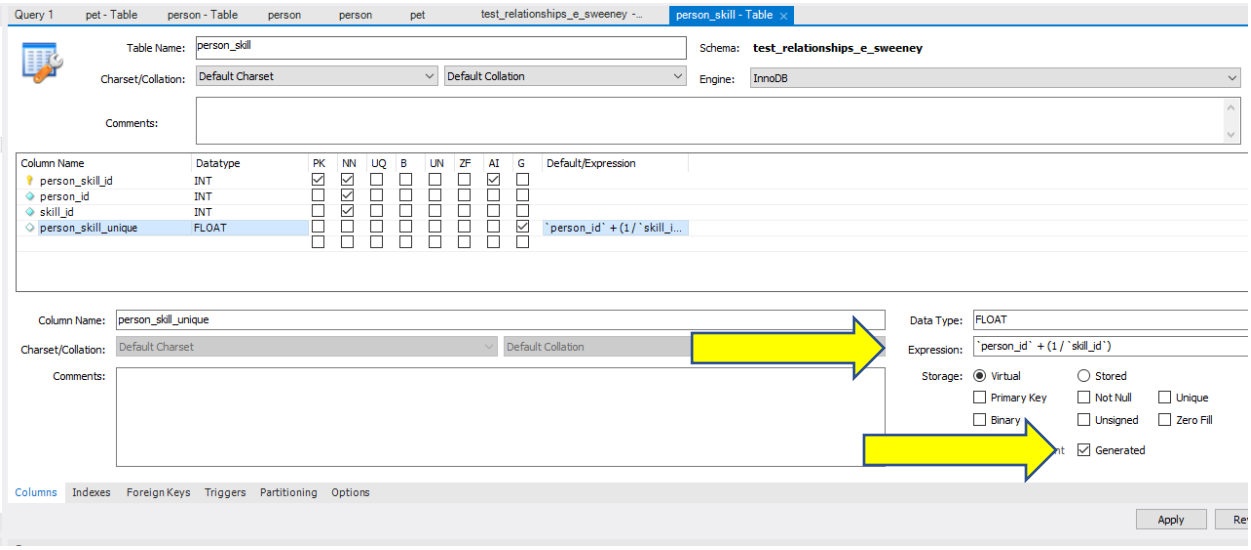
On your Person_Skill table, add both your foreign keys.
Make sure that person_id in the Person_Skill table matches the person_id in the Person table **and** skill_id in the Person_Skill table matches skill_id in the Skill table.



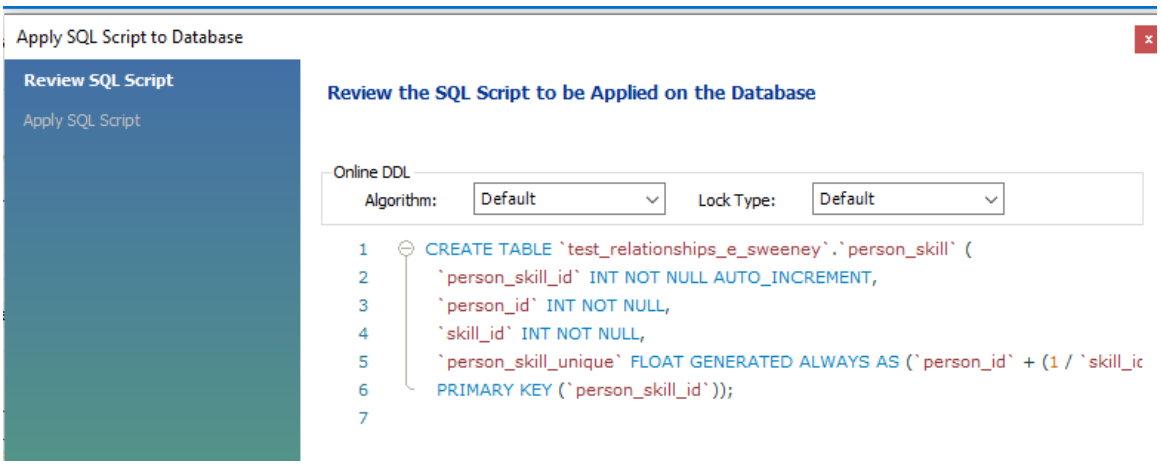
Once you are done adding your tables, remember that we wanted to ensure that a person couldn't have the same skill twice?

Let's create a new auto-generated field and add a constraint that it must be unique, to ensure that the combination of the person_id and skill_id columns cannot be repeated.

