|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Student ID Number**  *(Do not include student name as anonymous marking is implemented)* | Mateusz Utracki-Janeta | | | | | | |
| **Programme Title** | BSc Computing Technologies | | | | | | |
| **Module Title** | Data Modelling and SQL Language | | | | | | |
| **Module Code** *(listed on Moodle and in LTAFP)* |  | | | | | | |
| **Module Convenor** |  | | | | | | |
| **Coursework Title** | Assignment 2 | | | | | | |
| **Academic Declaration:**  *Students are reminded that the electronic copy of their essay may be checked, at any point during their degree, with Turnitin or other plagiarism detection software for plagiarised material.* | | | | | | | |
| **Word Count** | 1640 | | | **Date Submitted** | | **07/08/2019** | |
|  |  | | |  | |  | |
|  | | | | | | | |

Contents

[1. Data Manipulation Language (DML) 3](#_Toc15638908)

[1.1. UPDATE statement 3](#_Toc15638909)

[1.2. DELETE statement 4](#_Toc15638910)

[2. Data Control Language 5](#_Toc15638911)

[2.1. GRANT statement 5](#_Toc15638912)

[2.2. REVOKE statement 6](#_Toc15638913)

[3. Data Manipulation 6](#_Toc15638914)

[4. Database Testing 14](#_Toc15638915)

[4.1. Field size testing 15](#_Toc15638916)

[4.2. Null values testing 16](#_Toc15638917)

[4.3. Check constraints 17](#_Toc15638918)

[4.4. Referential integrity 19](#_Toc15638919)

[5. Appendix 23](#_Toc15638920)

[6. References 23](#_Toc15638921)

# Data Manipulation Language (DML)

## UPDATE statement

SQL stands for Standard Query Language and it is used to develop, manipulate and control the contents and the database itself. Data Manipulation Language is just one part of SQL used to as the name suggests manipulating data. One form of manipulating data is modifying existing value for that we use UPDATE statement.

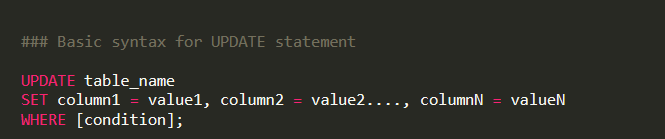


Figure 1

Adding WHERE to your statement helps filter your records by condition specified by you. But you need to be careful with that. Below is note from W3SCHOOLS you should be aware of.



Figure 2 (W3S UPDATE, 2019)

Let’s say for example we have a database of employees. Database is managed by DBMS and one day employee calls to inform us that her last name changed because she got married. We would use UPDATE statement, to modify the data given, using employees ID number as an indicator of that employee, we will add WHERE to our statement, to condition the update. Below is the sample code based on the table from brief to this assignment.

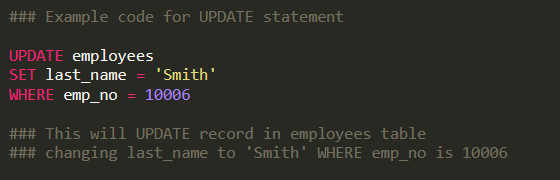


Figure 3

## DELETE statement

Another form of manipulation language is DELETE statement and it is really as it says in its name about deleting data. Basic delete statement syntax is showed below:

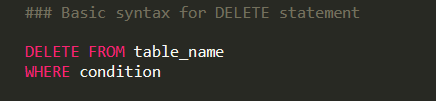


Figure 4

You can modify the delete statement with WHERE to filter what you about to delete, but you need to be careful about it because:

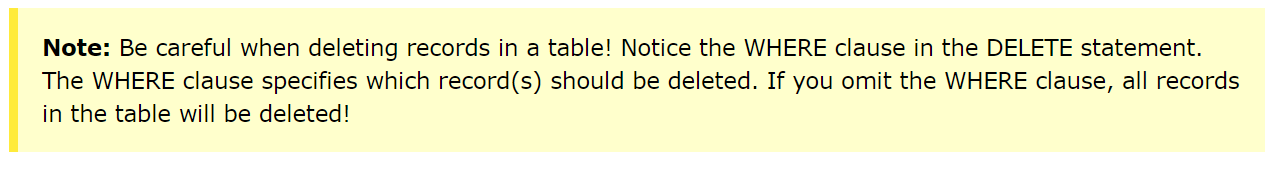


Figure 5 (W3S DELETE, 2019)

As an example, we can say that a customer decided he is going to stop using our online store or application and requested to delete all his data from the database, knowing the email address and unique id, we can delete record. Below is example SQL code of DELETE statement.

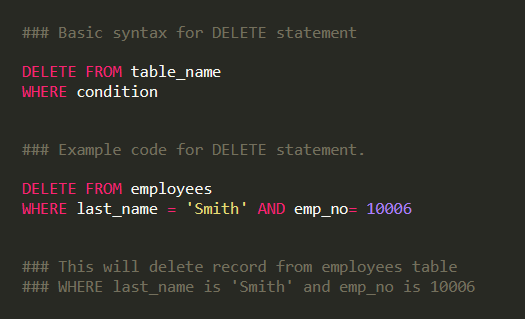


Figure 6

# Data Control Language

Data Control Language (DCL) is another part of SQL spectrum. DCL is used mainly for control and monitor purposes. I will explain two DCL statements GRANT and REVOKE below.

## GRANT statement

As a creator of an object or relation you have all the privileges for this object and by so you can grant someone the right to issue any query on your table or UPDATE, DELETE or INSERT any or specific value in it. You can also grant privileges with option to pass them by the user who was granted by you. Example syntax screenshot is below.

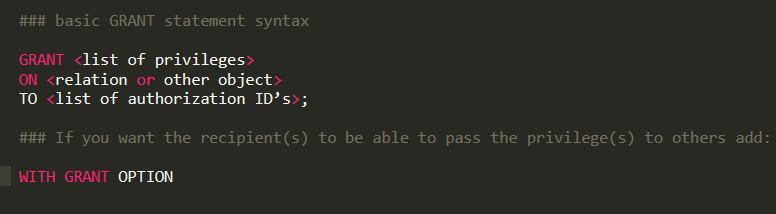


Figure 7

Lets see GRANT statement in example based on the database given in the assignment brief. First let’s create a user with Username - ‘Matt’, for that I will write:

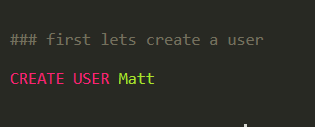


Figure 8

Now let’s assume I am the owner of employee table and I want to GRANT user Matt some privileges I would type something like this:

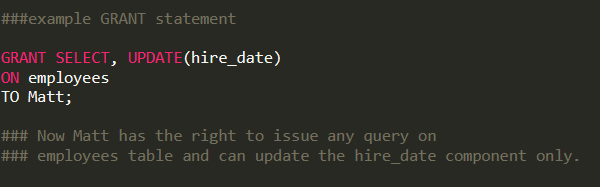
‘

Figure 9

## REVOKE statement

REVOKE statement is the reverse of GRANT statement it is used to invalidate given privileges. What it means that users can longer justify their privilege by your grant, but they can still have privileges if they obtained it independently elsewhere. (QAHE, 2019) Below is syntax for REVOKE statement.

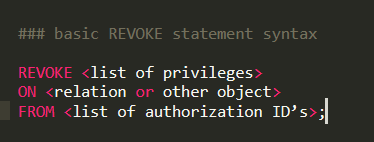


Figure 10

Based on GRANT example above I am now going to REVOKE the privileges I have given to user Matt on employees table.

