# Data Centric RAD

## Lab 2 MySQL Review II

### Part 1

* Get superheroes\_wk2\_part1.sql from Moodle.
* Import it into MySQL as follows:
  + Go to the following directory in the cmd tool: wamp64\bin\mysql\mysql5.7.9\bin
  + Run the following command:

mysql -u root -p < "*<Full Path >*\superheroes\_wk2\_part1.sql"

* use superheroes;
* Insert the following information into the superhero\_table:

name = Penguin

City = Gotham City

Real First Name = Danny

Real Surname = DeVitto

dob = January 14th 1955

powers = 44.57



* Insert the following information into the superhero\_table:

name = Joker

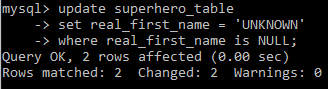
city = New York

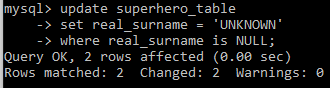
powers = 77.10



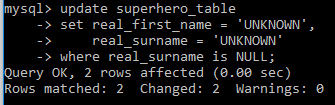
* Update the superhero\_table so that everywhere a first name or surname is NULL, it is now UNKNOWN instead.

HINT: When checking for NULL, you can’t say = NULL, you should say is NULL.

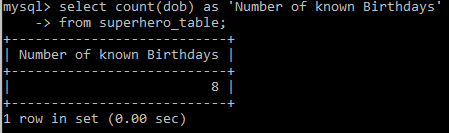




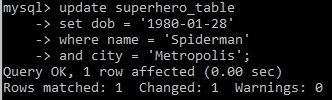
*or*



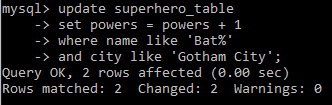
* Display the number of superheroes, whose birthdays are known, as ‘Number of known Birthdays’.



* Change the birthday of Spiderman from Metropolis to January 28th 1980.



* Increase the powers by 1 for each ‘Bat’ related superhero in Gotham City.

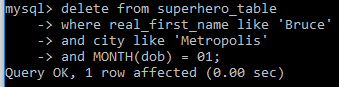


* Delete all superheroes whose *dob* is NULL.

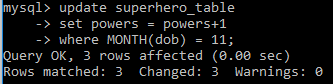
HINT: When checking for NULL, you can’t say = NULL, you should say is NULL.



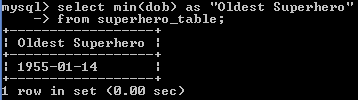
* Delete all superheroes whose real first name is *Bruce*, and who live in *Metropolis* and who were born in *January*.



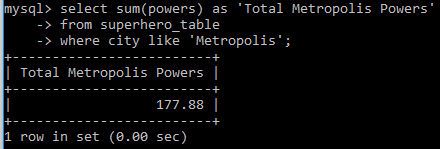
* Increase the powers of all superheroes who were born in *November* by 1.



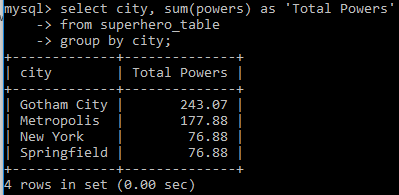
* Display dob (as *Oldest DOB*) of the *oldest* superhero.



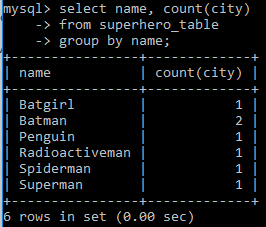
* Display the total powers (as *Total Metropolis Powers*) of all superheroes in *Metropolis*.



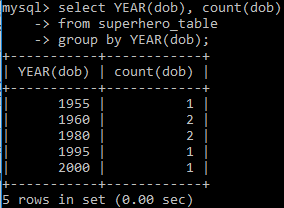
* Display the city and the total powers (as *Total Powers*) of all superheroes in the city, for each city.



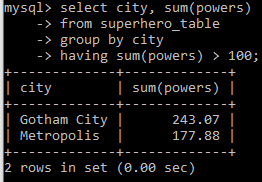
* Display the superhero name, and the number of cities a superhero of that name lives in.



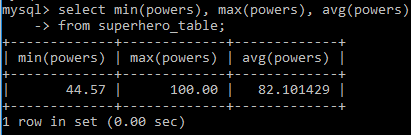
* Display the year and the number of superheroes born in that year.