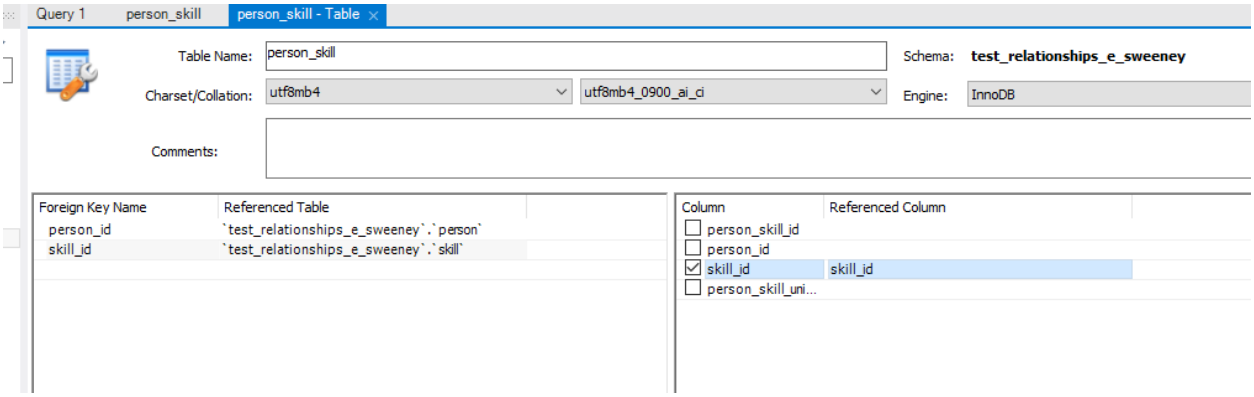
There are many ways to achieve this, probably the easiest in MySQL is to create a float (data type) field that, in addition to the skill_id and person_id foreign keys, create a single field that is a combination of both foreign keys, so it takes person_id as its whole number portion of the float number, and the skill_id as its decimal portion of the number.



The SQL code should look like this:

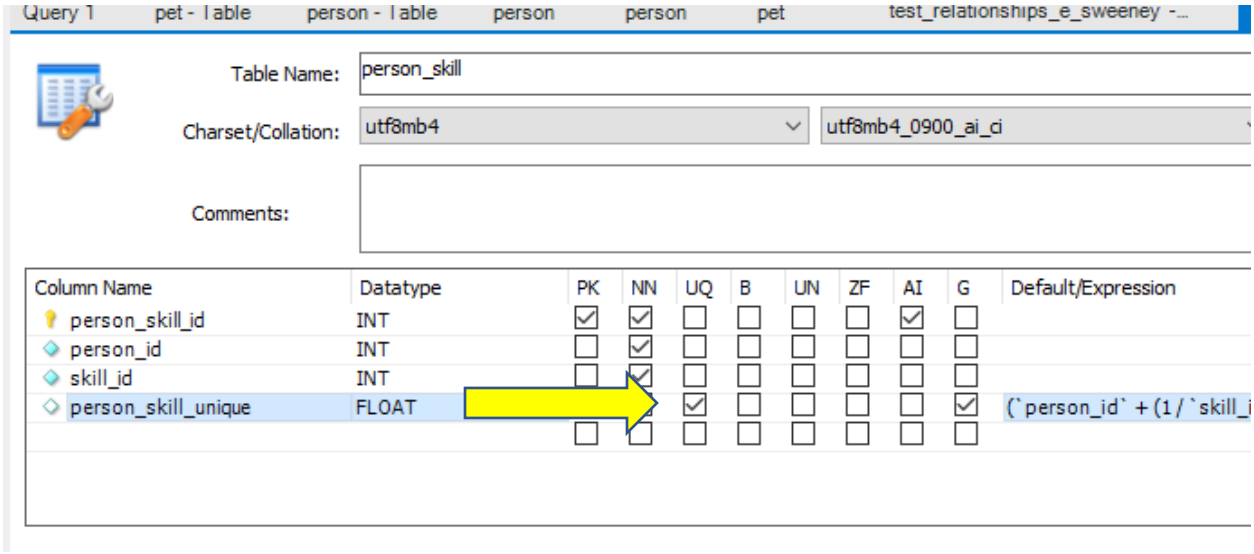


Next add the two foreign keys for skill and person



Click Apply

Let's add the constraint for the skill name to be unique so that no 2 skills have the same name.



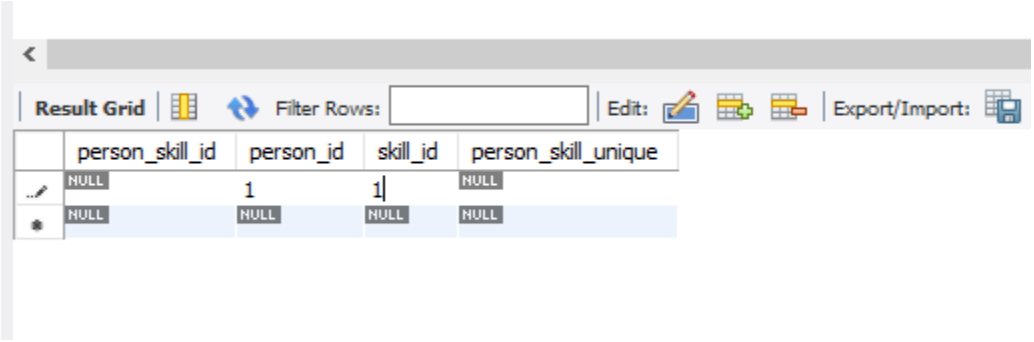
This should prevent adding "Golf" (or any other skill) twice as a skill.

Step 4:
Adding our skilled people.

What order do we need to add our data to our tables? If our database is doing its job, it should prevent us from adding to our person_skill table data before we have both people and skills data.

Let's test this.

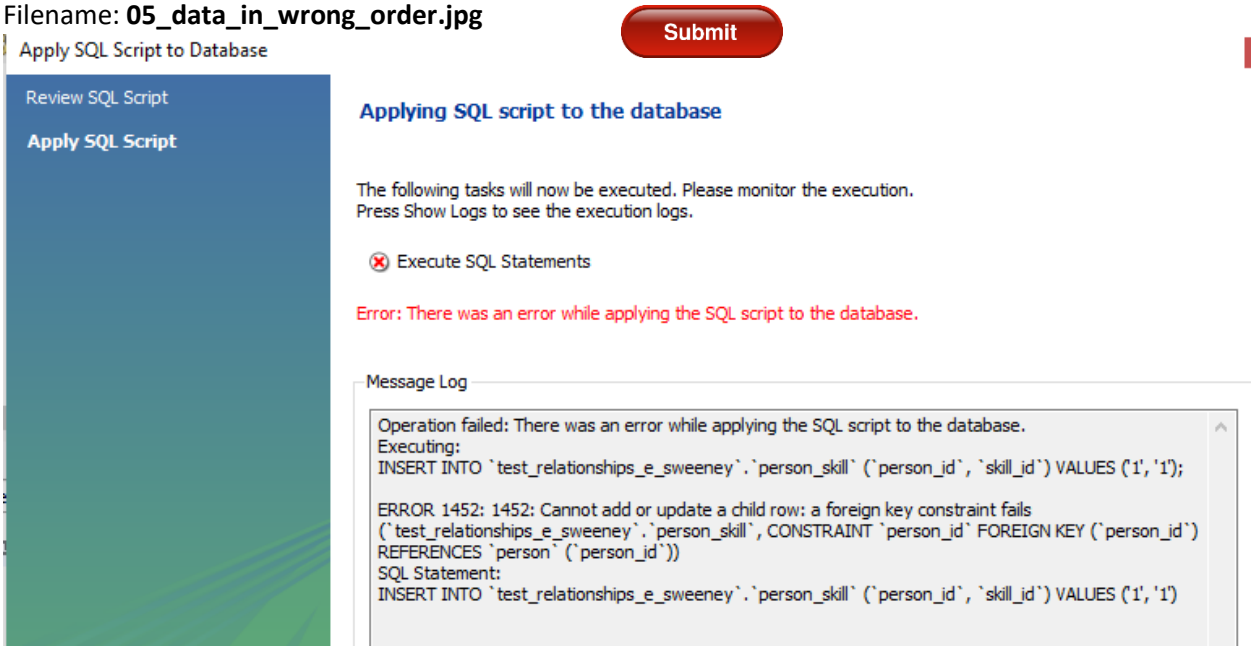
Using the table data editor, accessed by hovering over the table name in the navigator window, edit the person_skill table data and add a row with person_id = 1 and skill_id = 1.



Click Apply.

Submit your screenshot of you attempting to adding to person_skill before adding to the person or skill tables.

Filename: **05_data_in_wrong_order.jpg**



Add the following data, this time in the correct order.

Person	**Skills**
Mario Ben Turner	Carpentry, Metalwork, Automotive Repair, Fishing, Golf
Jessica Colleen Ingram	Painting, Rock Climbing, Rollerblading, Skiing
Zoila Jones	Cooking, Reading, Computer Programming, Golf
Craig Oyer	Hockey, Golf, Automotive Repair
Robert Larry Pickard	Rock Climbing, Computer Programming, Salsa Dancing

IMPORTANT! Pay close attention to what number gets assigned to each person and skill for your primary key columns, person_id and skill_id, so that you can properly link them in the person_skill table.

Submit your screenshot of your person table data.

Submit

Filename: **06_person_data.jpg**

	person_id	first_name	middle_name	last_name
		Mario	Ben	Turner
		Jessica	Colleen	Ingram
		Zoila	*NULL*	Jones
		Craig	*NULL*	Oyer
		Robert	Larry	Pickard
▶*	*NULL*	*NULL*	*NULL*	*NULL*

Submit your screenshot of your skill table data.

Submit

Filename: **07_skill_data.jpg**

	skill_id	name
		Carpentry
		Metalwork
		Automotive Re...
		Fishing
		Golf
		Painting
		Rock Climbing
		Rollerblading
		Skiing
		Cooking
		Reading
		Computer Prog...
		Hockey
		Salsa Dancing
▶*	*NULL*	*NULL*

Submit your screenshot of your person_skill table data.

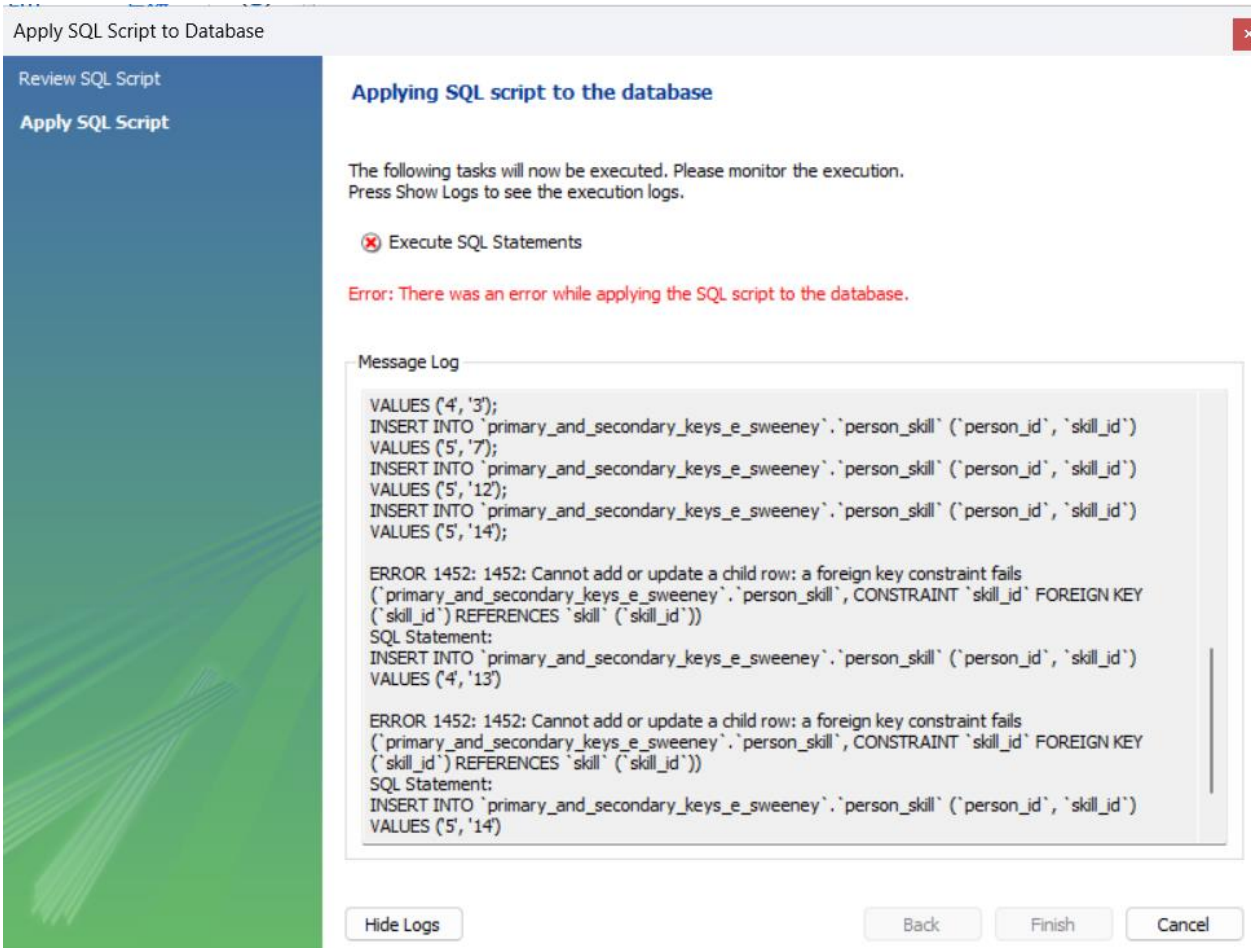
Submit

Filename: **08_person_skill_data.jpg**

person_skill_id	person_id	skill_id
	1	1
	1	2
	1	3
	1	4
	1	5
	2	6
	2	7
	2	8
	2	9
	3	10
	3	11
	3	12
	3	5
	4	13
	4	5
	4	3
	5	7
	5	12
	5	14
NULL	NULL	NULL

Let's try to add Mario's Carpentry skill a second time and confirm that our unique constraint is working.

Submit your error image:



Awesome! It failed like it should. This should help us accidentally add the same skill to the same person twice!

That's it! You're done!