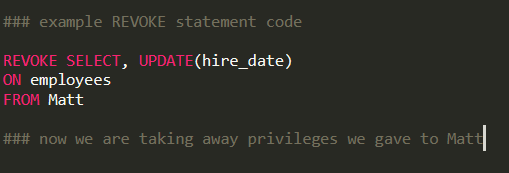


Figure 11

# Data Manipulation

1. Write an SQL statement to list all employees with their full names, gender and salary

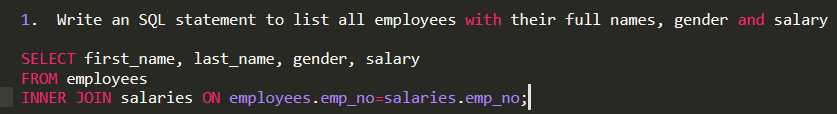


Figure SQL syntax for first query

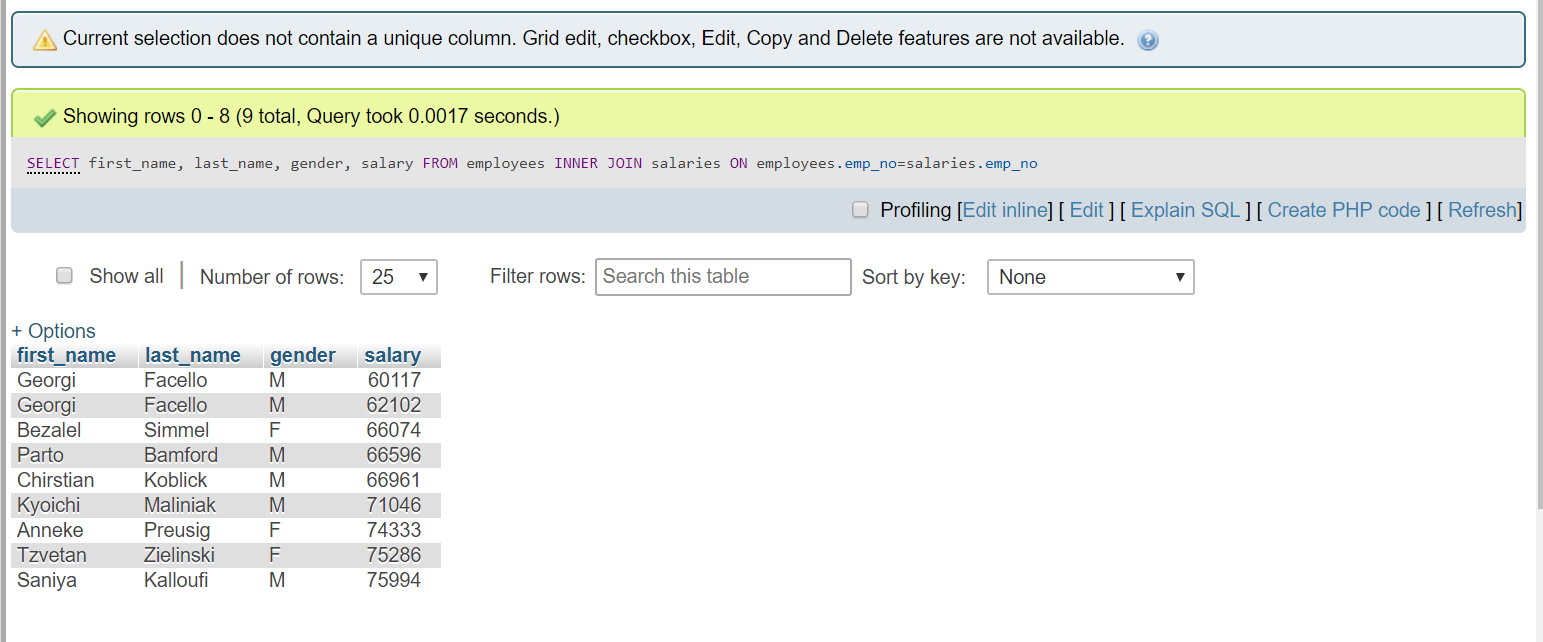


Figure outcome of first query

1. Write an SQL statement to show the title of all employees and their department name.

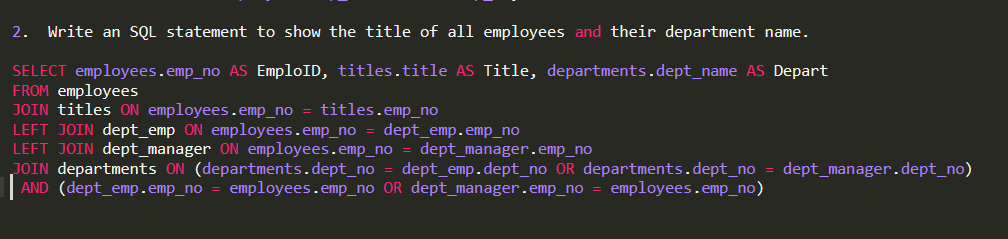
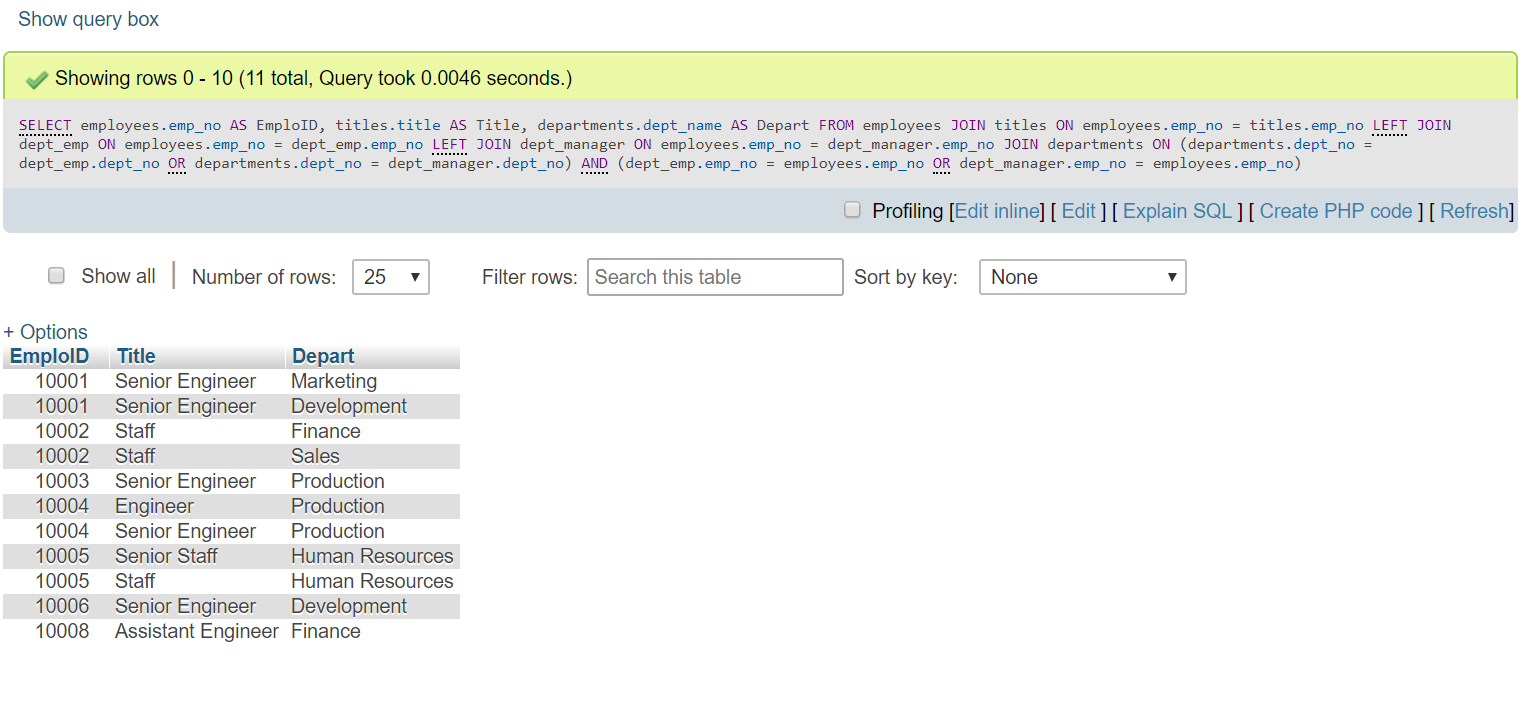


Figure SQL syntax for second queryFigure outcome of second query

1. Write an SQL statement to show the full names and gender who belong to department number 'd004'.

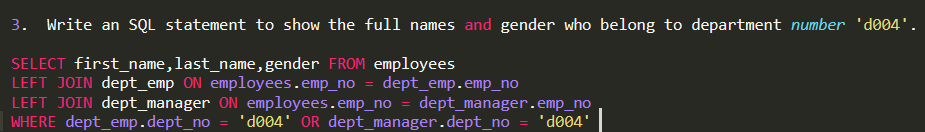


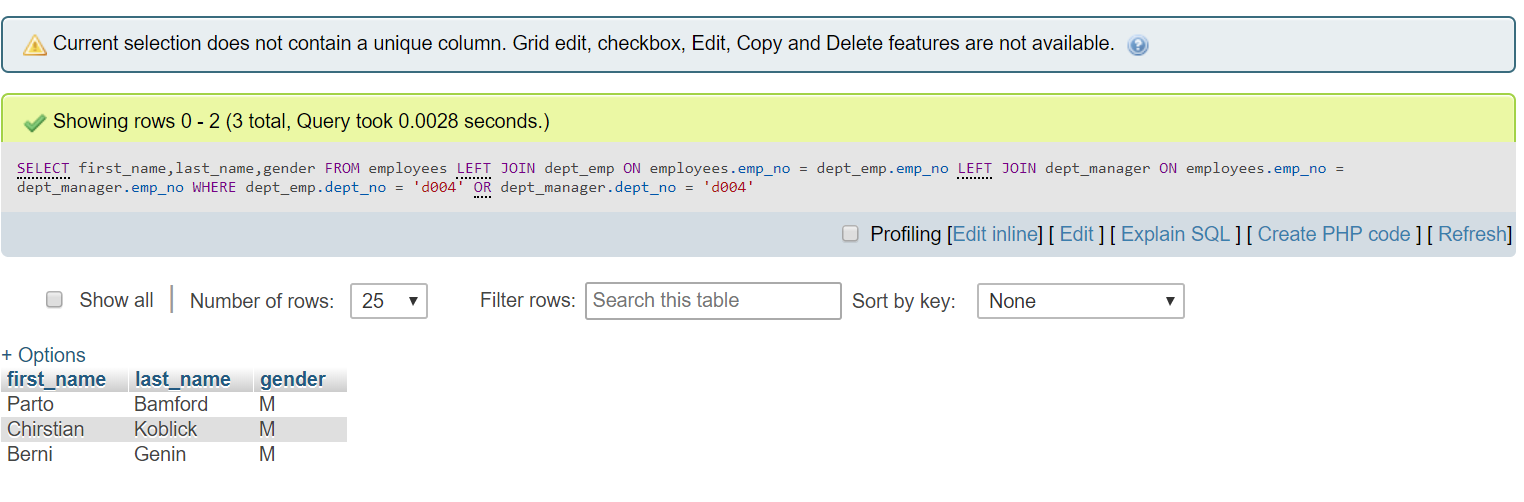
Figure SQL syntax for third query

Figure Outcome of the third query

1. Write an SQL statement to show the all departments and their department’s managers.

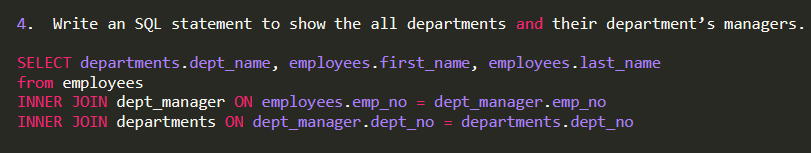


Figure SQL syntax for the fourth query

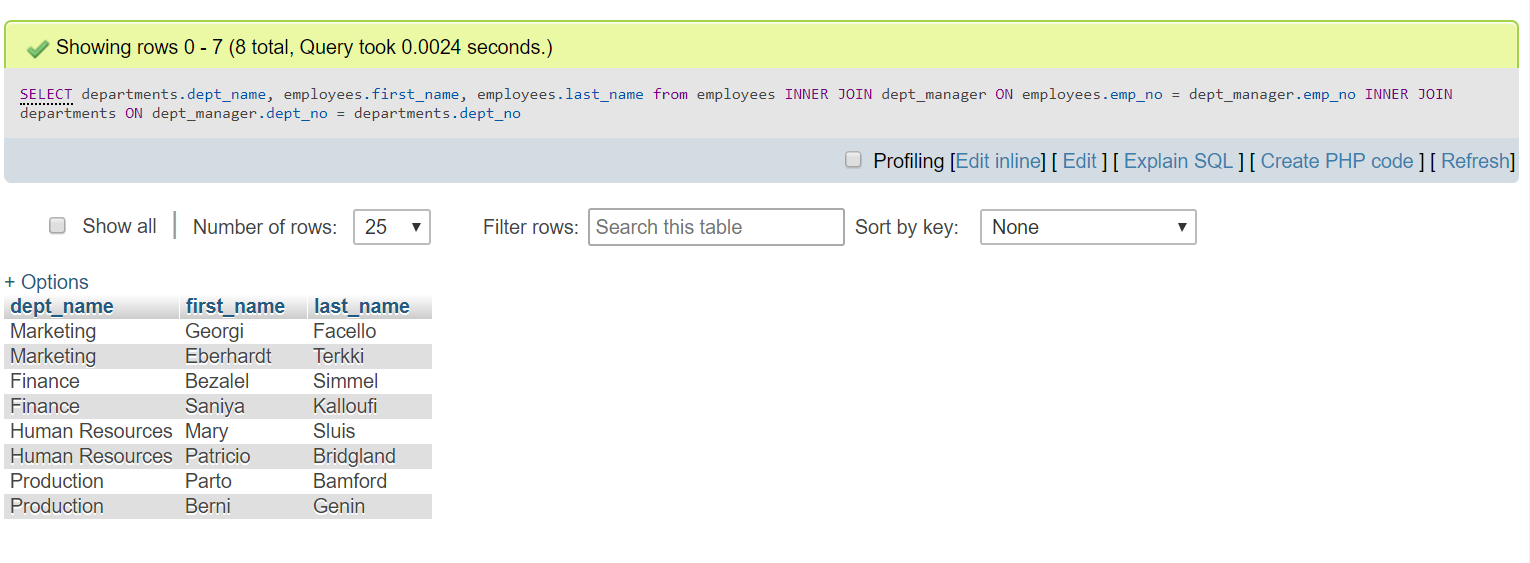


Figure Outcome of the fourth query

1. Write an SQL statement to show a list of department’s managers who were hired after 1990

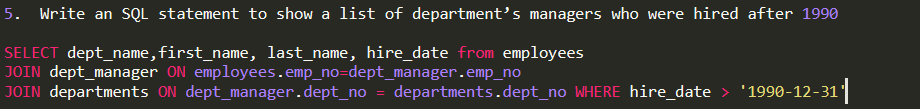


Figure 20 syntax for the fifth query

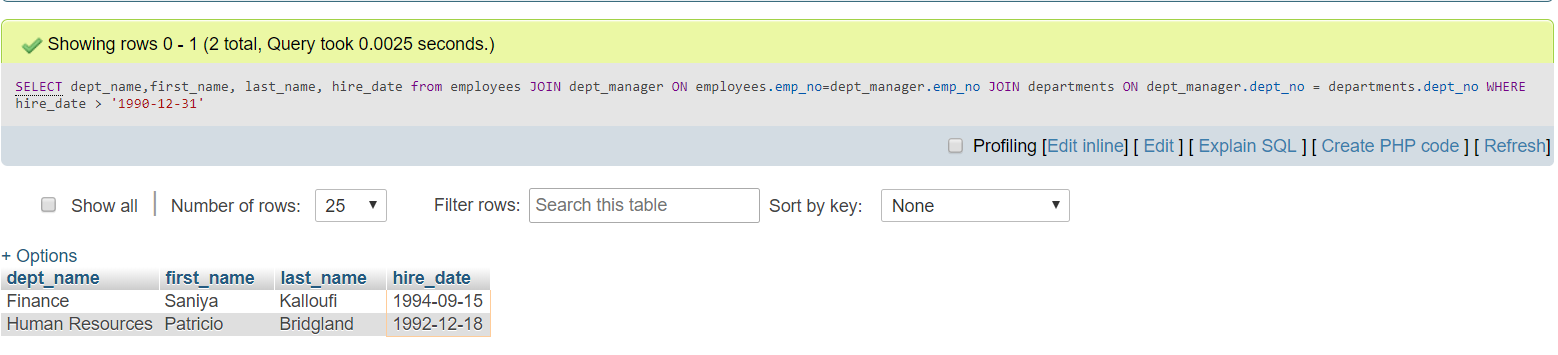


Figure 21 fifth query outcome

1. Write an SQL statement to change any employee’s date of birth. Assume the employee has just phoned in with his/her last name.

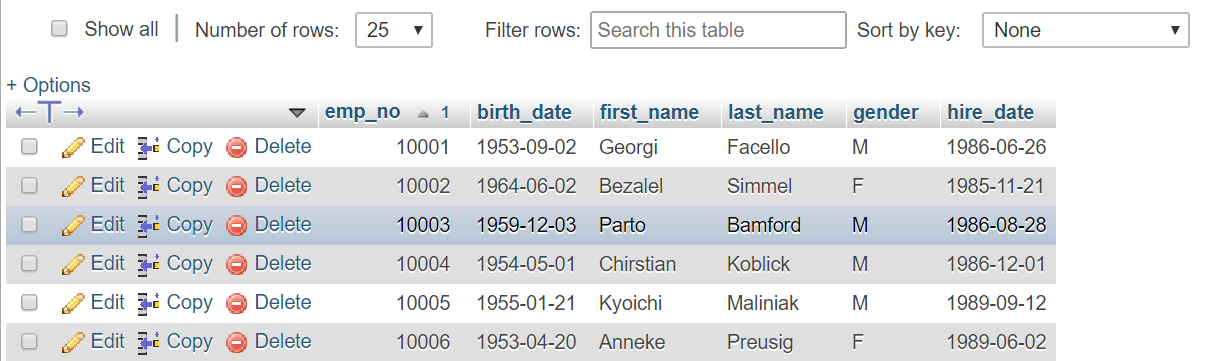


Figure employee table before changing birth\_date of emp\_no=10002

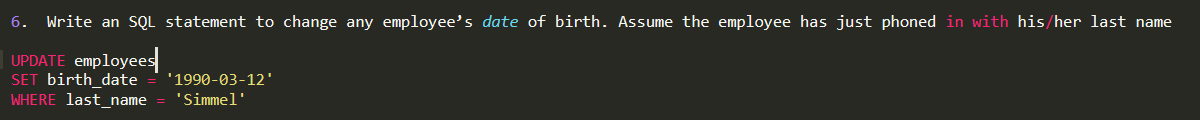


Figure SQL syntax for the sixth query

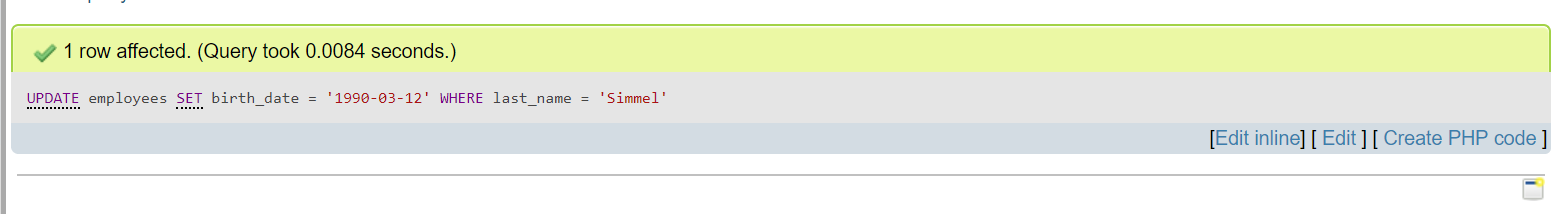
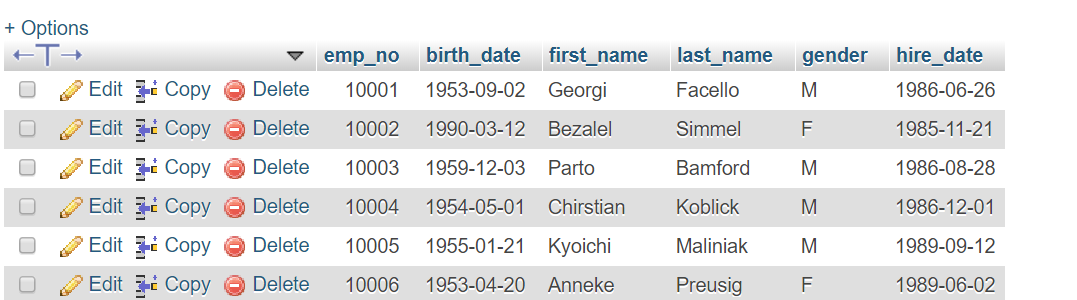
Figure query success

Figure employee table after changing the birth\_date of emp\_no = 10002

1. Write an SQL statement to increase salaries of all employees up to 5.5% who are working from 1997.

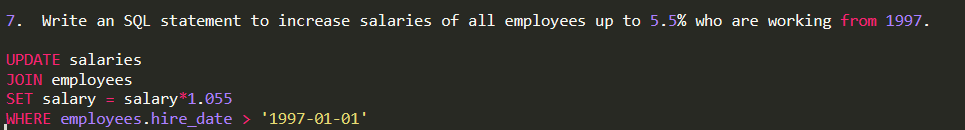


Figure SQL syntax for seventh query



Figure Outcome of seventh query

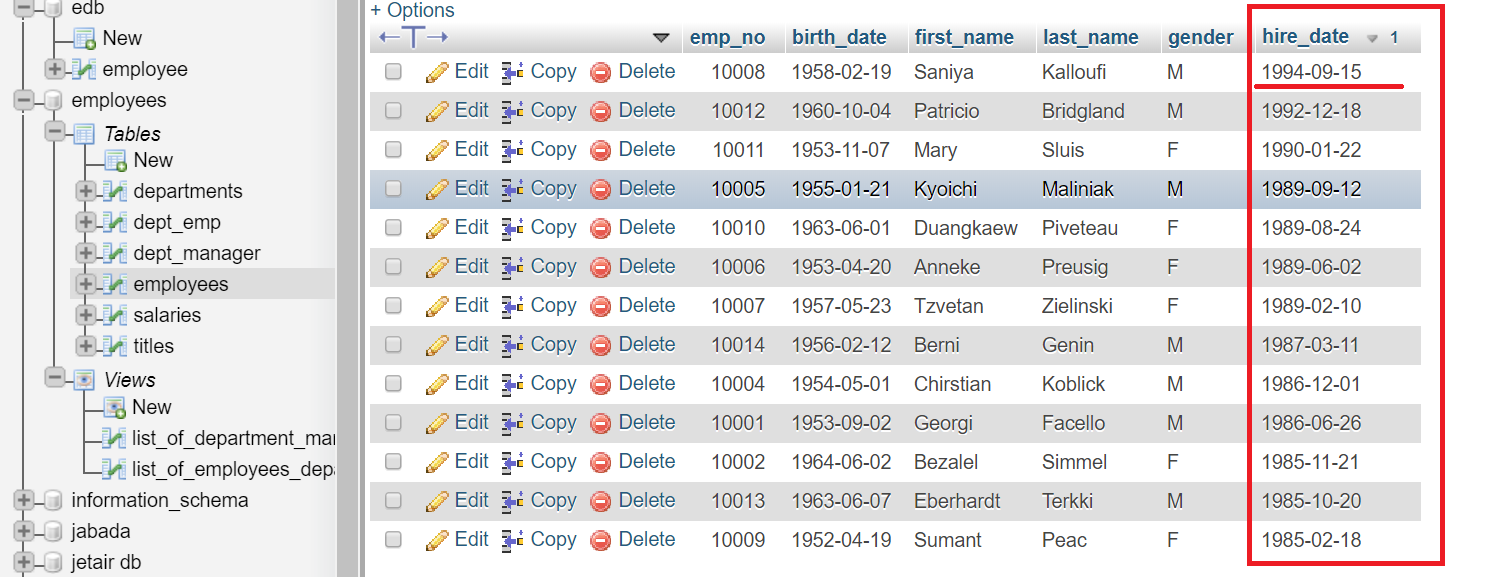


Figure employees table with hire\_dates in descending order to show that there are no employees hired after 1997

1. Write an SQL statement to delete employee’s record who belongs to department 'd004' and ID is 10003.

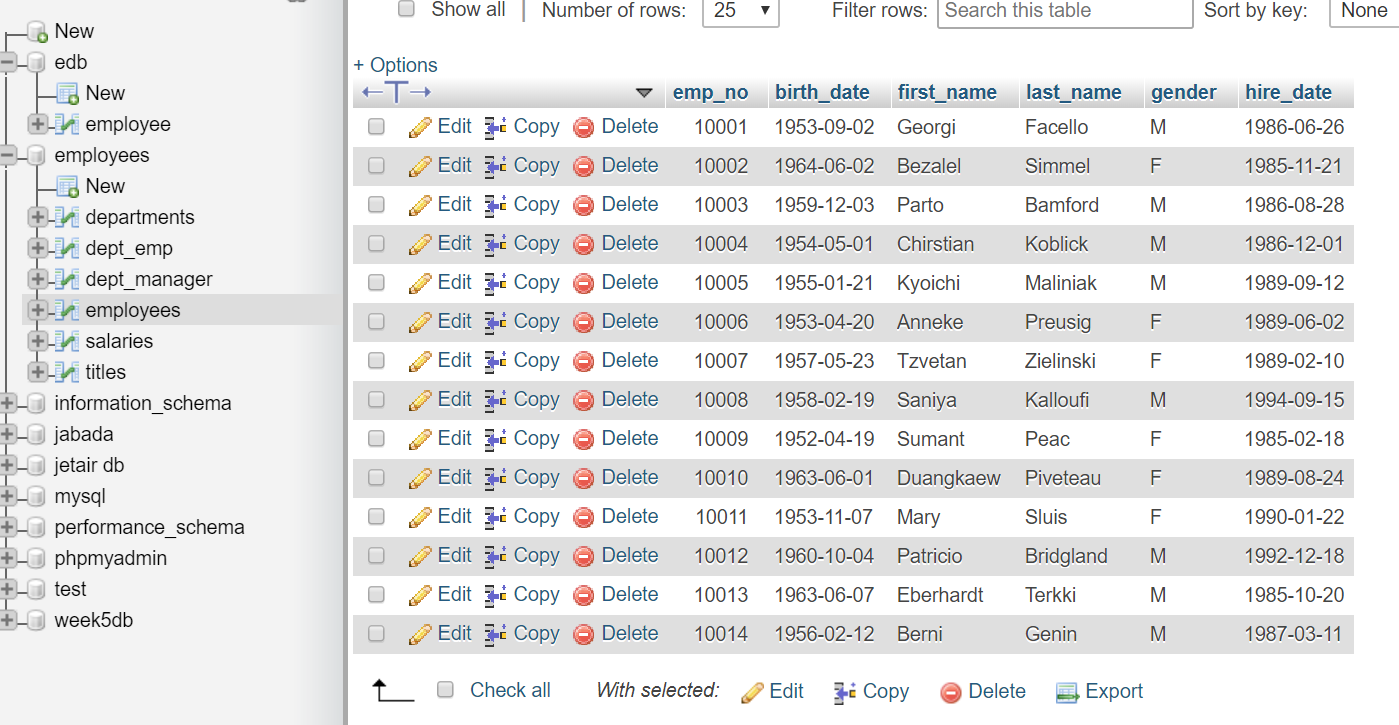


Figure employees table before deleting one row

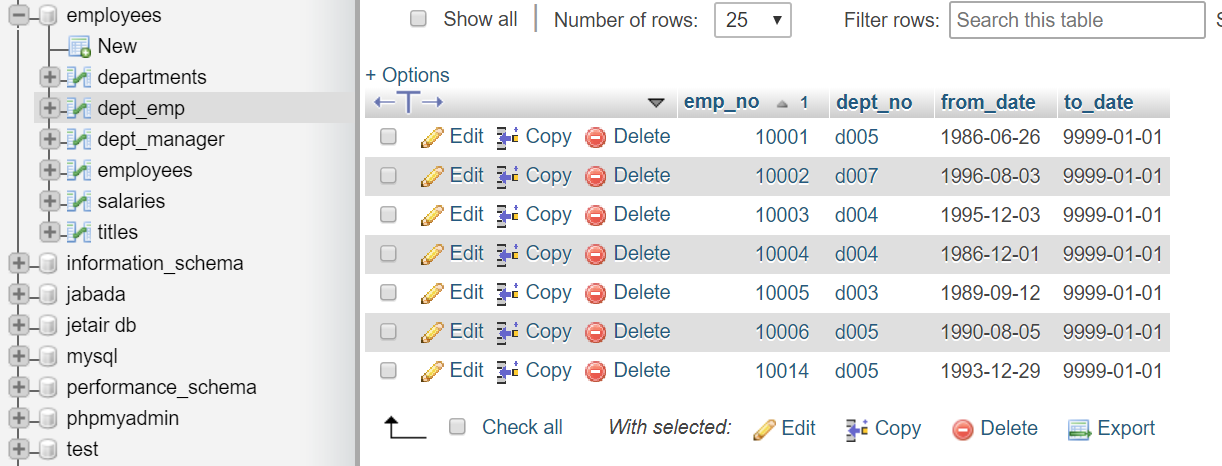


Figure dept\_emp table before deleting one row

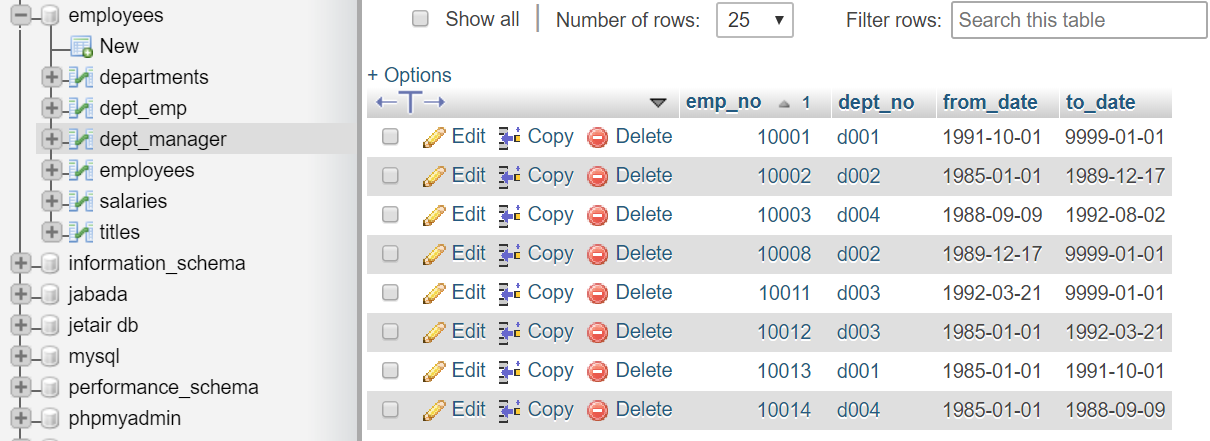


Figure dept\_manager table before deleting one row

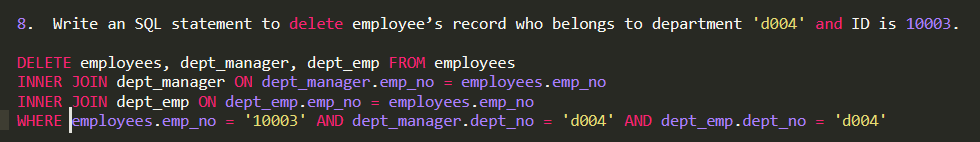


Figure SQL syntax for eighth query

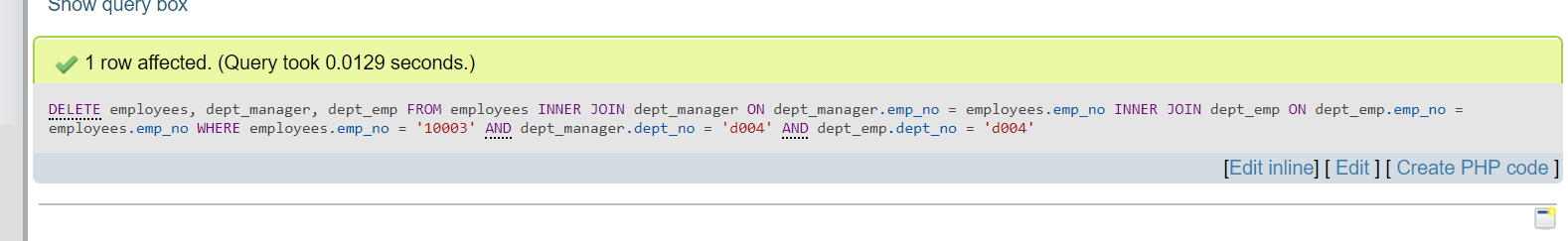


Figure query success message

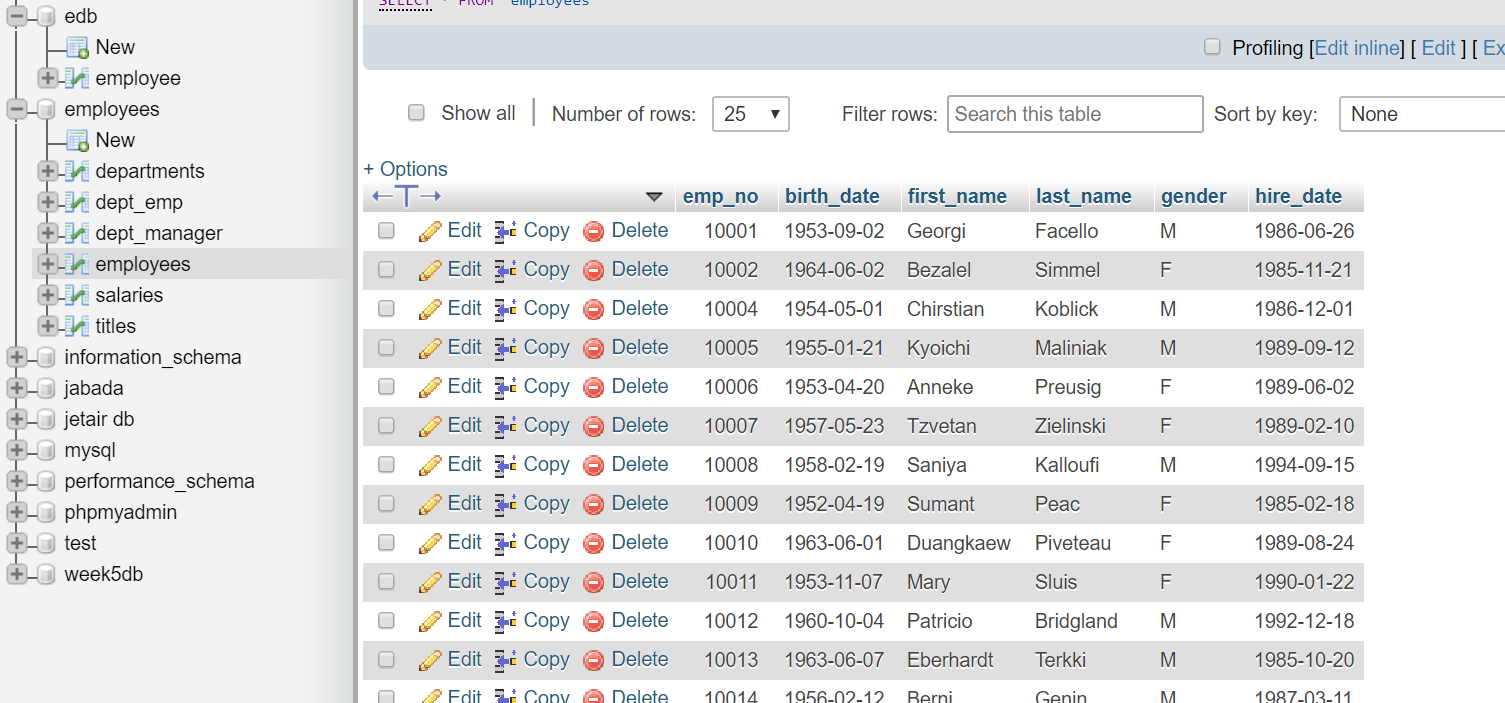


Figure employees table after deleting one row you can see now that there is no employee with emp\_no = 10003

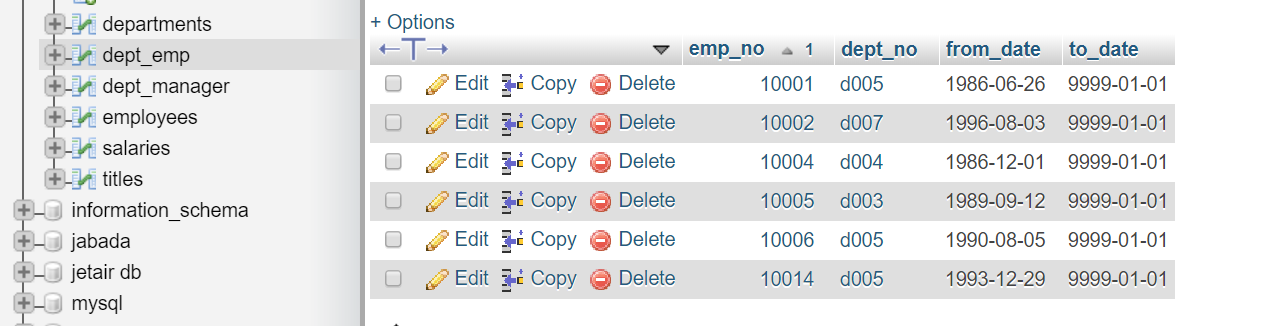


Figure dept\_emp table after deleting one row

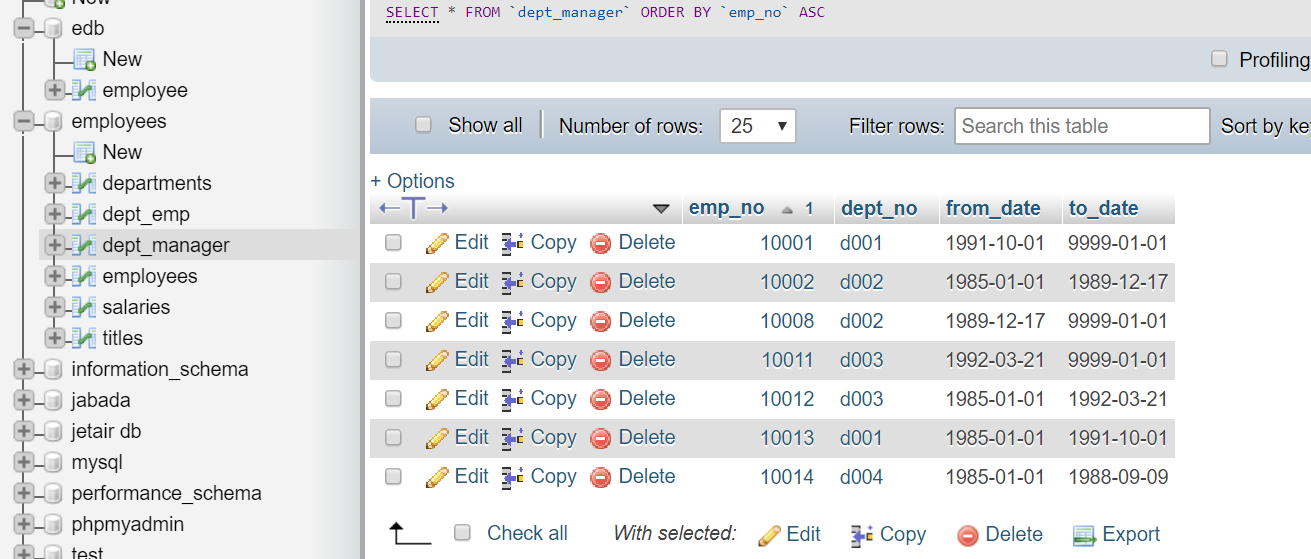


Figure dept\_manager table after deleting one row

1. Create a database **view** to list full names of all employees their departments and salaries.

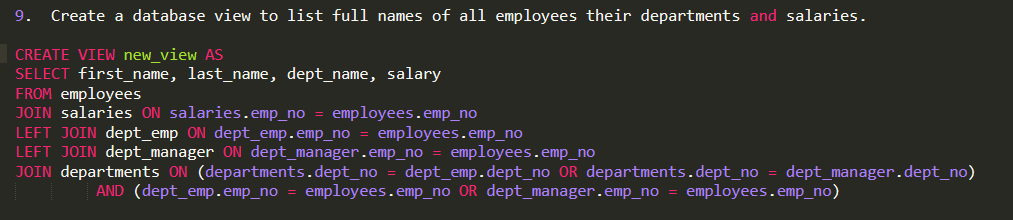


Figure SQL syntax for the view asked for in task 9

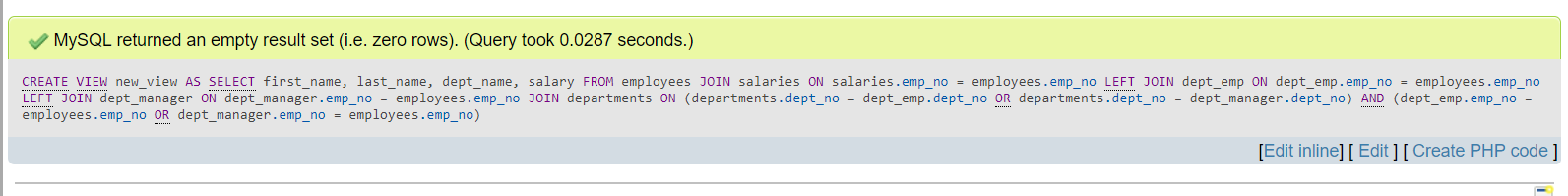


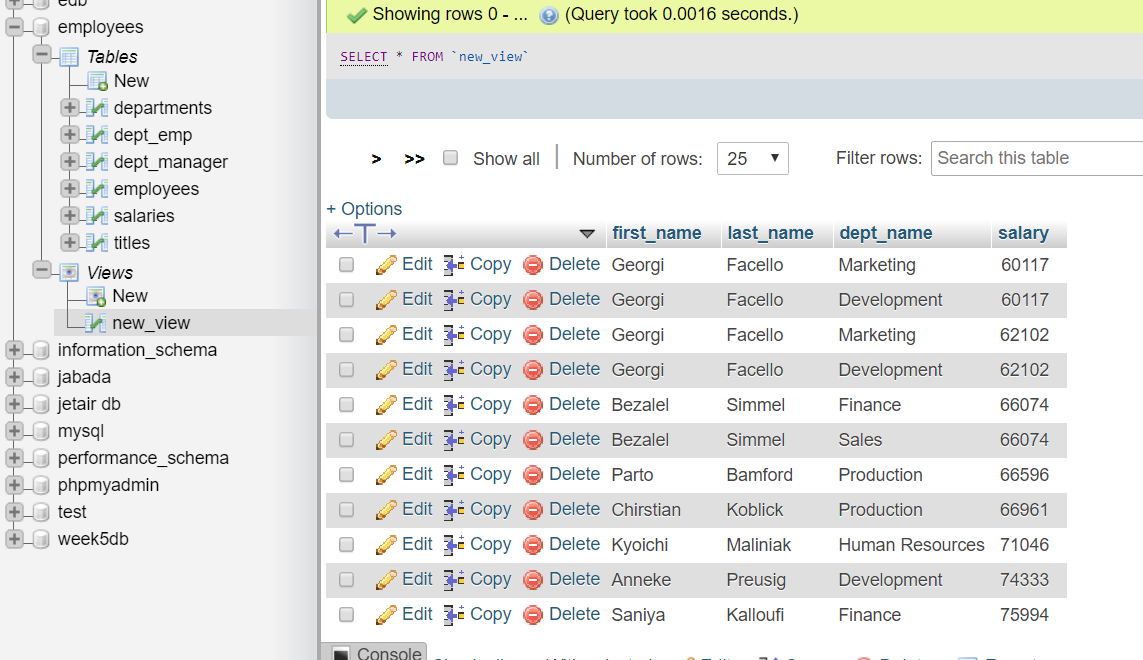
Figure view success

Figure screenshot of the view from task 9

1. Create a database **view** to list all departments and their department’s managers, who were hired between 1990 and 2000.

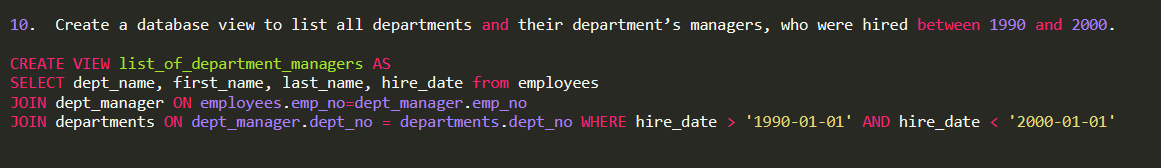


Figure SQL syntax for the view asked for in task 10

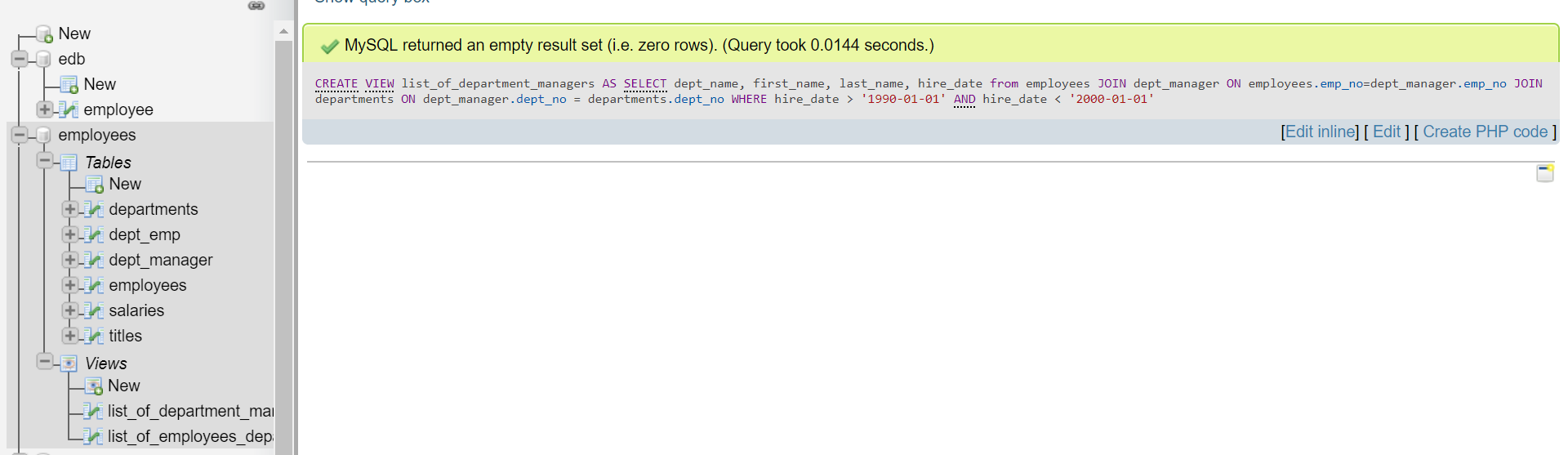


Figure view success

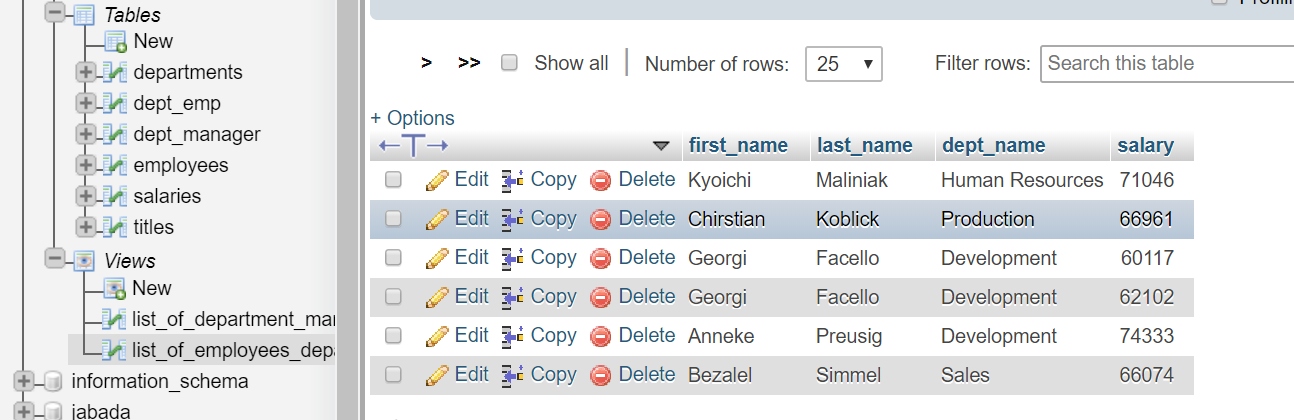


Figure screenshot of the view outcome from task 10

# Database Testing

Testing is an important part of any development process. It assures that process is replicable, free from errors and ready to be implemented. Like a chef before serving a dish tastes it, as a form of test, if it complies with his taste standards it is being served. Testing the database, tests if all our assumptions to the relations where right, if we declared data types and constraints correctly, etc. We can distinguish three categories of database testing:

* Structural Database Testing – it about testing the database schema, table and column testing, checking triggers, stored procedures and views testing, etc.
* Functional Testing – Testing the database from users’ point of view by checking its functionality, the most common types are white and black box testing. In white box testing the internal structure of the item being tested in known to the tester, in black box it is not.
* Non-Functional Testing – it includes stress testing, risk testing in database, load-testing, minimum system requirements and it addresses performance of the database

If we don’t do it the database may crash and with it our application or website, this can mean that our business won’t operate for some time, resulting in losing money.

## Field size testing

Every field in database can store certain number of letters, numbers or symbols or a mixture of all of them. It is very important to put constraints on fields because, each letter, symbol or number can take one byte of storage space, so if you leave it on default on put no constraint at all someone can crush your DB. Field size testing involves entering values that break the constraint.

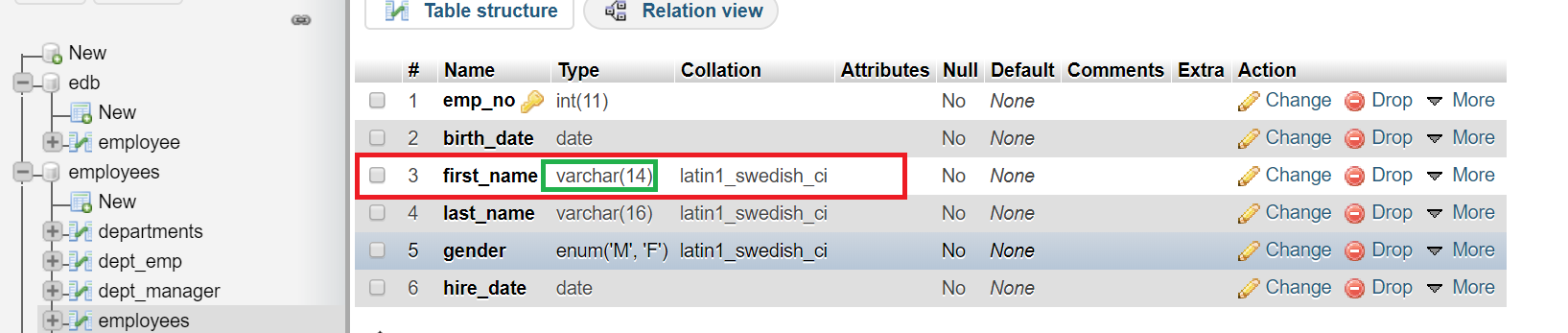


Figure Structure of employees table with constrained varchar datatype

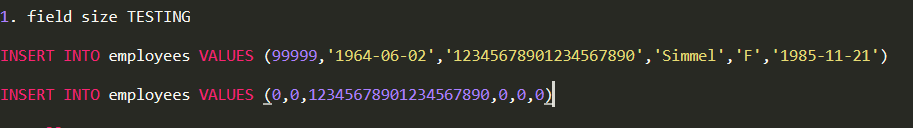


Figure syntax for field size testing, entering 20 signs first name

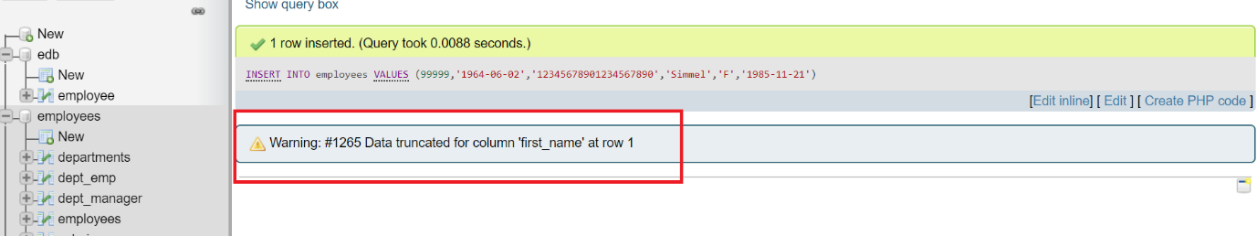


Figure Warning message about constraint

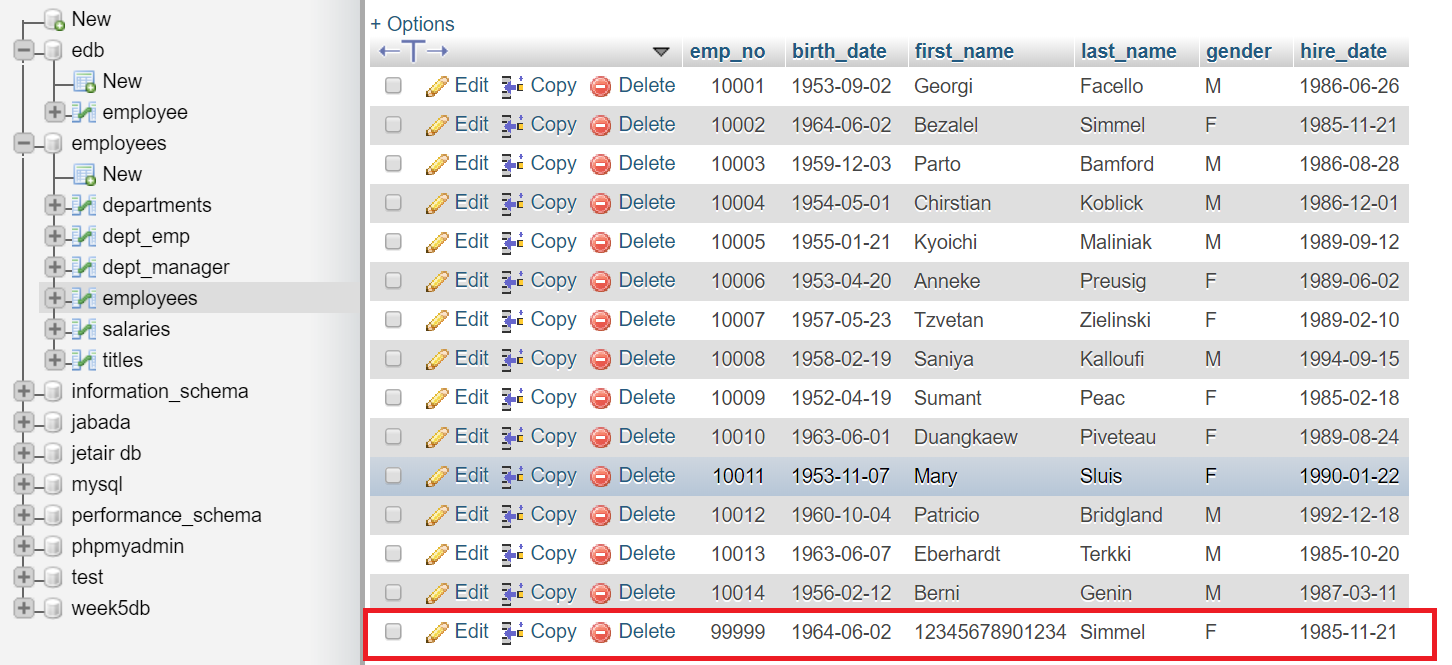


Figure field truncated to 14 characters

## Null values testing

Null is a special marker used in SQL to show that the value doesn’t exist in database. It was created by the creator of relational database model E.F.Codd. It shouldn’t be mistaken with ‘0’ or an empty string, these are not null values. Null values testing shows if the not null constraints are correctly formed and the table doesn’t accept this kind of values. Do to so we just need to INSERT into any tables VALUES which are NULL. Example below shows it.

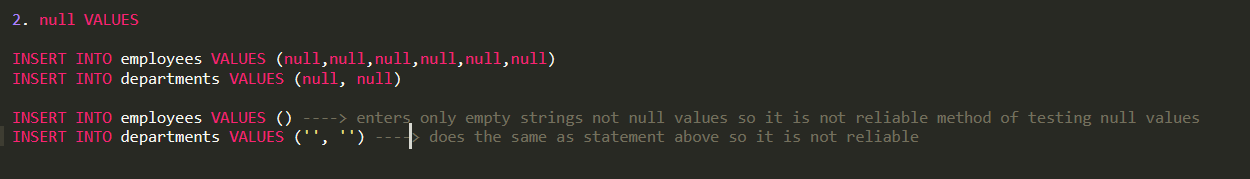


Figure syntax for null values testing, bottom is example of wrong syntax

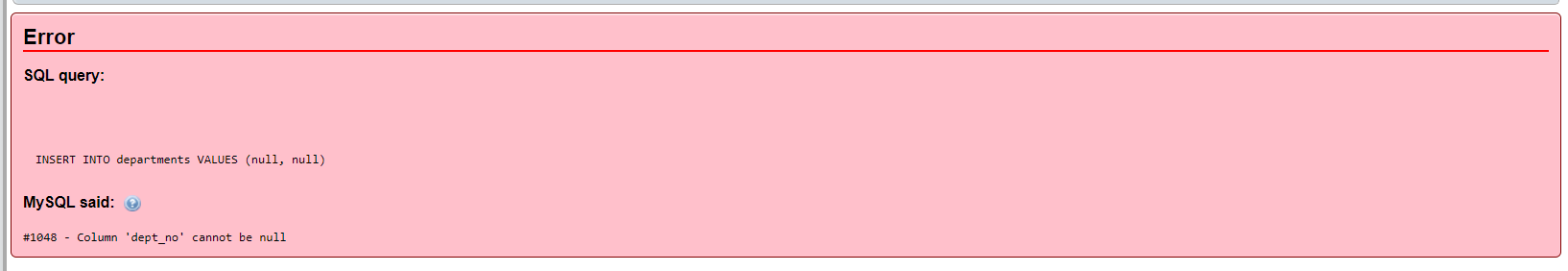
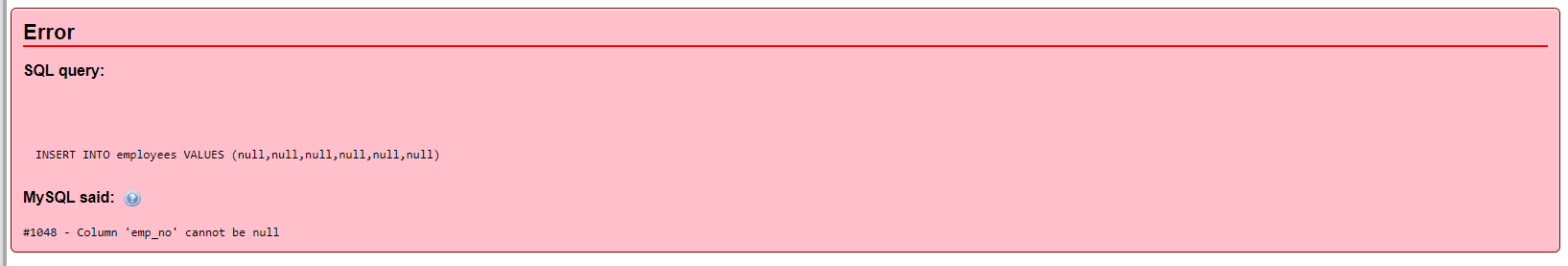
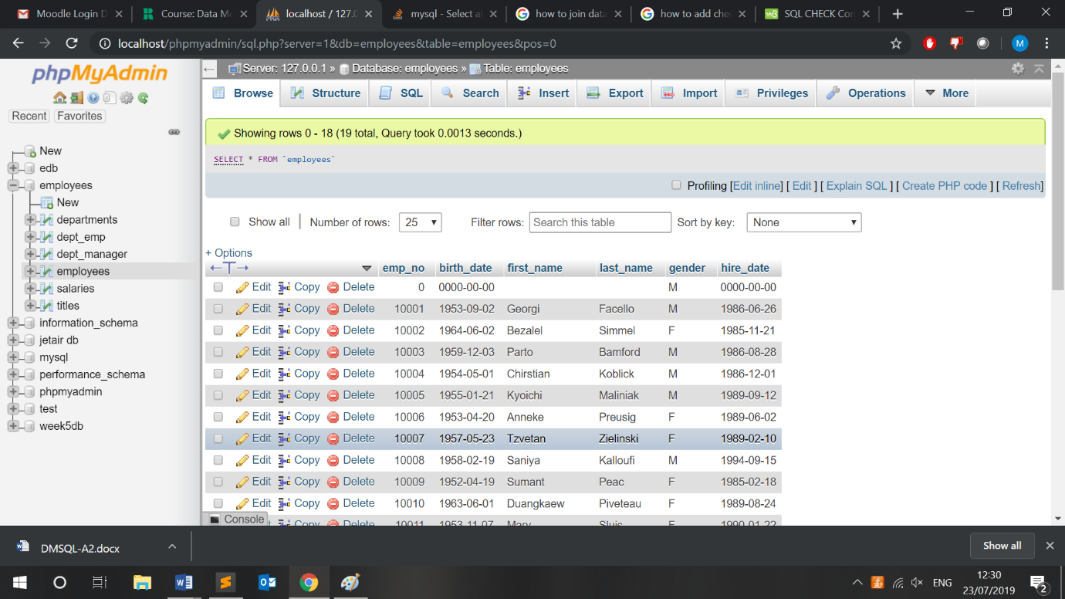
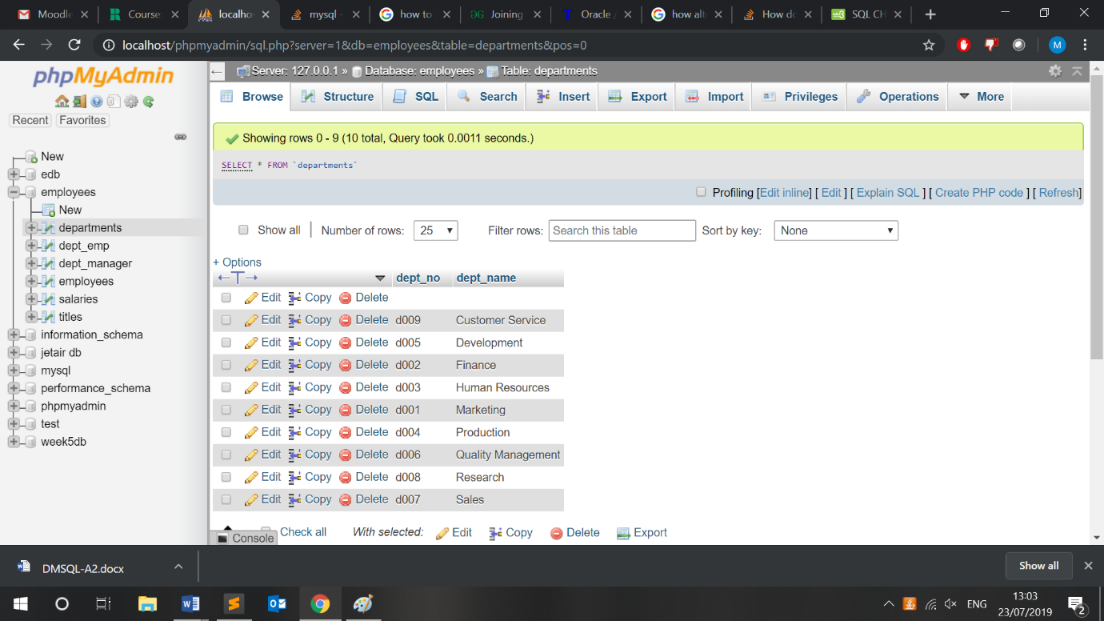


Figure error message if entering null values

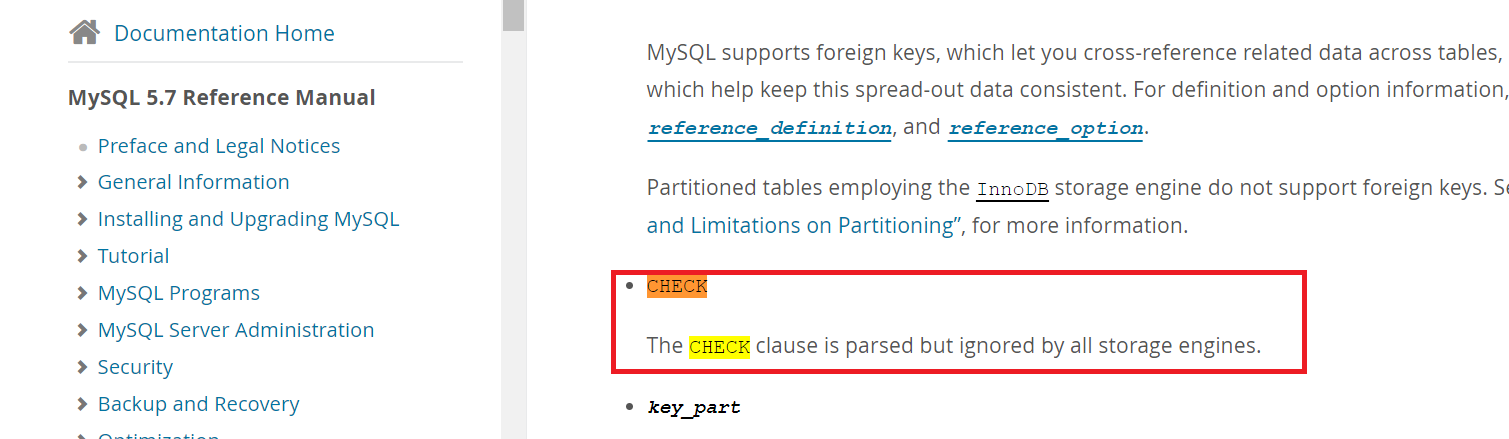
Error message above shows that NOT NULL constraints are formed correctly and don’t allow null values to be entered. Below is the output of the wrong null values testing which can be misleading. Inside cells on top there are not null values but empty strings!



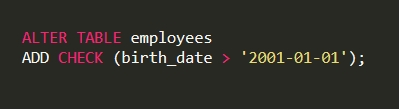


## Check constraints

SQL standard enables you to validate data for a column or group of columns before inserting and updating. We use the CHECK constraint to do that. Unfortunately, the CHECK constraint is not working in MySQL server we are using for this DB, you can add constraint, but it is being ignored. (Oracle Corp., 2019)



There is a possibility to do that what CHECK constraint creating stored procedure and then using triggers or views. (MySQL TUTORIAL, 2019) Images below show altering the table to add constraint.



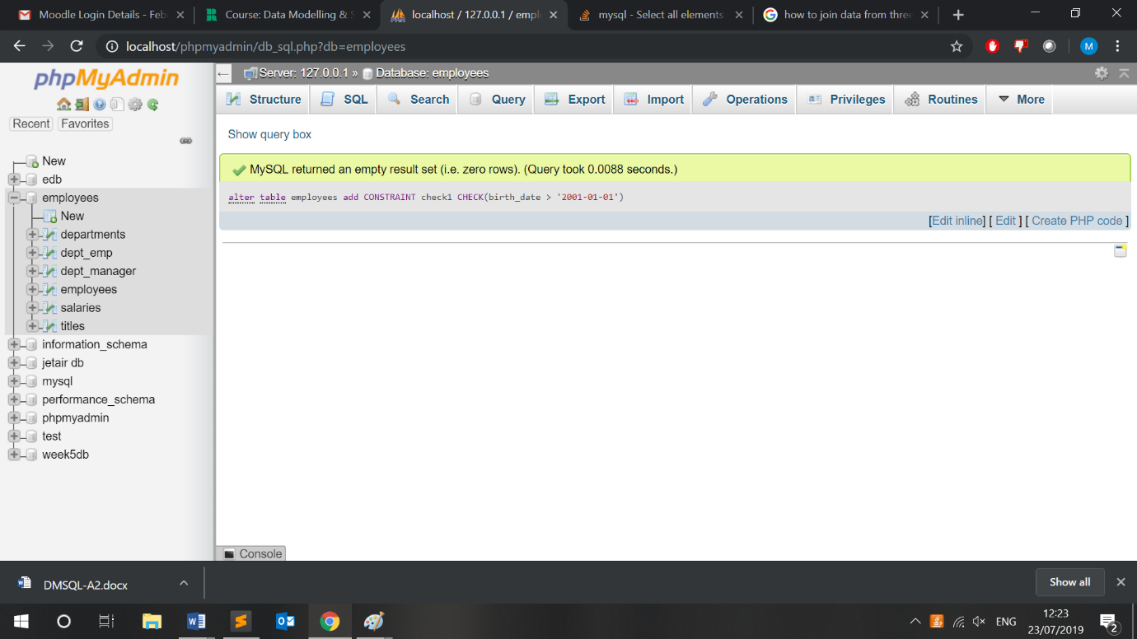


Figure 49



Figure 50 shows that values we insert are violating check constraint.

Figure

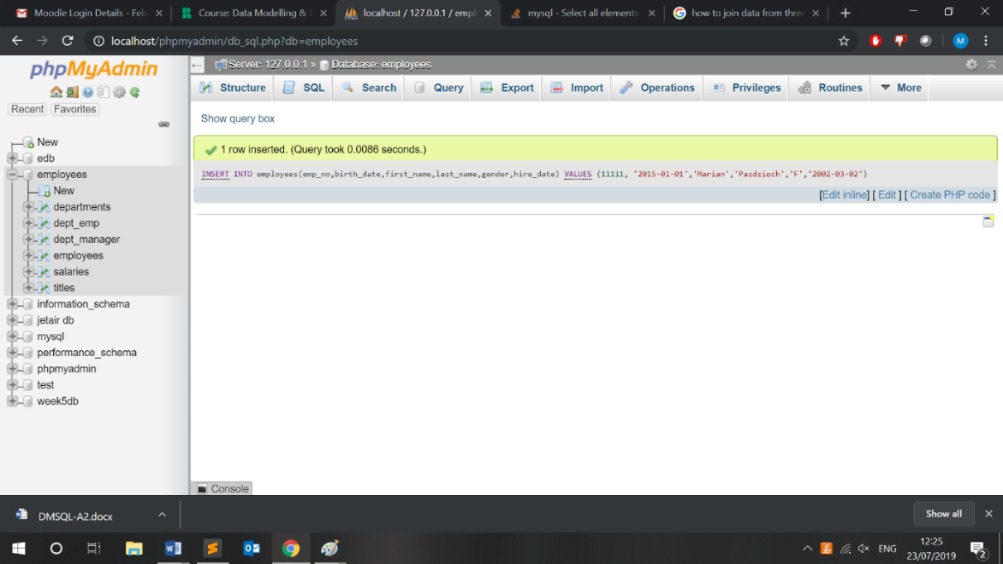
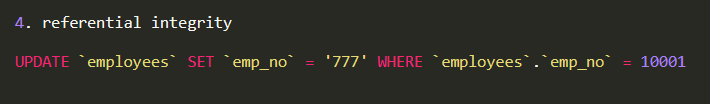


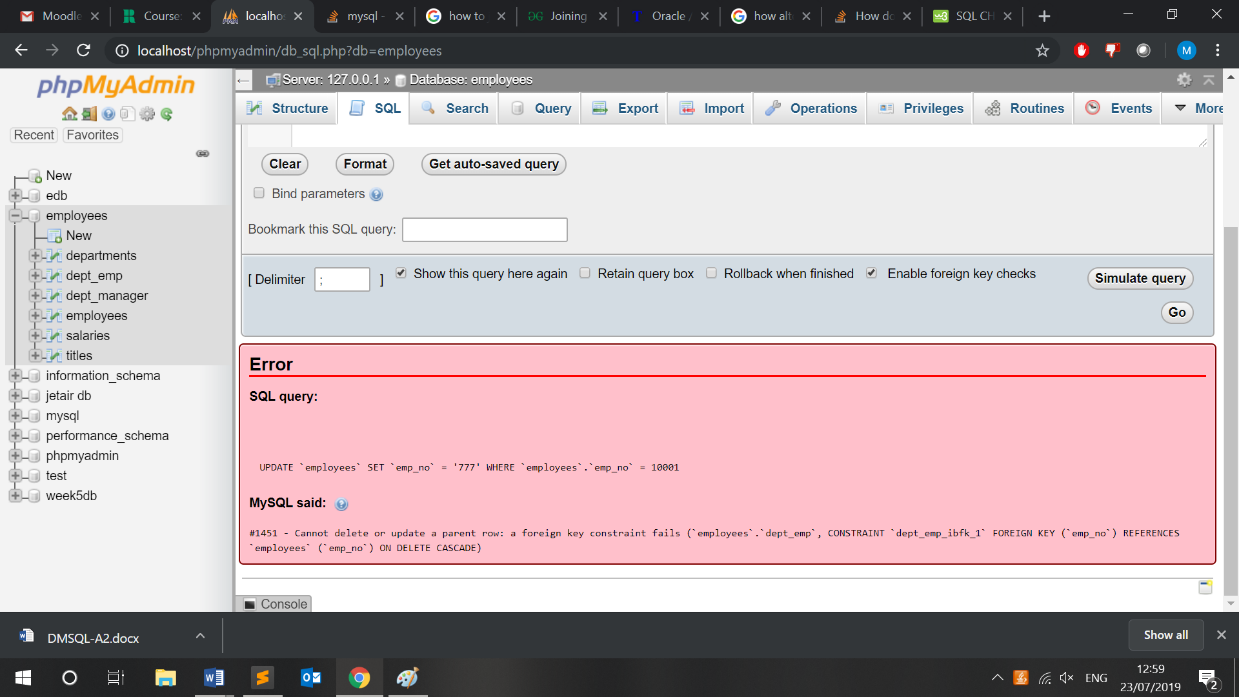
Figure 51 Successful addition of a person under 18 years old

## Referential integrity

Referential integrity relates to a relationship's precision and consistency of data. Data is connected between two or more tables in relationships. Referential integrity therefore needs that it must reference a valid, current main key in the parent table whenever a foreign key value is used.

So first we will check if referential integrity on update exists:





Figure

So, we need to change the DDL code so it will update records when changed. We will add this snippet of code: 



Figure before changes

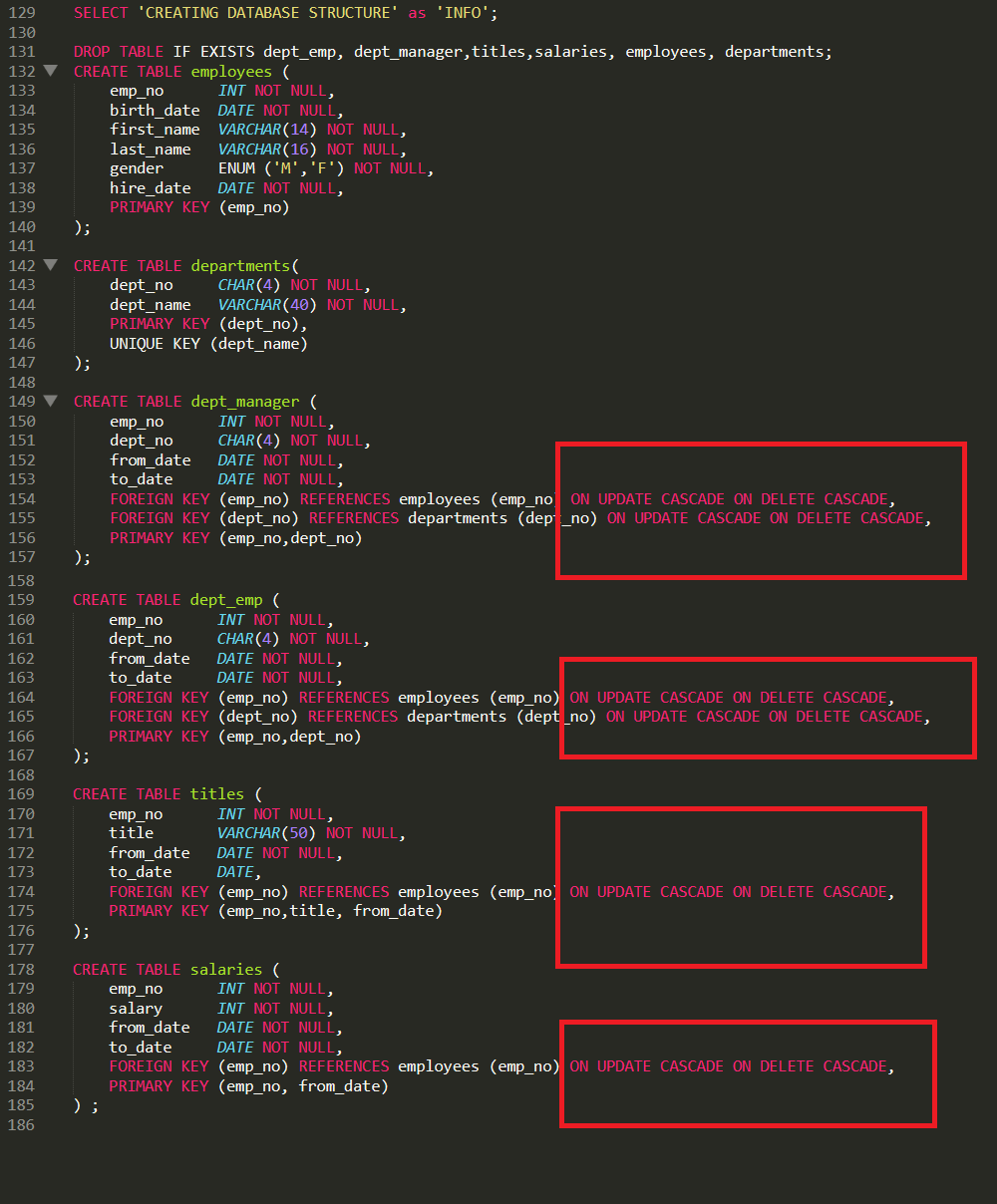
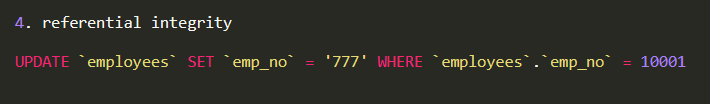
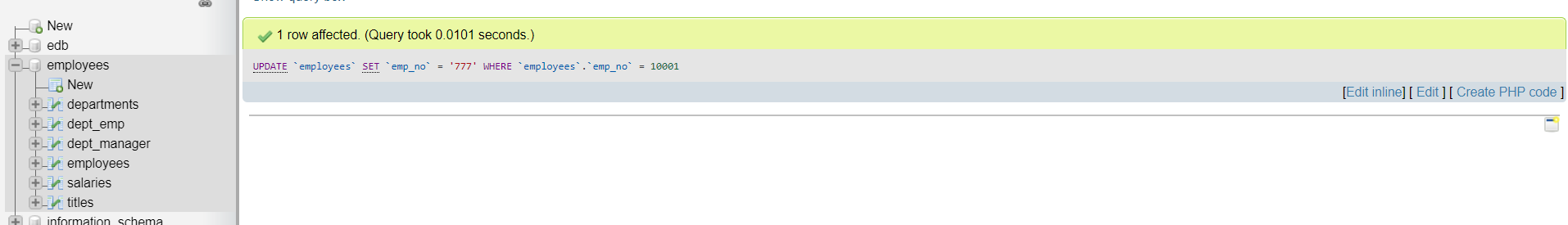
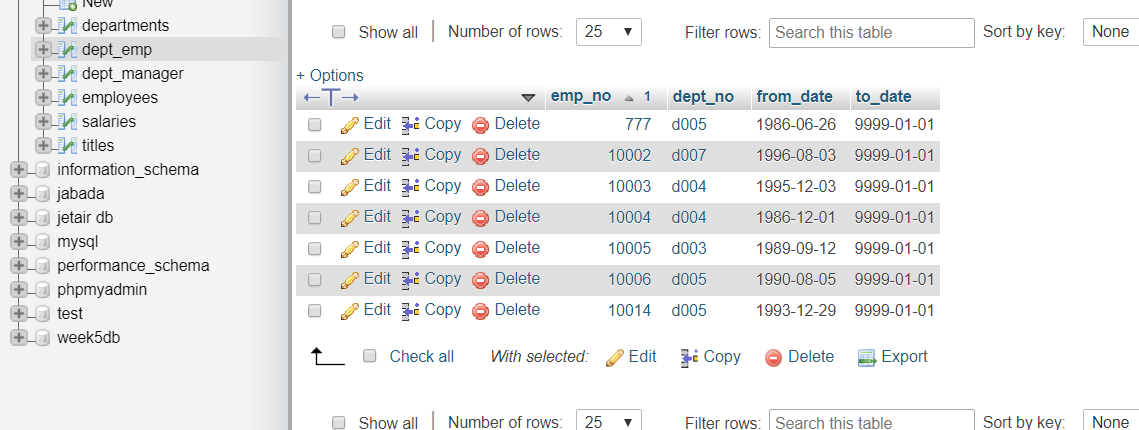


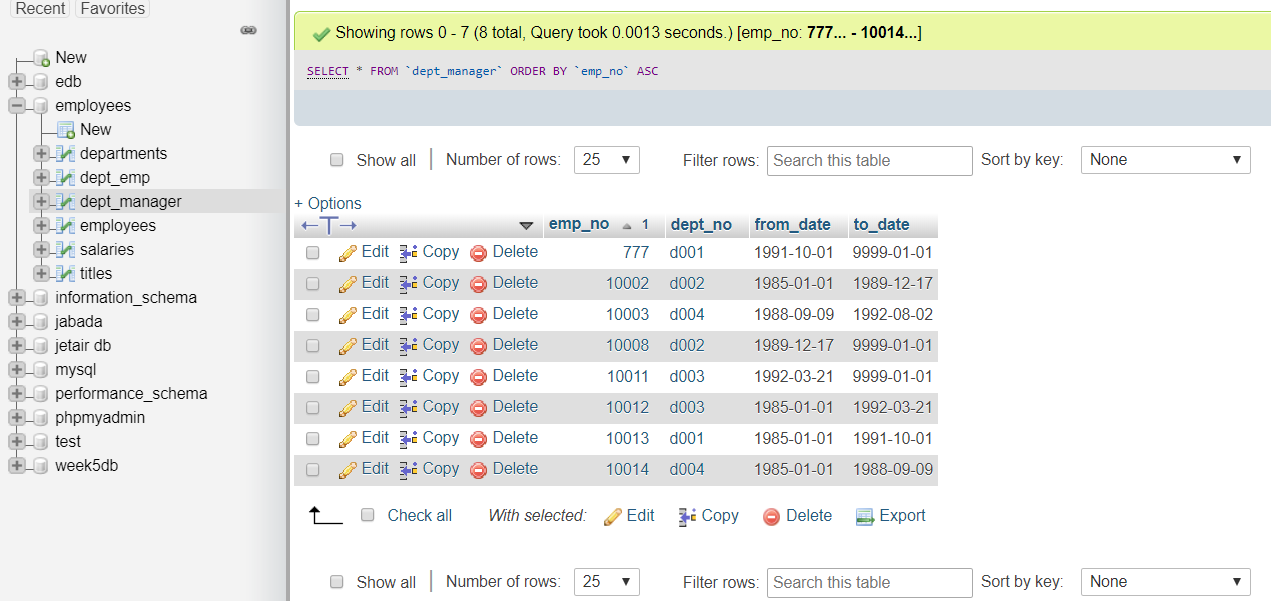
Figure after changes

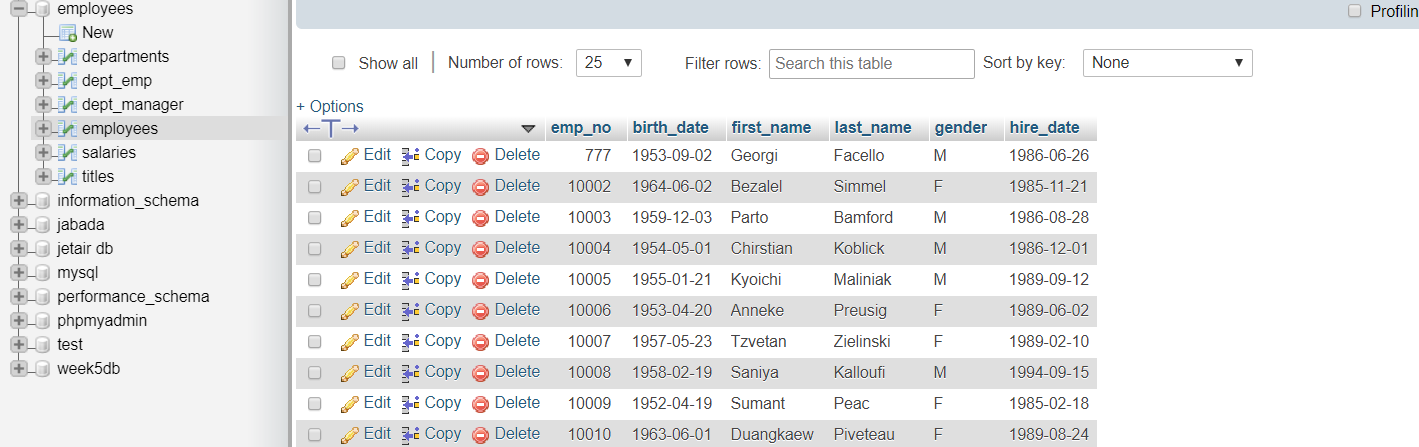




Screenshots confirming changes:







# Appendix



# References

1. (Codd, 1990) Codd, E.F. (1990), The Relational Model for Database Management: Version 2 Addison-Wesley
2. (W3S UPDATE, 2019) W3SCHOOLS (2019) UPDATE STATEMENT, AVAILABLE AT: <https://www.w3schools.com/sql/sql_update.asp> (ACCESSED ON 24/07/2019)
3. (W3S DELETE, 2019) W3SCHOOLS (2019) UPDATE STATEMENT, AVAILABLE AT: <https://www.w3schools.com/sql/sql_delete.asp> (ACCESSED ON 24/07/2019)
4. (TUTORIALSPOINT, 2019) TUTORIALSPOINT (2019) DELETE STATEMENT, AVAILABLE AT: <https://www.tutorialspoint.com/sql/sql-delete-query> (ACCESSED ON 24/07/2019)
5. (Conolly, 2015) Thomas M. Connolly (2015) Database systems: a practical approach to design, implementation and management (6th edition), Harlow: Pearson Education
6. (Ullman, 2014) Jeffrey D. Ullman (2014) A first course in database systems (3rd edition) Harlow: Pearson
7. (Tutorials Point, 2018) Tutorials Point (2018), DBMS Overview available at: <http://www.tutorialspoint.com/dbms/dbms_overview.htm> (accessed 04/07/2019)
8. (Techopedia, 2019) Techopedia (2019), Database Management System, available at: <https://www.techopedia.com/definition/24361/database-management-systems-dbms> (Accessed on 22/07/2019)
9. (Oracle, 2019) Oracle (2019) 5 Network Data Model Overflow available at: <https://docs.oracle.com/cd/B19306_01/appdev.102/b14256/sdo_net_concepts.htm> (accessed on 30/07/2019)
10. (Microsoft, 2018) Microsoft (2018), Structure of the Registry, available at: <https://docs.microsoft.com/en-gb/windows/win32/sysinfo/structure-of-the-registry> (Accessed on 22/07/2019)
11. (IBM, 2019) IBM (2019) Information Management System available at: <https://www.ibm.com/ibm/history/ibm100/us/en/icons/ibmims/> (accessed on 22/07/2019)
12. (Burleson, 2017) Donald Burleson (2017) Oracle Database Tips by Donald Burleson available at: <http://www.dba-oracle.com/t_object_top_down_bottom_up.htm> (accessed on 22/07/2019)
13. (MySQL TUTORIAL, 2019) MySQL TUTORIAL (2019) MySQL CHECK Constraint Emulation, available at: <http://www.mysqltutorial.org/mysql-check-constraint/> (accessed on 02/08/2019)
14. (Oracle Corp., 2019) Oracle Corporation (2019) MySQL documentation, available at: <https://dev.mysql.com/doc/refman/5.7/en/create-table.html> (accessed on 02/08/2019)