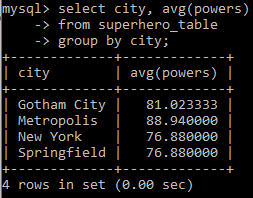
* Display the city and the total powers of all superheroes in that city, only for cities whose total superhero powers is greater than 100.



* Show the minimum, maximum and average powers of the superheroes.

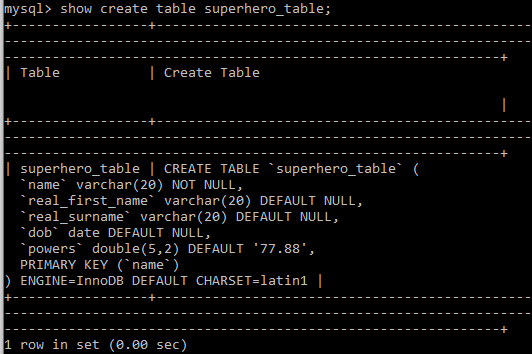


* Show the city and the average powers of all superheroes in that city; for all cities.



### Part 2

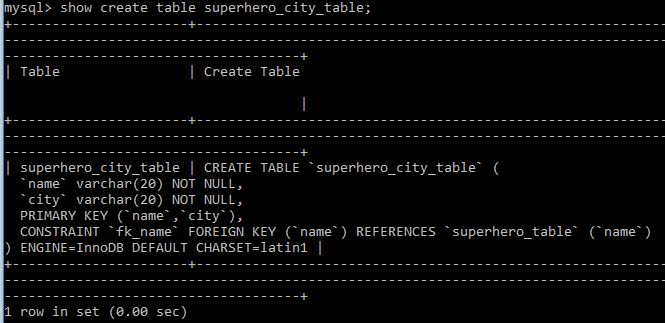
* Get superheroes\_wk2\_part2.sql from Moodle.
* Import it into MySQL described in Lab 1.
* use superheroes;
* Use the show create table <table name> command to find out the structure of the superhero\_table, and list the Primary Key(s) and Foreign Key(s).



Primary Key = name

There are no Foreign Keys.

* Use the show create table <table name> command to find out the structure of the superhero\_city\_table, and list the Primary Key(s) and Foreign Key(s).

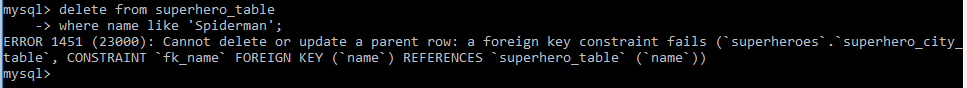


Primary Key = name, city

Foreign Key = name and it references the name column in the superhero\_table.

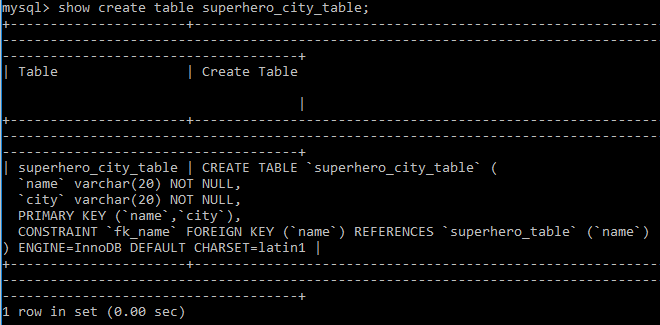
* Delete *Spiderman* from the superhero\_table.

What happens and why?



*Spiderman* cannot be deleted because it is referenced in the *superhero\_city\_table*.

When a show create table is performed on the *superhero\_city\_table* there is no ON DELTETE CASCADE, or ON DELETE SET NULL option, therefore the default is ON DELETE RESTRICT.



* Insert a new superhero in the *superhero\_table* as follows:

Name = ‘Joker’

real\_first\_name = ‘John’

real\_surname = ‘Jones’

dob = 1966-07-12

powers = 22



* Delete the superhero *Joker* from the *superhero\_table*.

What happens and why?

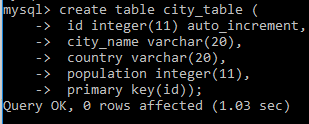


The row is successfully deleted because the value of the *name* column (*Joker*) is not used in the Foreign Key of the superhero\_city\_table.

