# C2- S4-PRACTICE

*NOTE: check your* ***THEORY slides*** *to answer those questions!*

# EXERCISE 1 – BOOK & AUTHORS

We want to manage books and authors:

* A book has always 1 author only
* An author could write many books.

|  |
| --- |
| **Author** |
| authorID |
| name |
| dateOfBirth |
| country |

|  |
| --- |
| **Book** |
| bookID |
| Title |
| publishYear |
| language |

**Q1** – What is the relation between Book and Author tables? Why?

The relation between Book and Author table is one to many

* + Complete the missing attributes or table to allow this relation

One many

|  |
| --- |
| * **Author** |
| authorID |
| name |
| dateOfBirth |
| country |

|  |
| --- |
| **Book** |
| bookID |
| Title |
| publishYear |
| language |
| authorID |

**Q2** – For each table, complete the following arrays, by specifying for each attribute:

* + The field type (SQL type) and size
  + Can be null or not?
  + Is a primary key or foreign keys?

**AUTHOR TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute name | Type / size | Can be Null? | Key |
| author\_id | string | Not null | PK |
| Name | string | Not null |  |
| birth | string | Not null |  |

**BOOK TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute name | Type / size | Can be Null? | Key |
| Book\_id | string | Not null | PK |
| Title | string | Not null |  |
| language | string | Not null |  |

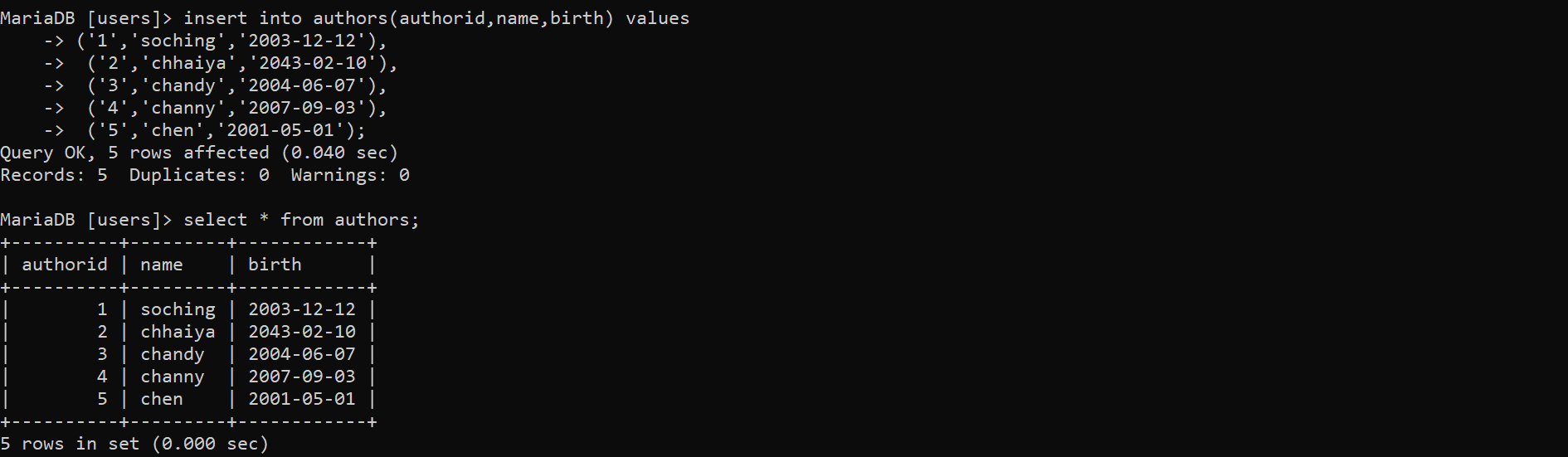
**Q3** – Write the SQL statement to create the 2 tables with appropriate properties

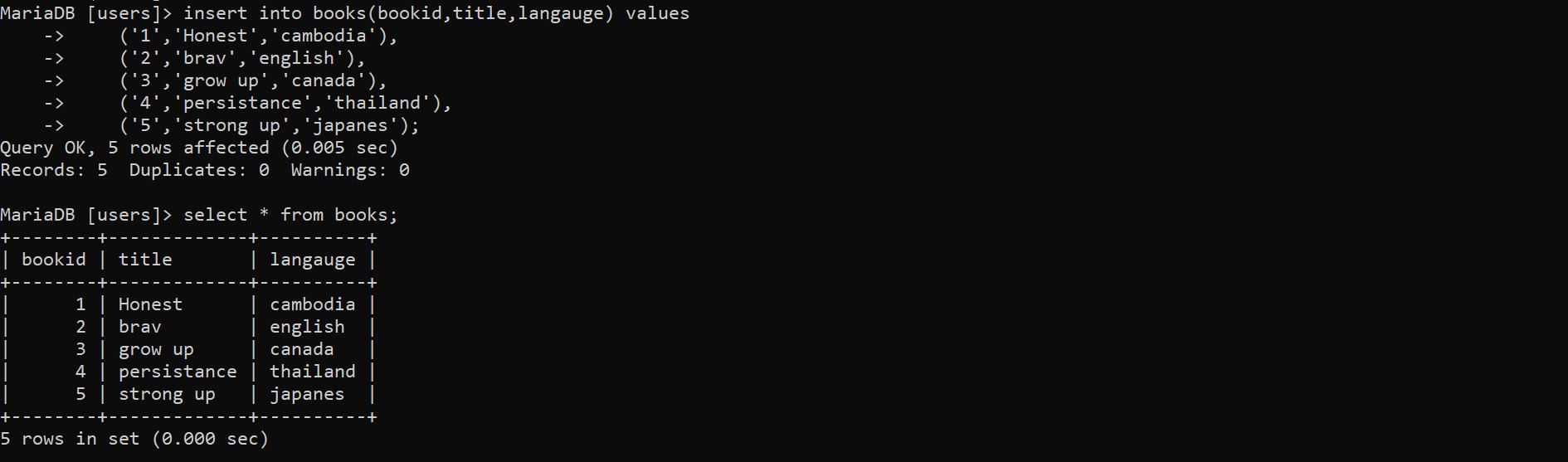


****

**Q4–** Write the statement to insert 5 books and 5 authors

* + Find the book and author information on the Internet



****

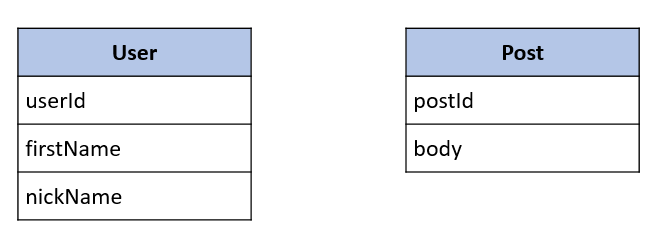
**Q5–** Write the SQL statement to **delete 3 of your books** from the database

# 

# EXERCISE 2 – USERS & POSTS

We want to manage **users** and **posts** (like posts on Facebook)

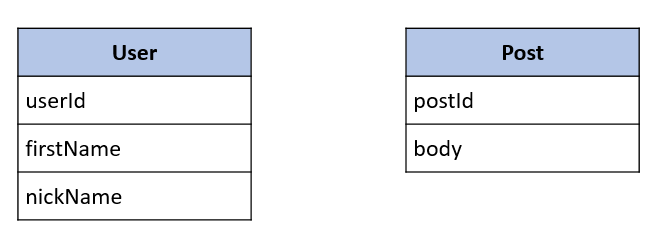
* A post is related to **1 user only**
  + A post has a body (the text of the post)
* User can have **many posts**
  + A user has a first name, and a nick name (optional)



one

**Q1** – What is the relation between User and Post Table?

* + Complete the missing attributes or table to allow this relation



userid

One many

**Q2** – For each table, complete the following arrays, by specifying for each attribute:

* + The attribute type (SQL type) and size
  + Can be null or not?
  + Is a primary key or foreign keys?

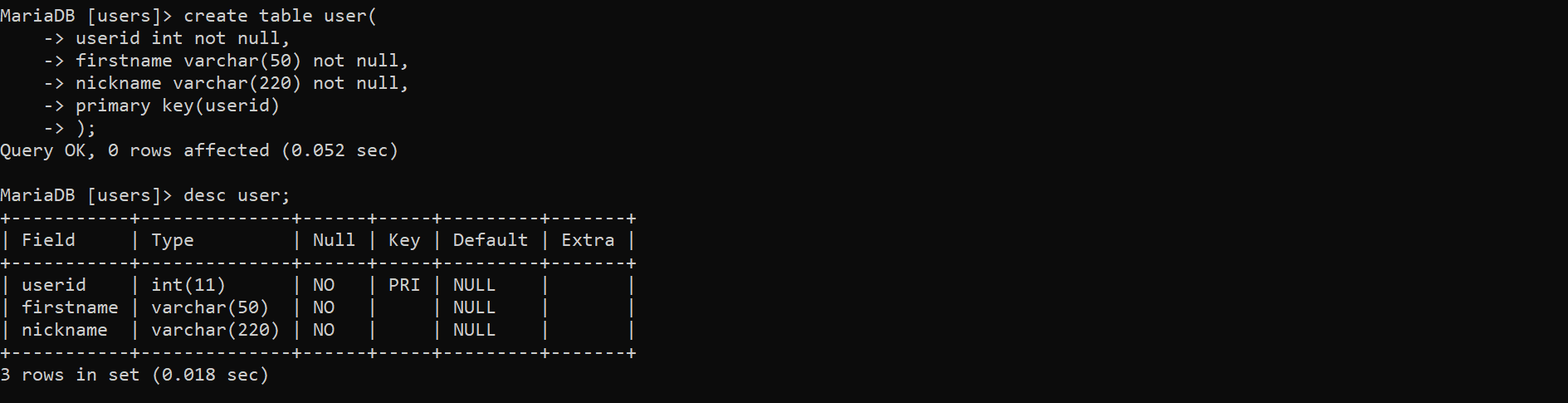
**USER TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute name | Type / size | Null? | Key |
| User\_id | int | Not null | PK |
| First name | Varchar(50) | Not null |  |
| nickname | Varchar(220) | Not null |  |

**POST TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute name | Type / size | Null? | Key |
| Post\_id | int | Not null | PK |
| body | Varchar(50) | Not null |  |
| User\_id | int | Not null | FK |

**Q3** – Write the SQL statement to create the 2 tables with appropriate properties



****

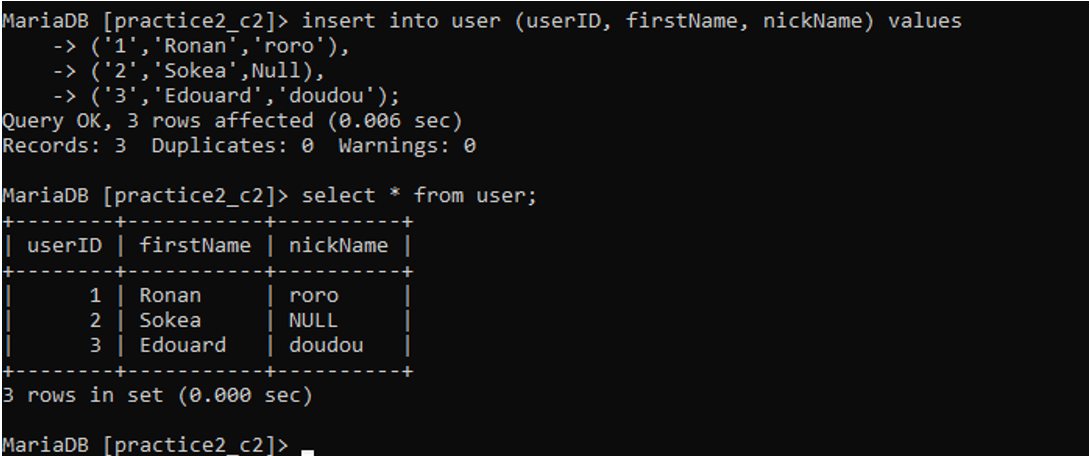
**Q4–** Write the statement to insert the following users and posts

Notes:

* ---- means: no value (the nickname is optional!)
* We don’t specify the KEY, it’s your business!

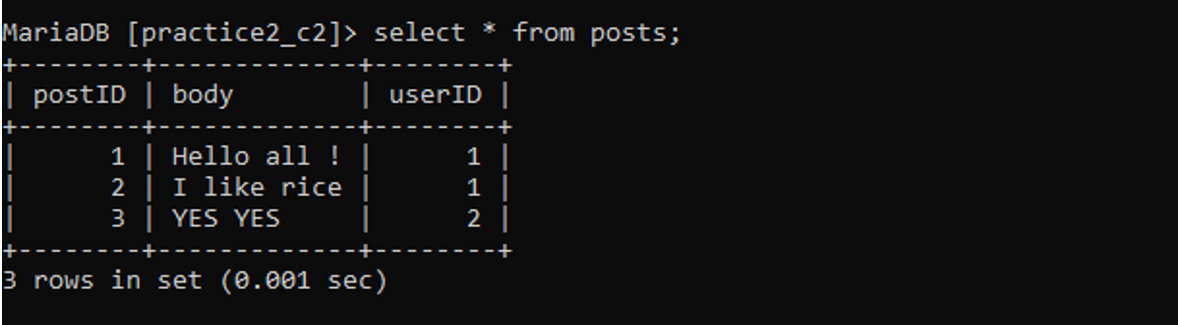
**USERS**

|  |  |
| --- | --- |
| First name | Nick name |
| Ronan | roro |
| Sokea | ---- |
| Edouard | doudou |



**POSTS**

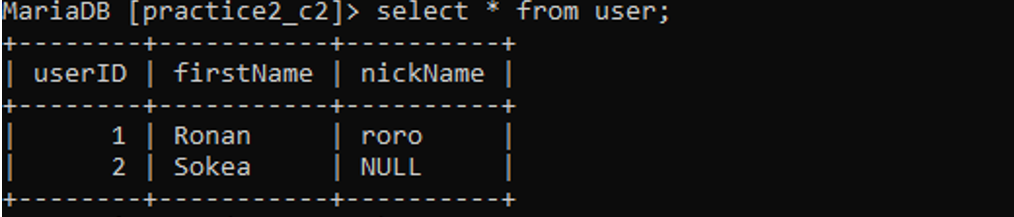
|  |  |
| --- | --- |
| Post body | From |
| Hello all ! | Ronan |
| I like rice | Ronan |
| YES YES | Sokea |



**Q5–** Write the statement to delete the user Edouard

* What’s happen? Can we delete it? Why?

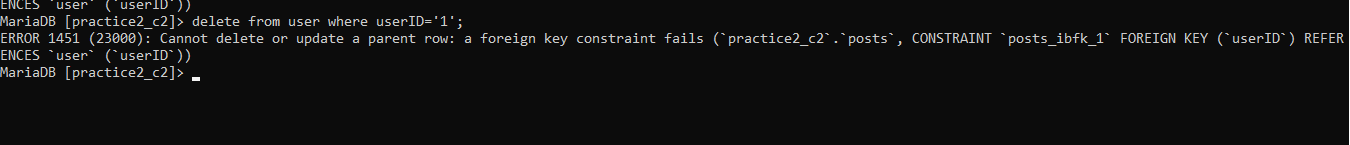
Yes, we can because it has row.



**Q6–** Write the statement to delete the user Ronan

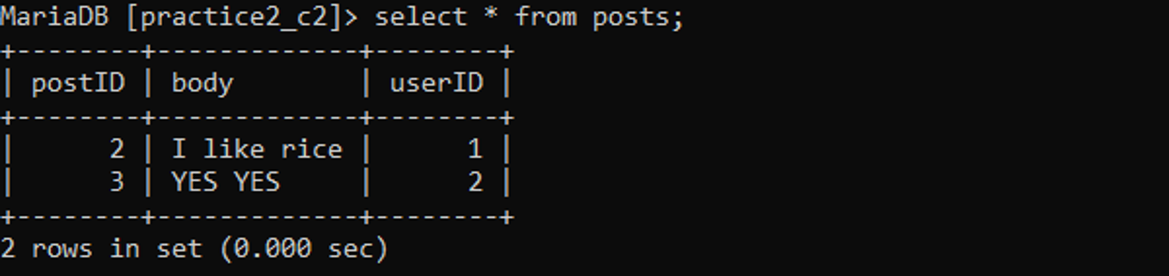
* What’s happen? Can we delete it? Why?

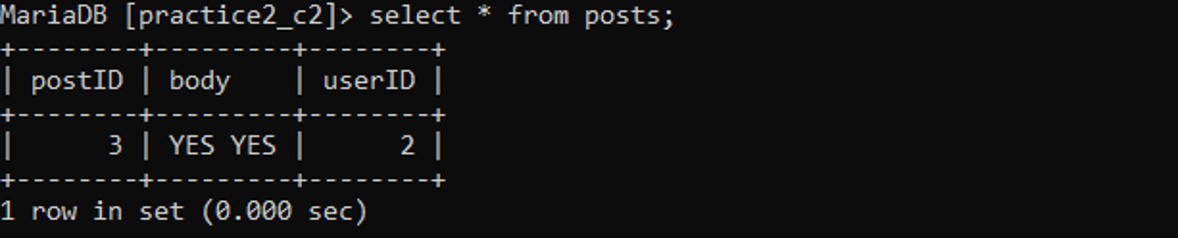
No, we can not because : you can see the picture on below



**Q7–** Write SQL statement to remove the rows related to Ronan user:

* Hello all!
* I like rice

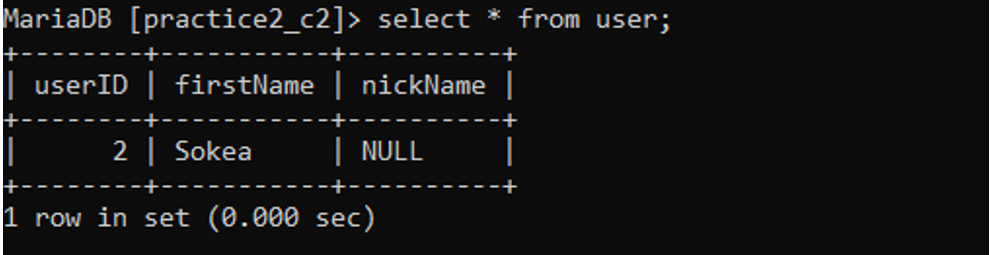


****

**Q8–** now try again to delete the user Ronan

* What’s happen? Can we delete it? What can you conclude?

Yes, I can



**Q9–** Add a new POST in the POST table with a userId which does not exist in the User table (ex: userID = 45)

* What’s happen? Why?