* Delete the superhero\_city\_table as follows:

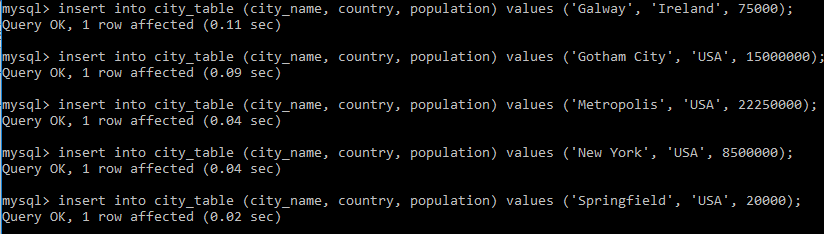


* Create a new table called *city\_table* as follows:



Populate it with the following data:

|  |  |  |
| --- | --- | --- |
| **Name** | **Country** | **Population** |
| Galway | Ireland | 75,000 |
| Gotham City | USA | 15,000,000 |
| Metropolis | USA | 22,250,000 |
| New York | USA | 8,500,000 |
| Springfield | USA | 20,000 |

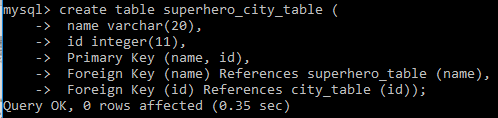


* Recreate the superhero\_city\_table, this time with two columns:
  + *name* which is a Foreign Key referring to the name column in the superhero\_table.
  + *city* which is a Foreign Key referring to the *id* column in the city\_table.

HINT: A Foreign Key is created using the following syntax:

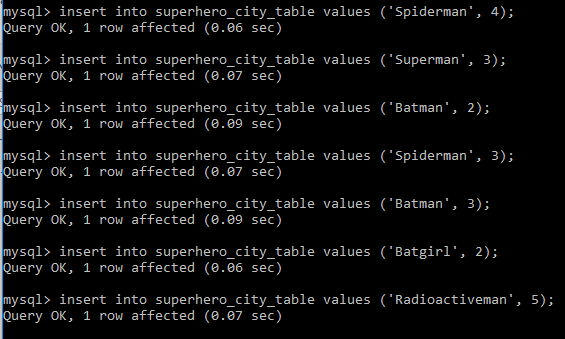
**Foreign Key(***column***) References** *table\_name* **(***column\_in\_referenced\_table***)**.

* + Primary Key is (name, id)



* Populate the superhero\_city\_table so that the following are associated:

|  |  |
| --- | --- |
| Spiderman | New York |
| Superman | Metropolis |
| Batman | Gotham City |
| Spiderman | Metropolis |
| Batman | Metropolis |
| Batgirl | Gotham City |
| Radioactiveman | Springfield |



* Delete *Galway* from city\_table.

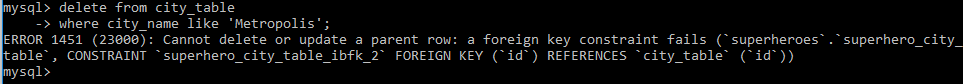
What happens and why?



The row is successfully deleted because the value of the *city\_name* column (*Galway*) is not used in the Foreign Key of the superhero\_city\_table.

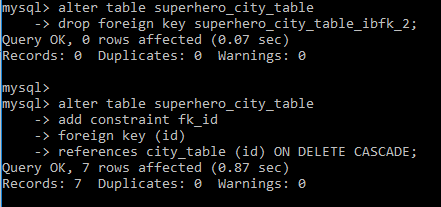
* Delete *Metropolis* from city\_table.

What happens and why?

 *Metropolis* cannot be deleted because it is referenced in the *superhero\_city\_table*.

When a show create table is performed on the *superhero\_city\_table* there is no ON DELTETE CASCADE, or ON DELETE SET NULL option, therefore the default is ON DELETE RESTRICT.

* Alter the city\_table as follows:

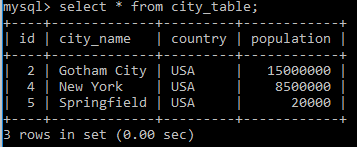


* Delete *Metropolis* from city\_table.

What happens and why?



*Metropolis* has been deleted from the city\_table:



And also from the superhero\_city\_table:

