Name: Shweta Rawat

Employee ID: **902483**

Batch: INTCDB21DW002





­

Table of Contents

[Exercise 1 3](#_Toc46940992)

[Exercise 2 5](#_Toc46940993)

[Exercise 3 8](#_Toc46940994)

[Exercise 4 8](#_Toc46940995)

[Exercise 5 9](#_Toc46940996)

[Exercise 6 9](#_Toc46940997)

[Exercise 7 10](#_Toc46940998)

[Exercise 8 10](#_Toc46940999)

[Exercise 9 10](#_Toc46941000)

[Exercise 10 11](#_Toc46941001)

[Exercise 11 12](#_Toc46941002)

[Exercise 12 13](#_Toc46941003)

[Exercise 13 14](#_Toc46941004)

[Exercise 14 14](#_Toc46941005)

[Exercise 15 14](#_Toc46941006)

[Exercise 16 14](#_Toc46941007)

[Exercise 17 14](#_Toc46941008)

[Exercise 18 15](#_Toc46941009)

[Exercise 19 15](#_Toc46941010)

[Exercise 20 15](#_Toc46941011)

[Exercise 21 16](#_Toc46941012)

[Exercise 22 16](#_Toc46941013)

[Exercise 23 16](#_Toc46941014)

[Exercise 24 16](#_Toc46941015)

[Exercise 25 17](#_Toc46941016)

[Exercise 26 17](#_Toc46941017)

Week1

DDL, DML, DQL, DCL, TCL

|  |  |
| --- | --- |
| Exercise 1 |  |
|  |  |
| Hands-on Exercise Objective | |
| After completing the hands-on exercises, you will be able to:  Understand How to Create Tables | |
|  |  |

Problem Statement:

You have to create all tables whose structure is as mentioned below:

|  |  |  |  |
| --- | --- | --- | --- |
| Trainer\_Info | | | |
| Field | Description | Type | Size |
| Trainer\_Id | Unique ID to every trainer | Character | 20 |
| Salutation | Salutaion of trainer | Character | 7 |
| Trainer\_Name | Name of trainer | Character | 30 |
| Trainer\_Location | Location of trainer | Character | 30 |
| Trainer\_Track | Track of trainer e.g. JAVA, .Net, Mainframe | Character | 15 |
| Trainer\_Qualification | Qualification of trainer e.g. MCA,B. Tech | Character | 100 |
| Trainer\_Experiance | Experience of trainer e.g .2,4 years | Integer | 11 |
| Trainer\_Email | Email Id of trainer | Character | 100 |
| Trainer\_Password | Login password of Trainer | Character | 20 |

|  |  |  |  |
| --- | --- | --- | --- |
| Batch\_Info | | | |
| Field | Description | Type | Size |
| Batch\_Id | Unique ID to every batch | Character | 20 |
| Batch\_Owner | Batch owner name | Character | 30 |
| Batch\_BU\_Name | Business unit to which batch belongs | Character | 30 |

|  |  |  |  |
| --- | --- | --- | --- |
| Module\_Info | | | |
| Field | Description | Type | Size |
| Module\_Id | Unique ID to every module e.g J2SE, J2EE | Character | 20 |
| Module\_Name | Name of module e.g. Core Java SE 1.6 | Character | 40 |
| Module\_Duration | Duration in hrs | Integer | 11 |

|  |  |  |  |
| --- | --- | --- | --- |
| Associate\_Info | | | |
| Field | Description | Type | Size |
| Associate\_Id | Unique ID to every associate | Character | 20 |
| Salutation | Salutaion of associate | Character | 7 |
| Associate\_Name | Name of associate | Character | 30 |
| Associate\_Location | Location of associate | Character | 30 |
| Associate\_Track | Track of associate e.g. JAVA, .Net, Mainframe | Character | 15 |
| Associate\_Qualification | Qualification of associate e.g. MCA,B. Tech | Character | 200 |
| Associate\_Email | Email Id of associate | Character | 100 |
| Associate\_Password | Login password of associate | Character | 20 |

|  |  |  |  |
| --- | --- | --- | --- |
| Questions | | | |
| Field | Description | Type | Size |
| Question\_Id | Unique ID to every question | Character | 20 |
| Module\_Id | Module id from module | Character | 20 |
| Question\_Text | Actual feedback question | Character | 900 |

|  |  |  |  |
| --- | --- | --- | --- |
| Associate\_Status | | | |
| Field | Description | Type | Size |
| Associate\_Id | Associate Id from associate\_Info | Character | 20 |
| Module\_Id | Module Id from module | Character | 20 |
| Start\_Date | Date when associate batch started | Character | 20 |
| End\_Date | Date when associate batch ended | Character | 20 |
| AFeedbackGiven | Flag to test whether Associate has given feedback or not | Character | 20 |
| TFeedbackGiven | Flag to test whether Trainer has given feedback or not | Character | 20 |

|  |  |  |  |
| --- | --- | --- | --- |
| Trainer\_Feedback | | | |
| Field | Description | Type | Size |
| Trainer\_Id | Trainer Id from trainer\_Info | Character | 20 |
| Question\_Id | Question Id from Questions | Character | 20 |
| Batch\_Id | Batch Id from batch\_Info | Character | 20 |
| Module\_Id | Module Id from module | Character | 20 |
| Trainer\_Rating | Rating given by trainer for a perticular question | Integer | 11 |

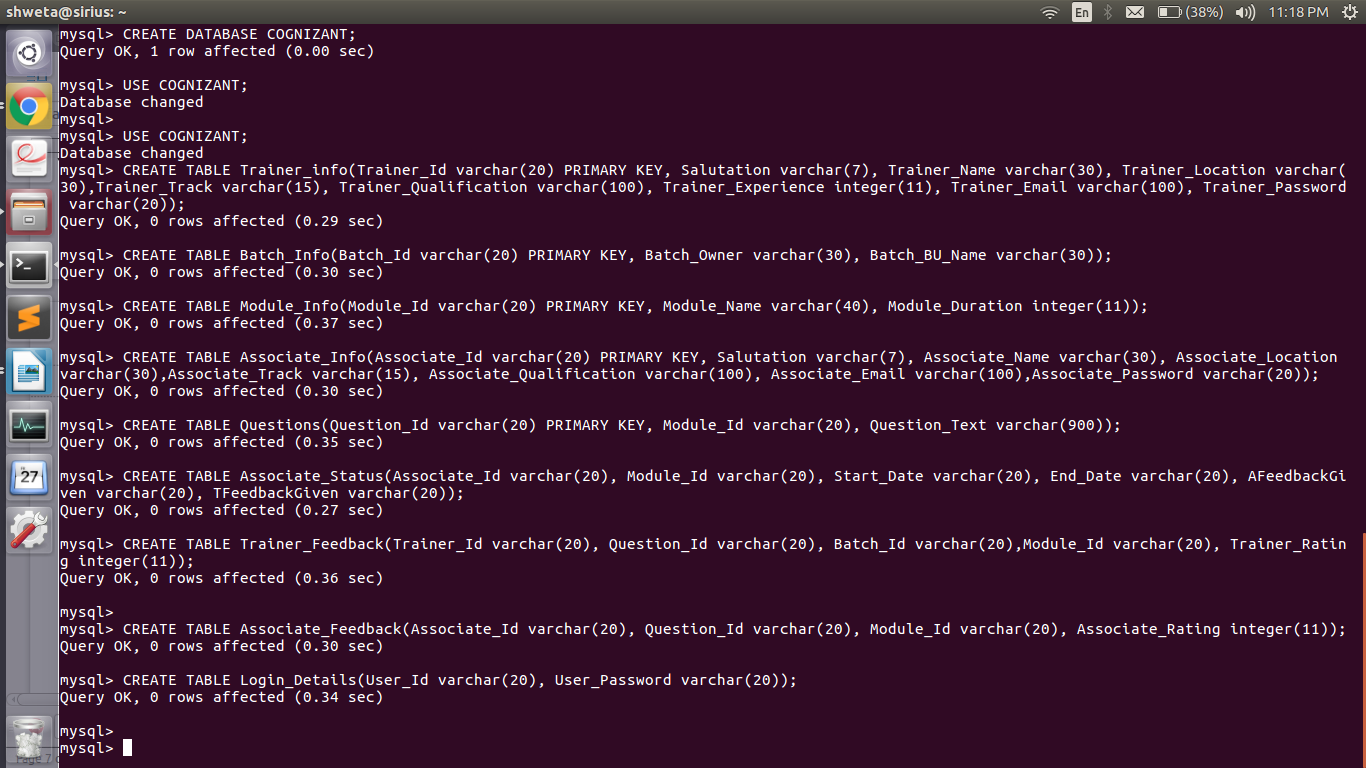
|  |  |  |  |
| --- | --- | --- | --- |
| Associate\_Feedback | | | |
| Field | Description | Type | Size |
| Associate\_Id | Associate Id from associate\_Info | Character | 20 |
| Question\_Id | Question Id from Questions | Character | 20 |
| Module\_Id | Module Id from module | Character | 20 |
| Associate\_Rating | Rating given by associate for a perticular question | Integer | 11 |

|  |  |  |  |
| --- | --- | --- | --- |
| Login\_Details | | | |
| Field | Description | Type | Size |
| User\_Id | User id for login | Character | 20 |
| User\_Password | Password to authenticate user | Character | 20 |

Deliverables Expected:

All Tables creation as per details mentioned above

Output:



Exercise 2

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| Hands-on Exercise Objective | |
| After completing the hands-on exercises, you will be able to:  Insert data into tables | |
|  |  |

Problem Statement:

Insert below details into table:

Trainer\_Info

Trainer\_Id,Salutation,Trainer\_Name,Trainer\_Location,Trainer\_Track,Trainer\_Qualification,Trainer\_Experiance,Trainer\_Email,Trainer\_Password

F001,Mr.,PANKAJ GHOSH,Pune,Java,Bachelor of Technology,12,Pankaj.Ghosh@alliance.com,fac1@123

F002,Mr.,SANJAY RADHAKRISHNAN ,Bangalore,DotNet,Bachelor of Technology,12,Sanjay.Radhakrishnan@alliance.com,fac2@123

F003,Mr.,VIJAY MATHUR,Chennai,Mainframe,Bachelor of Technology,10,Vijay.Mathur@alliance.com,fac3@123

F004,Mrs.,NANDINI NAIR,Kolkata,Java,Master of Computer Applications,9,Nandini.Nair@alliance.com,fac4@123

F005,Miss.,ANITHA PAREKH,Hyderabad,Testing,Master of Computer Applications,6,Anitha.Parekh@alliance.com,fac5@123

F006,Mr.,MANOJ AGRAWAL ,Mumbai,Mainframe,Bachelor of Technology,9,Manoj.Agrawal@alliance.com,fac6@123

F007,Ms.,MEENA KULKARNI,Coimbatore,Testing,Bachelor of Technology,5,Meena.Kulkarni@alliance.com,fac7@123

F009,Mr.,SAGAR MENON ,Mumbai,Java,Master of Science In Information Technology,12,Sagar.Menon@alliance.com,fac8@123

Batch\_Info

Batch\_Id,Batch\_Owner,Batch\_BU\_Name

B001,MRS.SWATI ROY,MSP

B002,MRS.ARURNA K,HEALTHCARE

B003,MR.RAJESH KRISHNAN,LIFE SCIENCES

B004,MR.SACHIN SHETTY,BFS

B005,MR.RAMESH PATEL,COMMUNICATIONS

B006,MRS.SUSAN CHERIAN,RETAIL & HOSPITALITY

B007,MRS.SAMPADA JAIN,MSP

B008,MRS.KAVITA REGE,BPO

B009,MR.RAVI SEJPAL,MSP

Module\_Info

Module\_Id,Module\_Name,Module\_Duration

O10SQL,Oracle 10g SQL ,16

O10PLSQL,Oracle 10g PL/ SQL ,16

J2SE,Core Java SE 1.6,288

J2EE,Advanced Java EE 1.6,80

JAVAFX,JavaFX 2.1,80

DOTNT4,.Net Framework 4.0 ,50

SQL2008,MS SQl Server 2008,120

MSBI08,MS BI Studio 2008,158

SHRPNT,MS Share Point ,80

ANDRD4,Android 4.0,200

EM001,Instructor,0

EM002,Course Material,0

EM003,Learning Effectiveness,0

EM004,Environment,0

EM005,Job Impact,0

TM001,Attendees,0

TM002,Course Material,0

TM003,Environment,0

Associate\_Info

Associate\_Id,Salutation,Associate\_Name,Associate\_Location,Associate\_Track,Associate\_Qualification,Associate\_Email,Associate\_Password

A001,Miss.,GAYATHRI NARAYANAN,Gurgaon,Java,Bachelor of Technology,Gayathri.Narayanan@hp.com,tne1@123

A002,Mrs.,RADHIKA MOHAN,Kerala,Java,Bachelor of Engineering In Information Technology,Radhika.Mohan@cognizant.com,tne2@123

A003,Mr.,KISHORE SRINIVAS,Chennai,Java,Bachelor of Engineering In Computers,Kishore.Srinivas@ibm.com,tne3@123

A004,Mr.,ANAND RANGANATHAN,Mumbai,DotNet,Master of Computer Applications,Anand.Ranganathan@finolex.com,tne4@123

A005,Miss.,LEELA MENON,Kerala,Mainframe,Bachelor of Engineering In Information Technology,Leela.Menon@microsoft.com,tne5@123

A006,Mrs.,ARTI KRISHNAN,Pune,Testing,Master of Computer Applications,Arti.Krishnan@cognizant.com,tne6@123

A007,Mr.,PRABHAKAR SHUNMUGHAM,Mumbai,Java,Bachelor of Technology,Prabhakar.Shunmugham@honda.com,tne7@123

Questions

Question\_Id,Module\_Id,Question\_Text

Q001,EM001,Instructor knowledgeable and able to handle all your queries

Q002,EM001,All the topics in a particular course handled by the trainer without any gaps or slippages

Q003,EM002,The course materials presentation, handson, etc. refered during the training are relevant and useful.

Q004,EM002,The Hands on session adequate enough to grasp the understanding of the topic.

Q005,EM002,The reference materials suggested for each module are adequate.

Q006,EM003,Knowledge and skills presented in this training are applicatible at your work

Q007,EM003,This training increases my proficiency level

Q008,EM004,The physical environment e.g. classroom space, air-conditioning was conducive to learning.

Q009,EM004,The software/hardware environment provided was sufficient for the purpose of the training.

Q010,EM005,This training will improve your job performance.

Q011,EM005,This training align with the business priorities and goals.

Q012,TM001,Participants were receptive and had attitude towards learning.

Q013,TM001,All participats gained the knowledge and the practical skills after this training.

Q014,TM002,The course materials presentation, handson, etc. available for the session covers the entire objectives of the course.

Q015,TM002,Complexity of the course is adequate for the particpate level.

Q016,TM002,Case study and practical demos helpful in understanding of the topic

Q017,TM003,The physical environment e.g. classroom space, air-conditioning was conducive to learning.

Q018,TM003,The software/hardware environment provided was adequate for the purpose of the training.

Associate\_Status

Associate\_Id,Module\_Id,Batch\_Id,Trainer\_Id,Start\_Date,End\_Date

A001,O10SQL,B001,F001,2000-12-15,2000-12-25

A002,O10SQL,B001,F001,2000-12-15,2000-12-25

A003,O10SQL,B001,F001,2000-12-15,2000-12-25

A001,O10PLSQL,B002,F002,2001-2-1,2001-2-12

A002,O10PLSQL,B002,F002,2001-2-1,2001-2-12

A003,O10PLSQL,B002,F002,2001-2-1,2001-2-12

A001,J2SE,B003,F003,2002-8-20,2002-10-25

A002,J2SE,B003,F003,2002-8-20,2002-10-25

A001,J2EE,B004,F004,2005-12-1,2005-12-25

A002,J2EE,B004,F004,2005-12-1,2005-12-25

A003,J2EE,B004,F004,2005-12-1,2005-12-25

A004,J2EE,B004,F004,2005-12-1,2005-12-25

A005,JAVAFX,B005,F006,2005-12-4,2005-12-20

A006,JAVAFX,B005,F006,2005-12-4,2005-12-20

A006,SQL2008,B006,F007,2007-6-21,2007-6-28

A007,SQL2008,B006,F007,2007-6-21,2007-6-28

A002,MSBI08,B007,F006,2009-6-26,2009-6-29

A003,MSBI08,B007,F006,2009-6-26,2009-6-29

A004,MSBI08,B007,F006,2009-6-26,2009-6-29

A002,ANDRD4,B008,F005,2010-6-5,2010-6-28

A005,ANDRD4,B008,F005,2010-6-5,2010-6-28

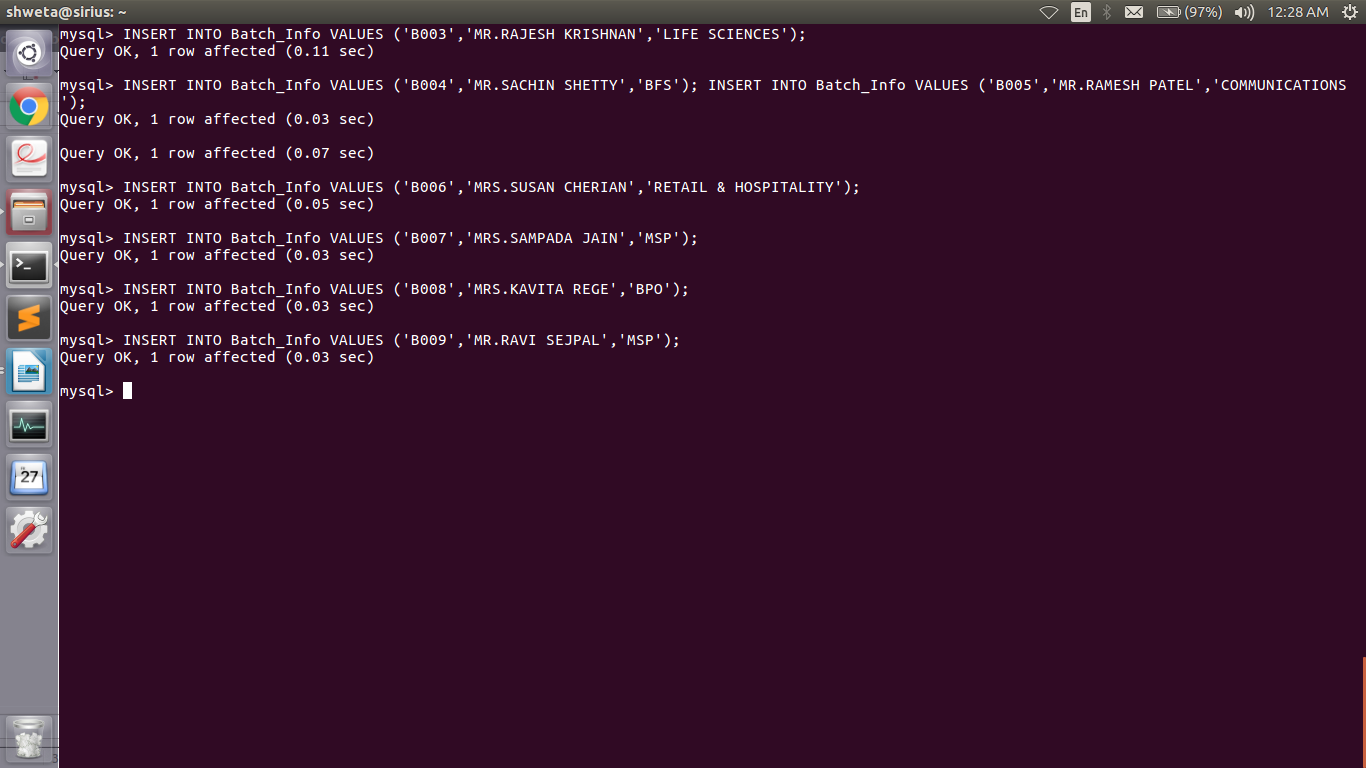
A003,ANDRD4,B009,F005,2011-8-1,2011-8-20

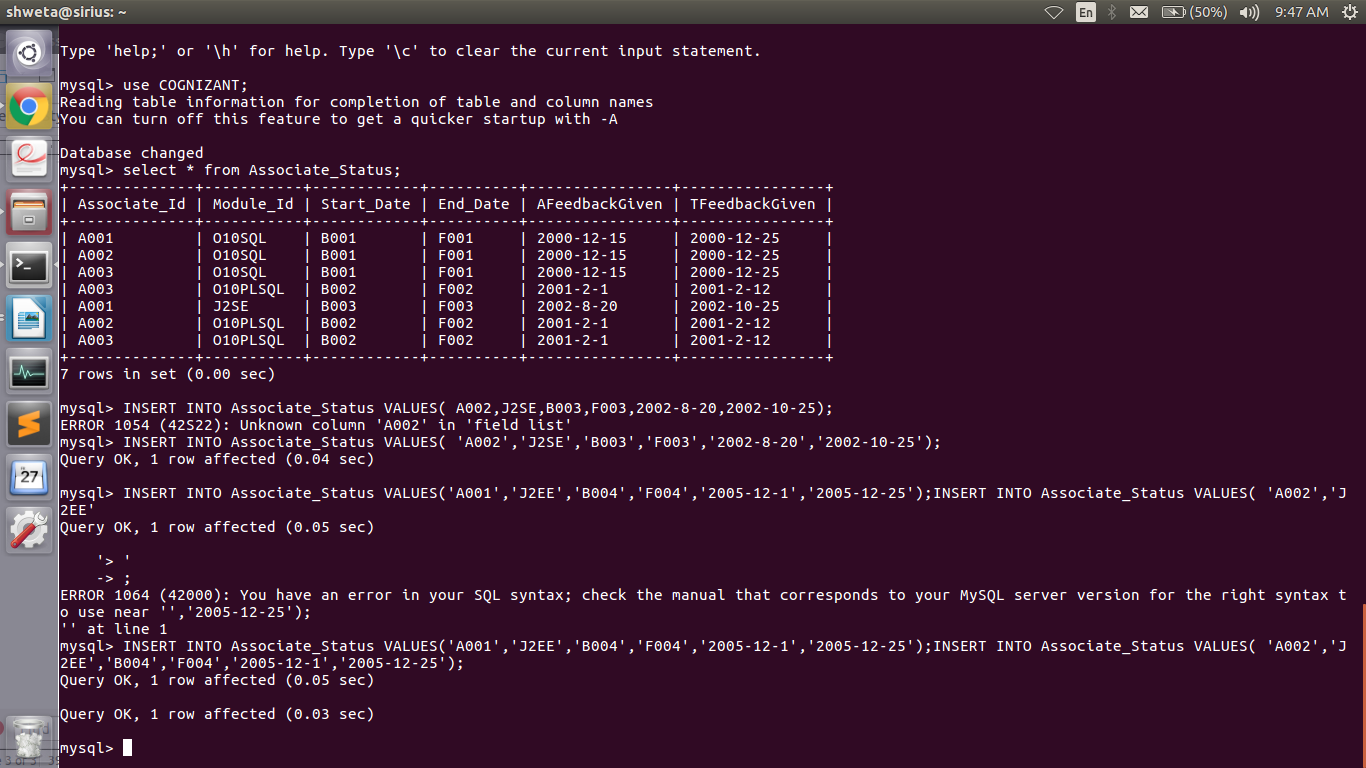
A006,ANDRD4,B009,F005,2011-8-1,2011-8-20

Deliverables Expected:

All records inserted in tables

Output: Values inserted successfully





Exercise 3

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | |
|  | Hands-on Exercise Objective | | |
|  | After completing the hands-on exercises, you will be able to:  Understand how to change data in table | | |
|  |  | |  |

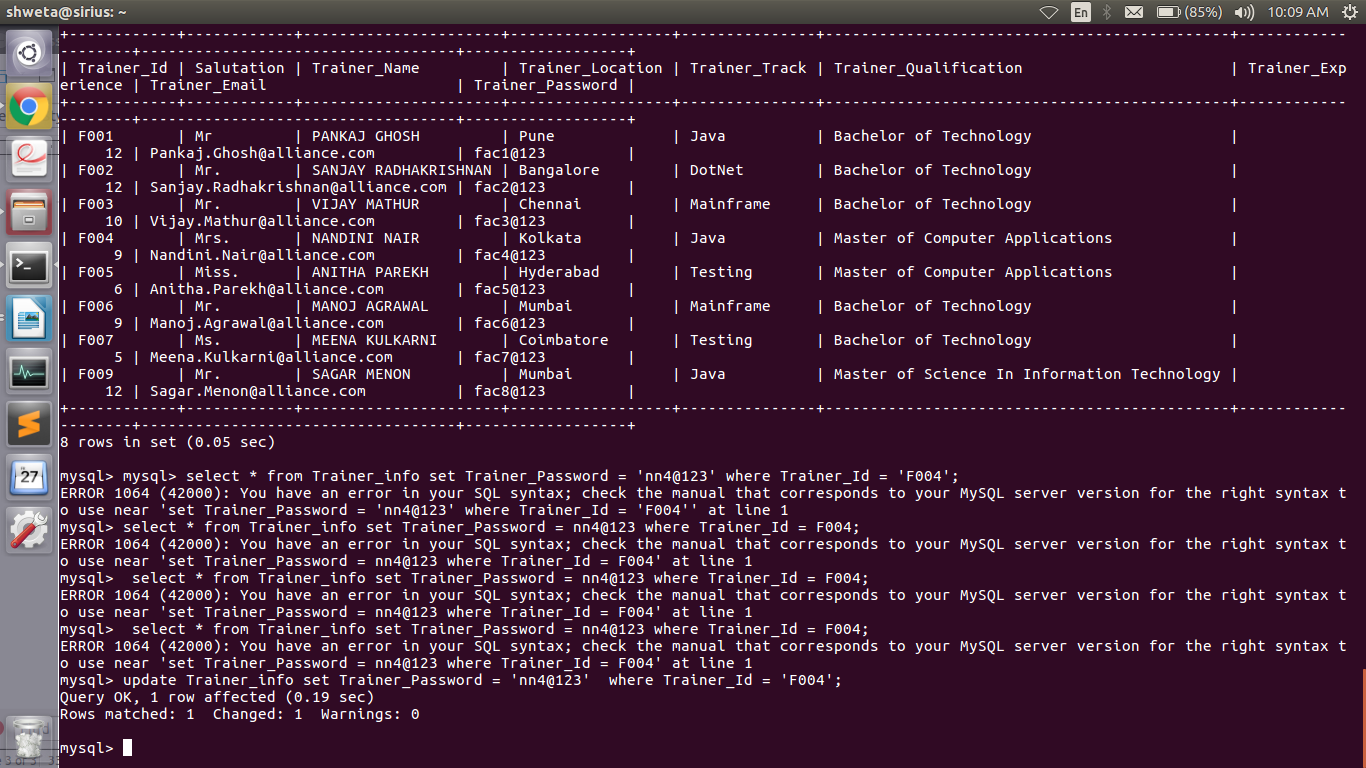
Problem Statement:

Change the password of trainer F004 from fac4@123 to nn4@123

Deliverables Expected:

Updated password of trainer F004.

**Output:**



Exercise 4

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| Hands-on Exercise Objective | |
| After completing the hands-on exercises, you will be able to:  Understand how to delete data from table. | |
|  |  |

Problem Statement:

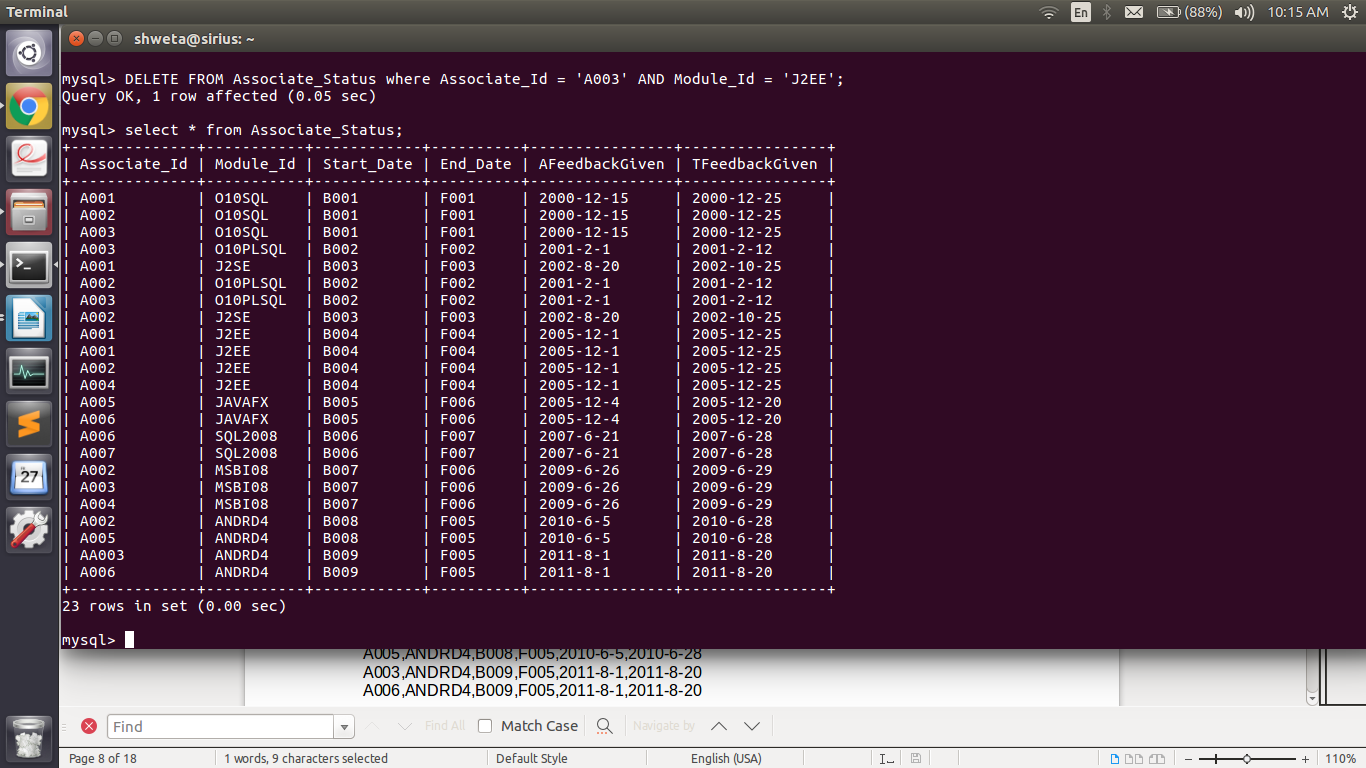
Remove following record form associate\_status table

A003,J2EE,B004,F004,2005-12-1,2005-12-25

Deliverables Expected:

Removed record from associate status.

**Output:**



Exercise 5

|  |  |
| --- | --- |
| Hands-on Exercise Objective | |
| After completing the hands-on exercises, you will be able to:  Understand how to perform TOP N records. | |
|  |  |

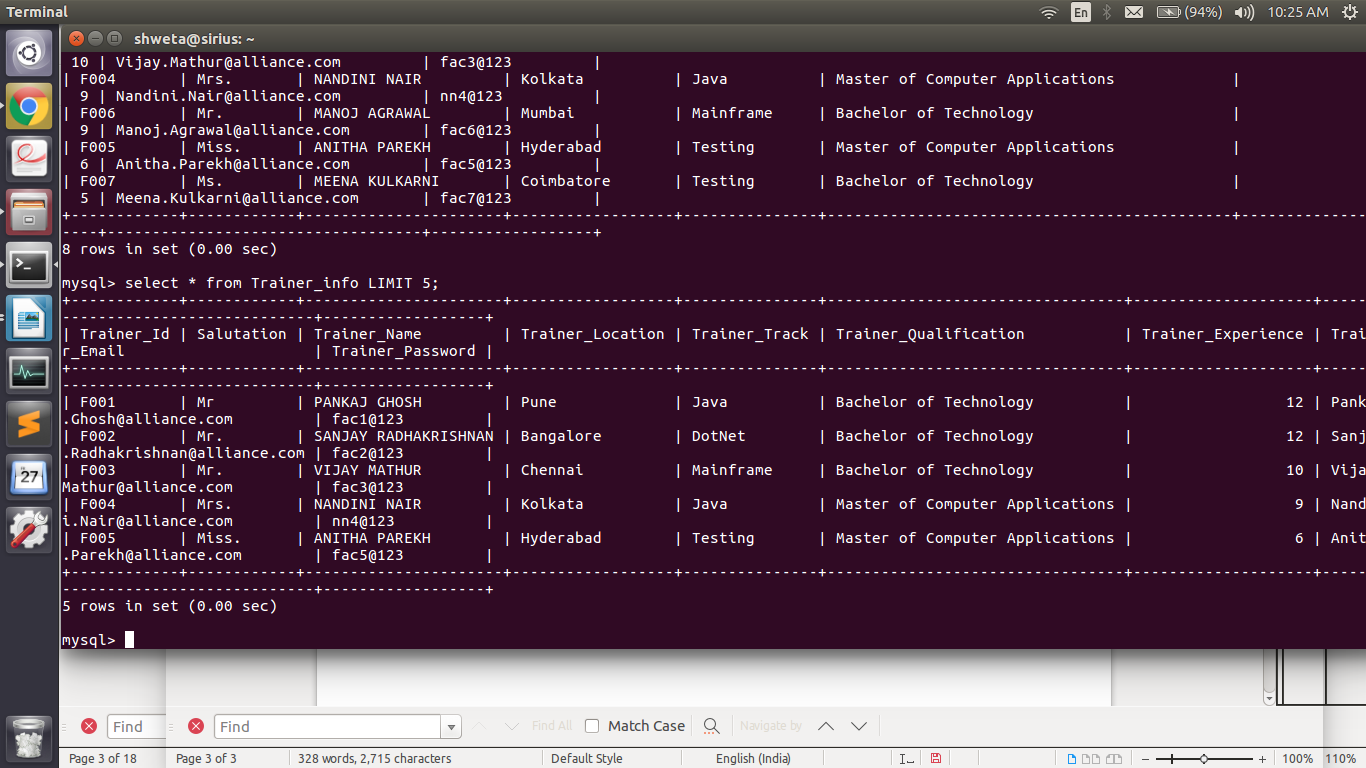
Problem Statement:

Fetch first five trainers who have maximum years of experience & display there details.

Deliverables Expected:

Top five experience trainers.

Output:



Exercise 6

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| Hands-on Exercise Objective | |
| After completing the hands-on exercises, you will be able to:  Understand transaction. | |
|  |  |

Problem Statement:

Begin transaction & insert below to records into Login\_Details

'U001' Admin1@123

'U002' Admin2@123

Perform rollback operation & verify whether any records are inserted in table or not.

Deliverables Expected:

Empty table

Output:



Exercise 7

|  |
| --- |
|  |

|  |  |
| --- | --- |
| Hands-on Exercise Objective | |
| After completing the hands-on exercises, you will be able to:  Understand TCL. | |
|  |  |

Problem Statement:

Create a dummy user in database. Grant create & select table privilege to him/her.

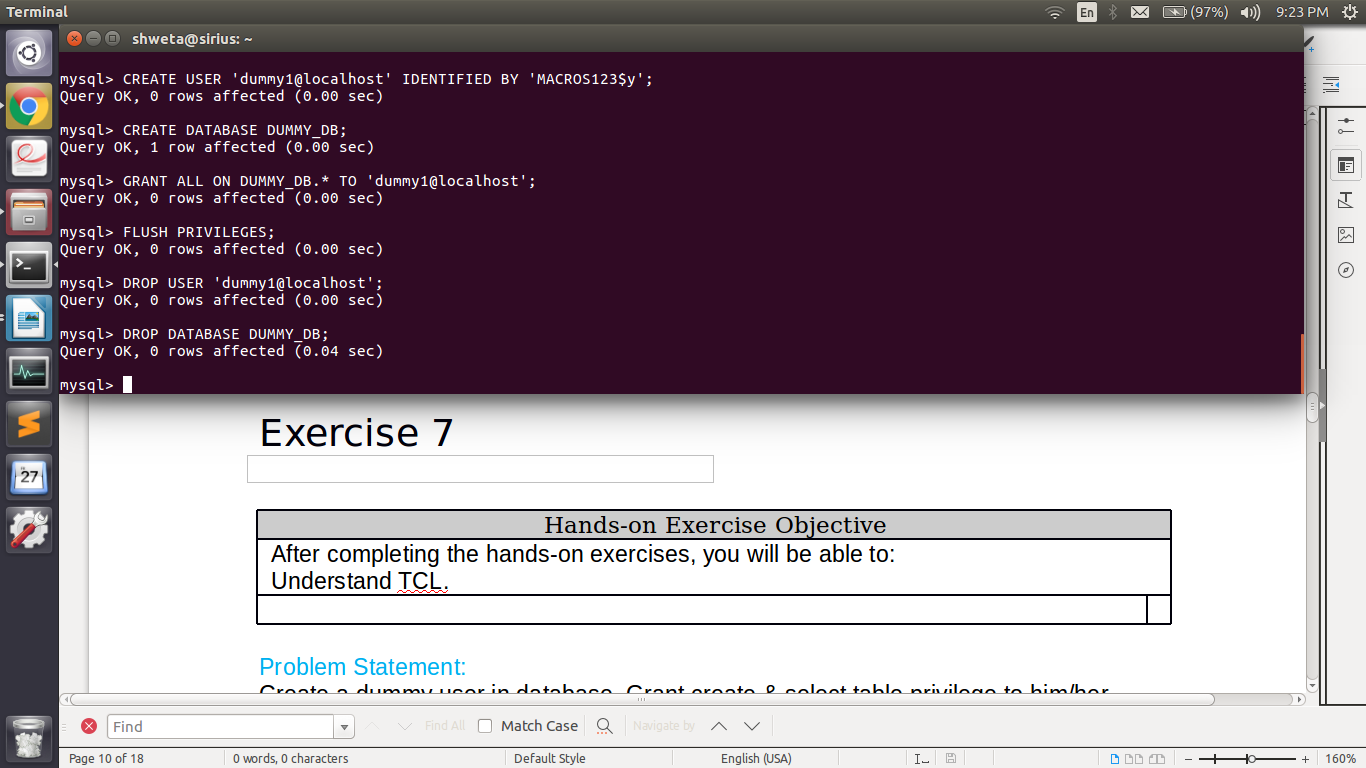
Repeat the above all queries using login credentials of newly created user.

Revoke the privilege assigned to this newly created user.

Deliverables Expected:

Same result as achieved till now.

Output:



Exercise 8

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| Hands-on Exercise Objective | |
| After completing the hands-on exercises, you will be able to:  Understanding how to drop table. | |
|  |  |

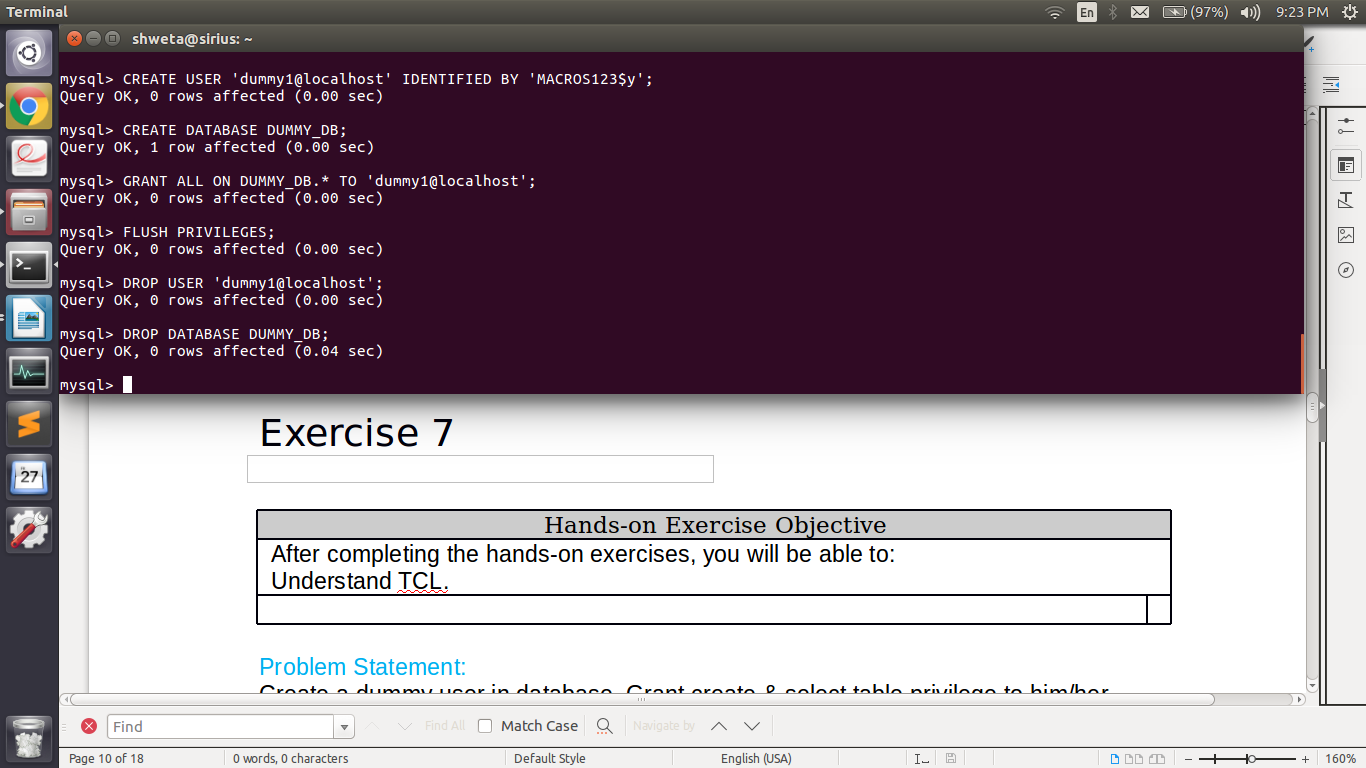
Problem Statement:

Remove table Login\_Details from database.

Deliverables Expected:

No table found message while querying Login\_Details table.

Output:



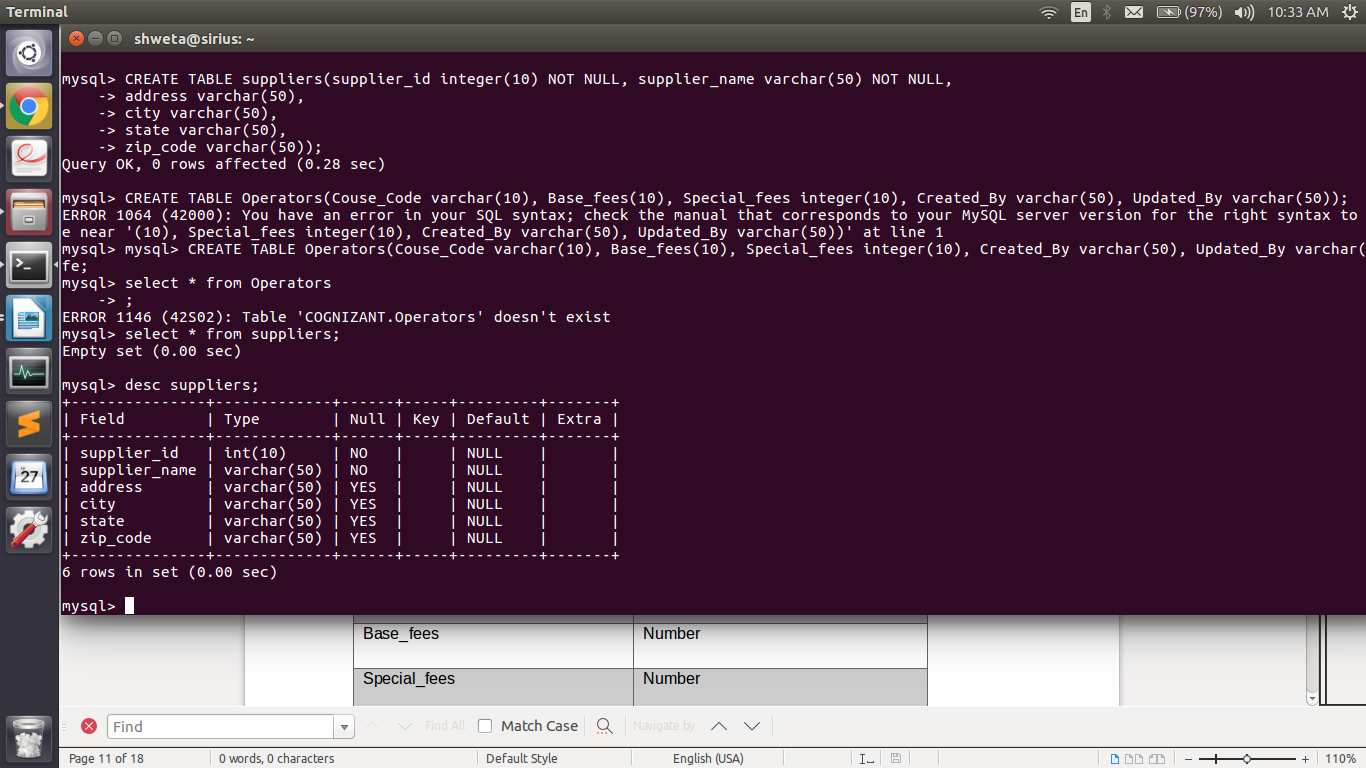
Constraints

Exercise 9

Create a table called suppliers that stores supplier ID, name, and address information.

|  |  |  |
| --- | --- | --- |
| **Field** | **Data type** | **Constraint** |
| supplier\_id | number(10) | Not Null |
| supplier\_name | varchar2(50) | Not Null |
| address | varchar2(50) |  |
| city | varchar2(50), |  |
| state | varchar2(25) |  |
| zip\_code | varchar2(10)) |  |

**Output:**



Operators

Exercise 10

The prerequisite for the given activity is to create the following tables.

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| Course\_Code | Varchar2 |
| Base\_fees | Number |
| Special\_fees | Number |
| Created\_By | Varchar2 |
| Updated\_By | Varchar2 |

Course Fees

|  |  |  |  |
| --- | --- | --- | --- |
| **COURSE\_CODE** | **BASE\_FEES** | **SPECIAL\_FEES** | **DISCOUNT** |
| 1 | 180 | 100 | 10 |
| 2 | 150 | 110 | 10 |
| 3 | 160 | 170 | 5 |
| 4 | 150 | 100 | 10 |
| 6 | 190 | 100 | 40 |

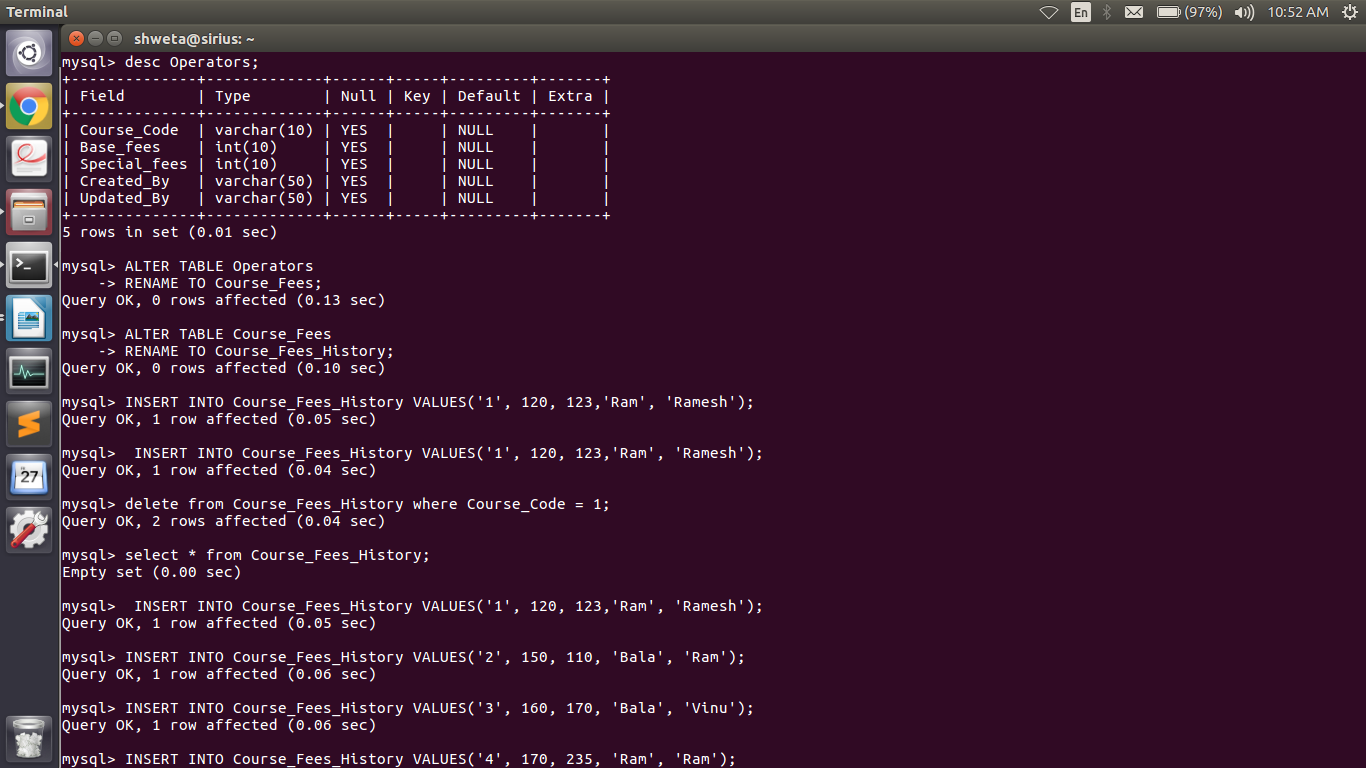
Course Fees\_History

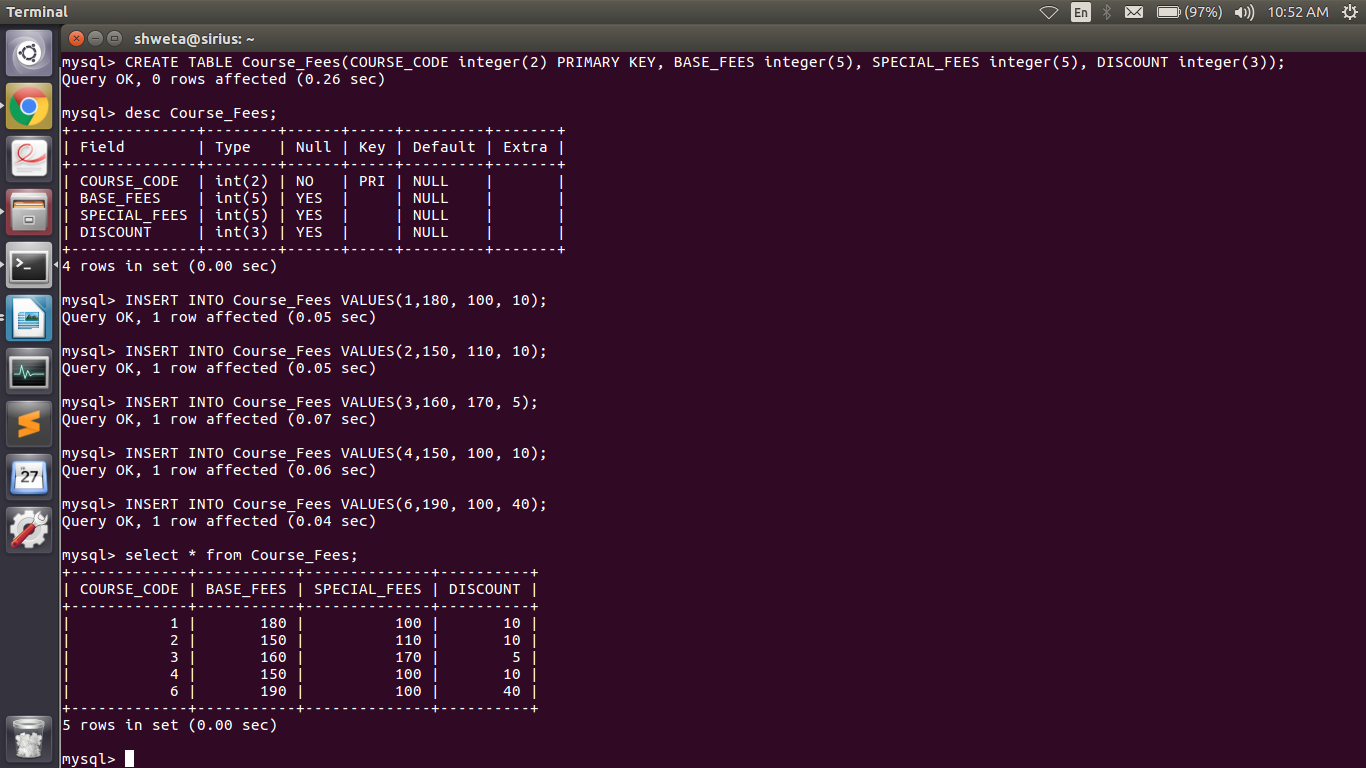
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COURSE\_CODE** | **BASE\_FEES** | **SPECIAL\_FEES** | **CREATED\_BY** | **Updated \_By** |
| 1 | 120 | 123 | Ram | Ramesh |
| 2 | 150 | 110 | Bala | Ram |
| 3 | 160 | 170 | Bala | Vinu |
| 4 | 170 | 235 | Ram | Ram |
| 6 | 190 | 100 | Vinod | Vinod |

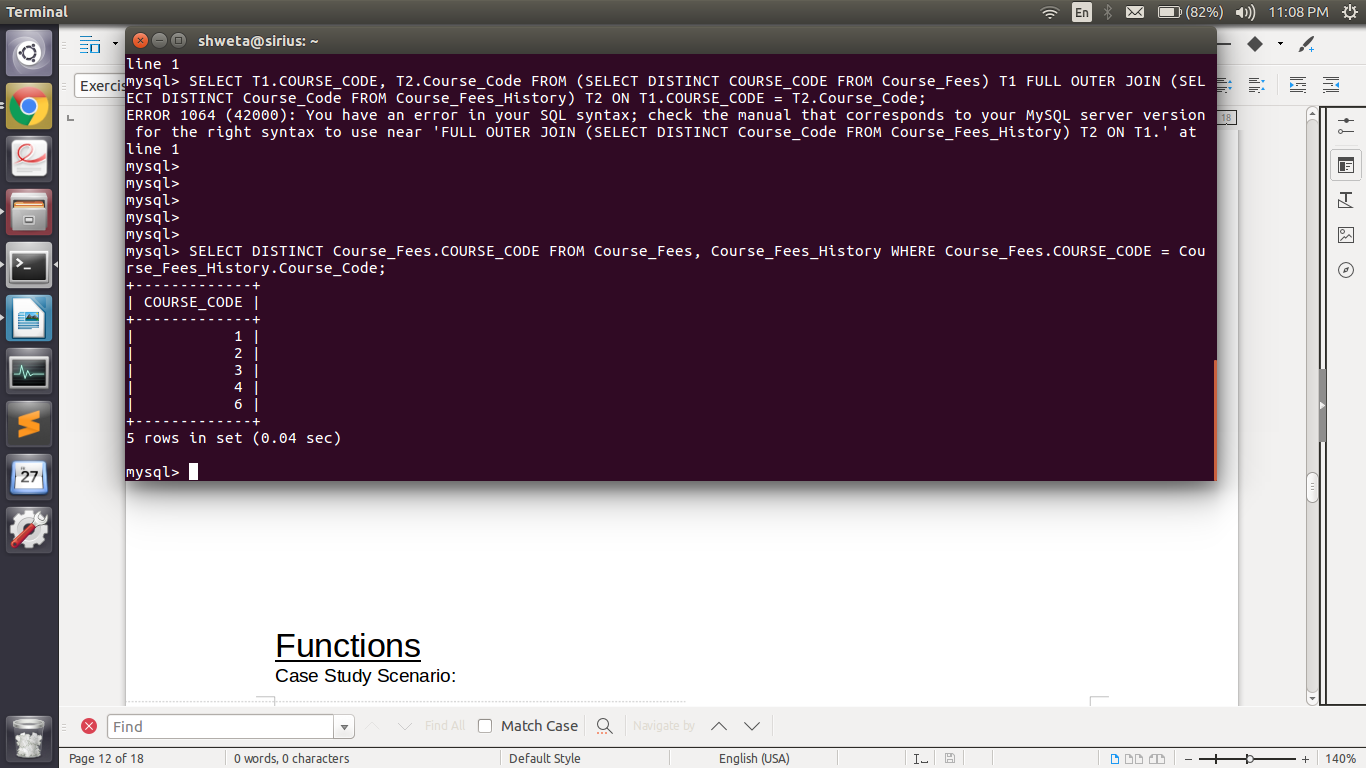
Problem Statement:

Display all the unique courses between course fees and course fees\_history.

Output:







Exercise 11

Use the following columns to check for uniqueness of Course\_Code, BASE\_FEES and SPECIAL\_FEES of the courses in both the COURSE\_FEES and COURSE\_FEES\_HISTORY.

Output:



Functions

Case Study Scenario:

This case study is to develop a Course Management System (CMS) for ABC University. The following are the two use cases for which the database needs to be designed.

Add Course:

To add the course details into the course management system.

Retrieve Course:

Retrieve the courses stored in the system and display it.

The courses to be added will have the following attributes: Course Code, Course Name, Number of participants, Course Description, Course Duration, Course start date, and Course Type.

Pre-requisite

Create a table named “COURSE\_INFO” & “Student\_Info” with following column name, data type, data size, and following constraints:

COURSE\_CODE – PRIMARY KEY

COURSE\_NAME – NOT NULL.

STUDENT\_ID –PRIMARY KEY

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Data Size** |
| COURSE\_CODE | varchar | 10 |
| COURSE\_NAME | varchar | 20 |
| COURSE\_DESCRIPTION | varchar | 25 |
| COURSE\_START\_DATE | Date |  |
| COURSE\_DURATION | int |  |
| NO\_OF\_PARTICIPANTS | int |  |
| COURSE\_TYPE | Char(3) |  |

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Data Size** |
| STUDENT\_ID | varchar | 10 |
| FIRST\_NAME | varchar | 20 |
| LAST\_NAME | varchar | 25 |
| ADDRESS | varchar | 150 |

Exercise 12

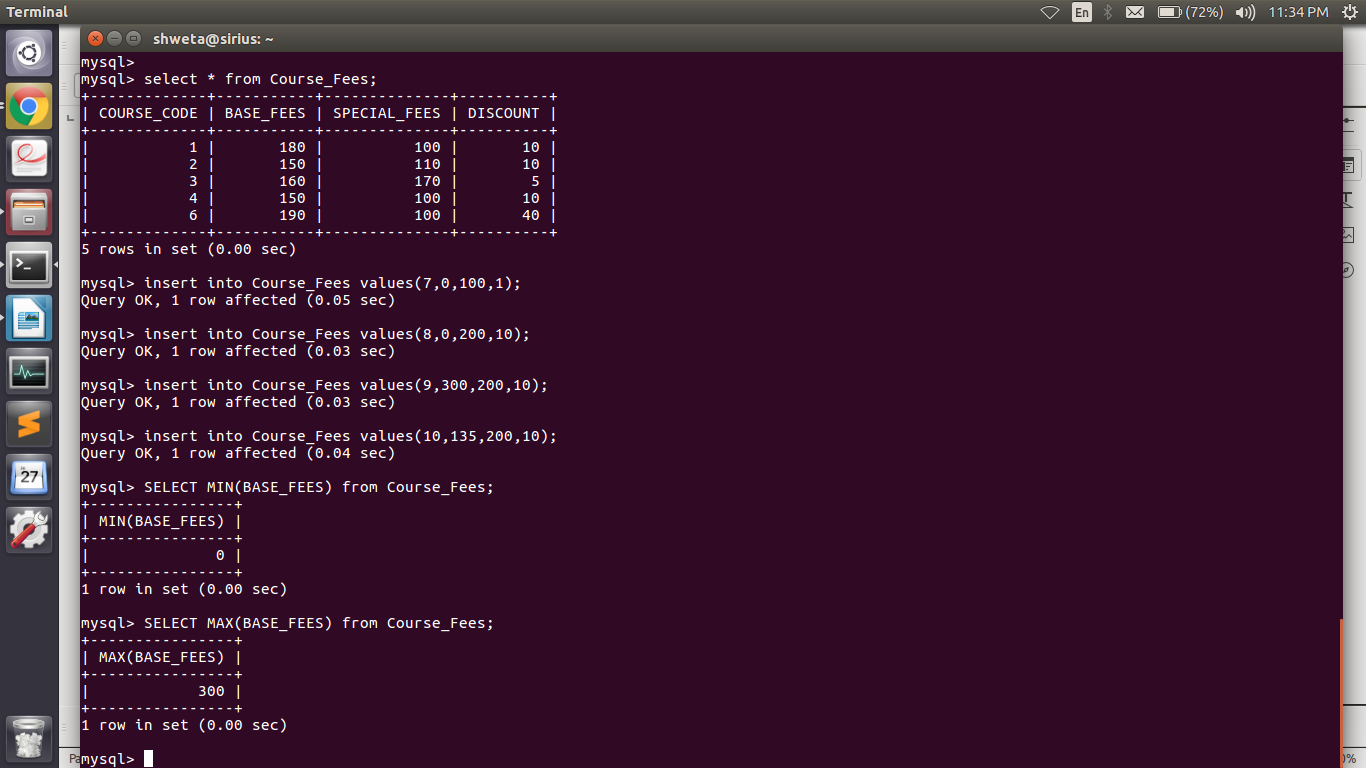
Pre-requisite : Use the Course\_Info and Course\_Fees table.

* + Insert 2 records in course\_fees table with base fees as null.
  + Insert 2 records in course\_fees table with base fees as 300 and 175.

Problem Statement:

Display the minimum and maximum base fees of the courses.

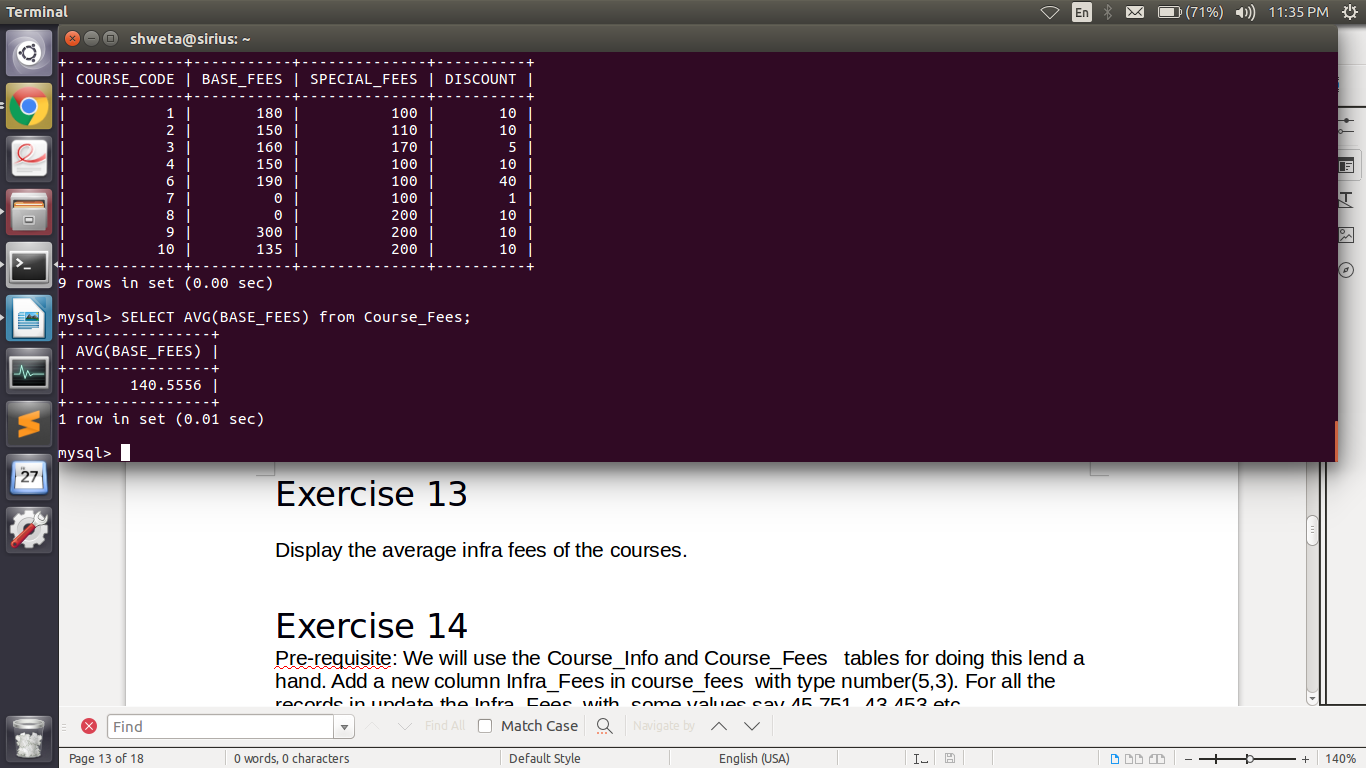
Output:



Exercise 13

Display the average infra fees of the courses.

**Output:**



Exercise 14

Pre-requisite: We will use the Course\_Info and Course\_Fees tables for doing this lend a hand. Add a new column Infra\_Fees in course\_fees with type number(5,3). For all the records in update the Infra\_Fees with some values say 45.751, 43.453 etc.

Hints: Use joins wherever needed

Problem Statement:

Develop a query which will display the course name and the number of days between the current date and course start date in Course\_Info table

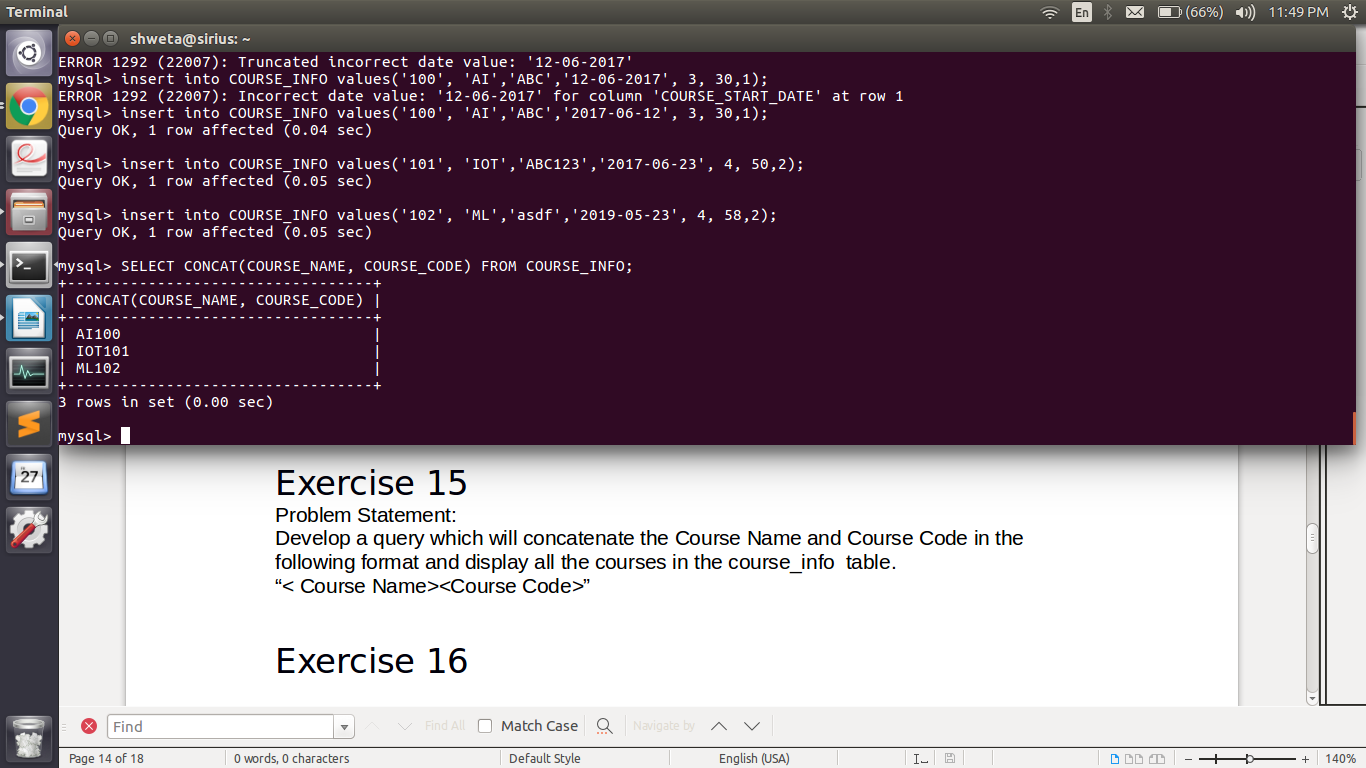
Exercise 15

Problem Statement:

Develop a query which will concatenate the Course Name and Course Code in the following format and display all the courses in the course\_info table.

“< Course Name><Course Code>”

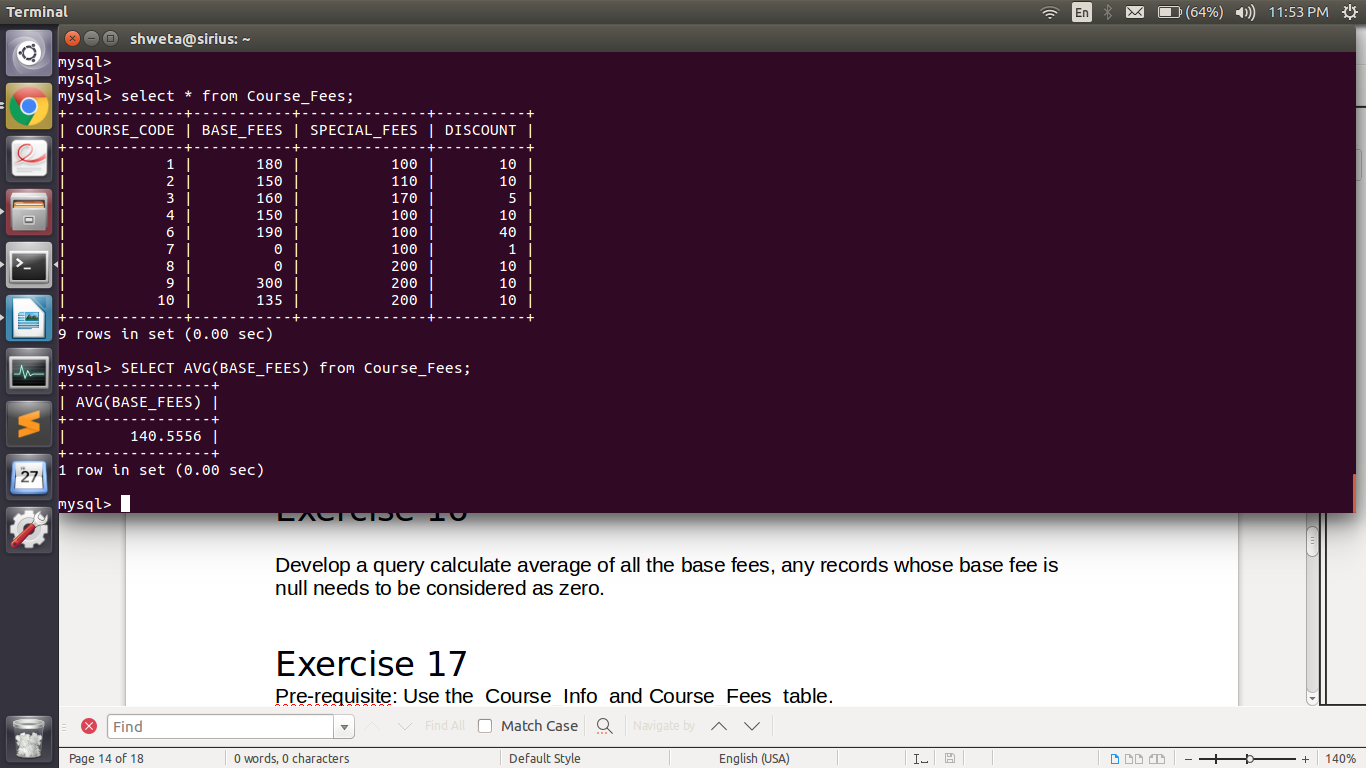
Output:



Exercise 16

Develop a query calculate average of all the base fees, any records whose base fee is null needs to be considered as zero.

Output:



Exercise 17

Pre-requisite: Use the Course\_Info and Course\_Fees table.

* + Insert 2 records in course\_fees table with base fees as null.
  + Insert 2 records in course\_fees table with base fees as 300 and 175.
  + Insert 3 records in course\_info table each course with course type CLR,EL, OF

Problem Statement:

Write a query which will display the course type and the appropriate message as mentioned below.

|  |  |
| --- | --- |
| **Course\_Type** | **Message** |
| CLR | ‘Class Room’ |
| EL | ‘ELearning’ |
| OF | ‘Offline Reading’ |

Clauses

• Case Study Scenario:

– This case study is to develop a Course Management System (CMS) for ABC University. The following are the two use cases for which the database needs to be designed.

• Add Course:

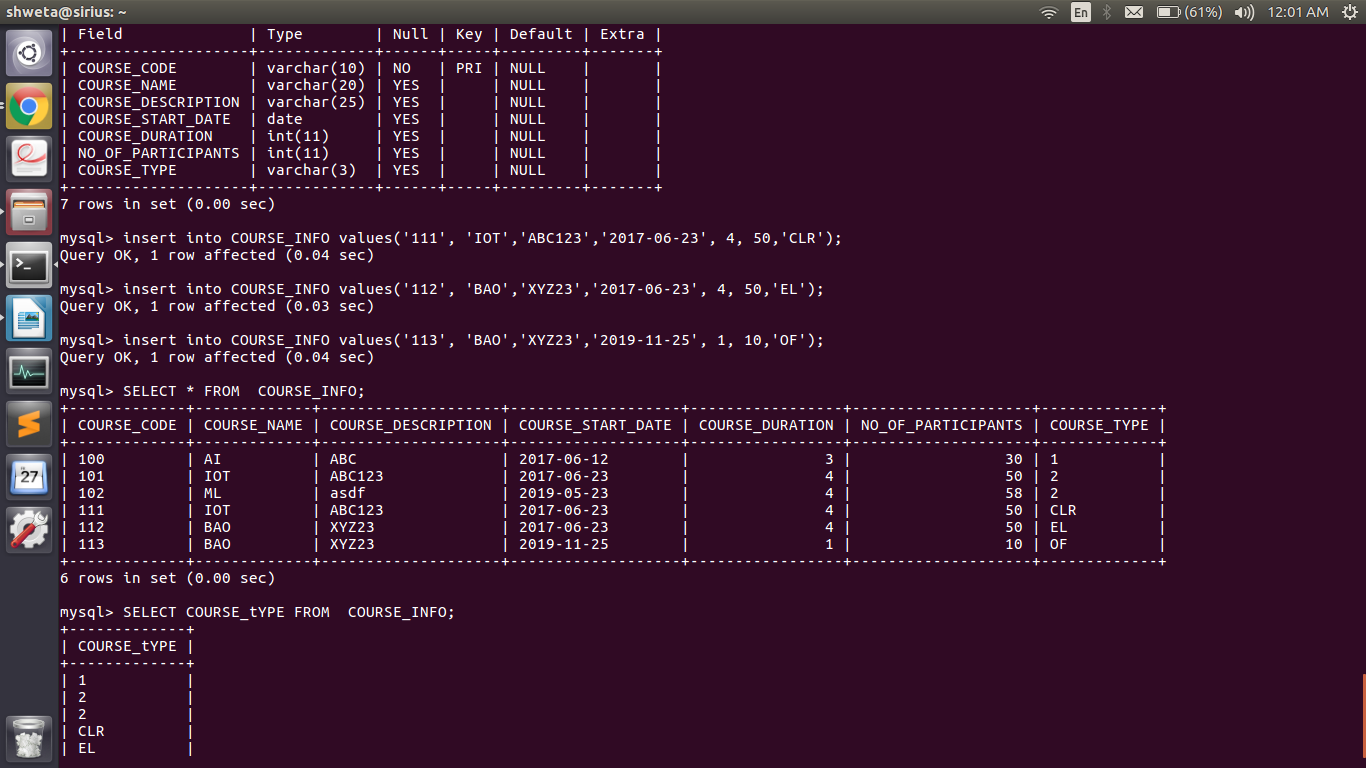
– To add the course details into the course management system.

• Retrieve Course:

– Retrieve the courses stored in the system and display it.

• The courses to be added will have the following attributes: Course Code, Course Name, Number of participants, Course Description, Course Duration, Course start date, and Course Type.

Output:



Exercise 18

Problem Statement: Develop a query which would retrieve the total number of students enrolled for courses on a specific date grouped by course start date and display course start date and total number of students.

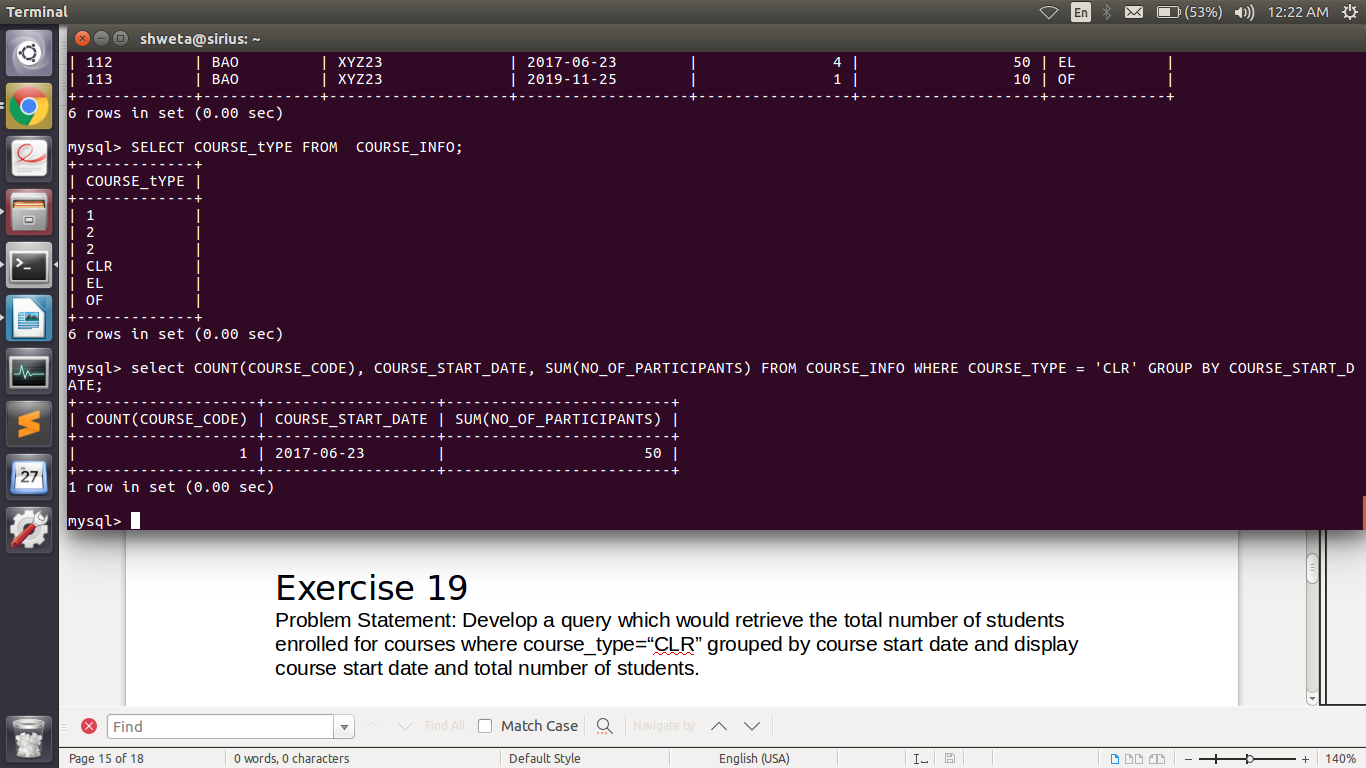
Output:



Exercise 19

Problem Statement: Develop a query which would retrieve the total number of students enrolled for courses where course\_type=“CLR” grouped by course start date and display course start date and total number of students.

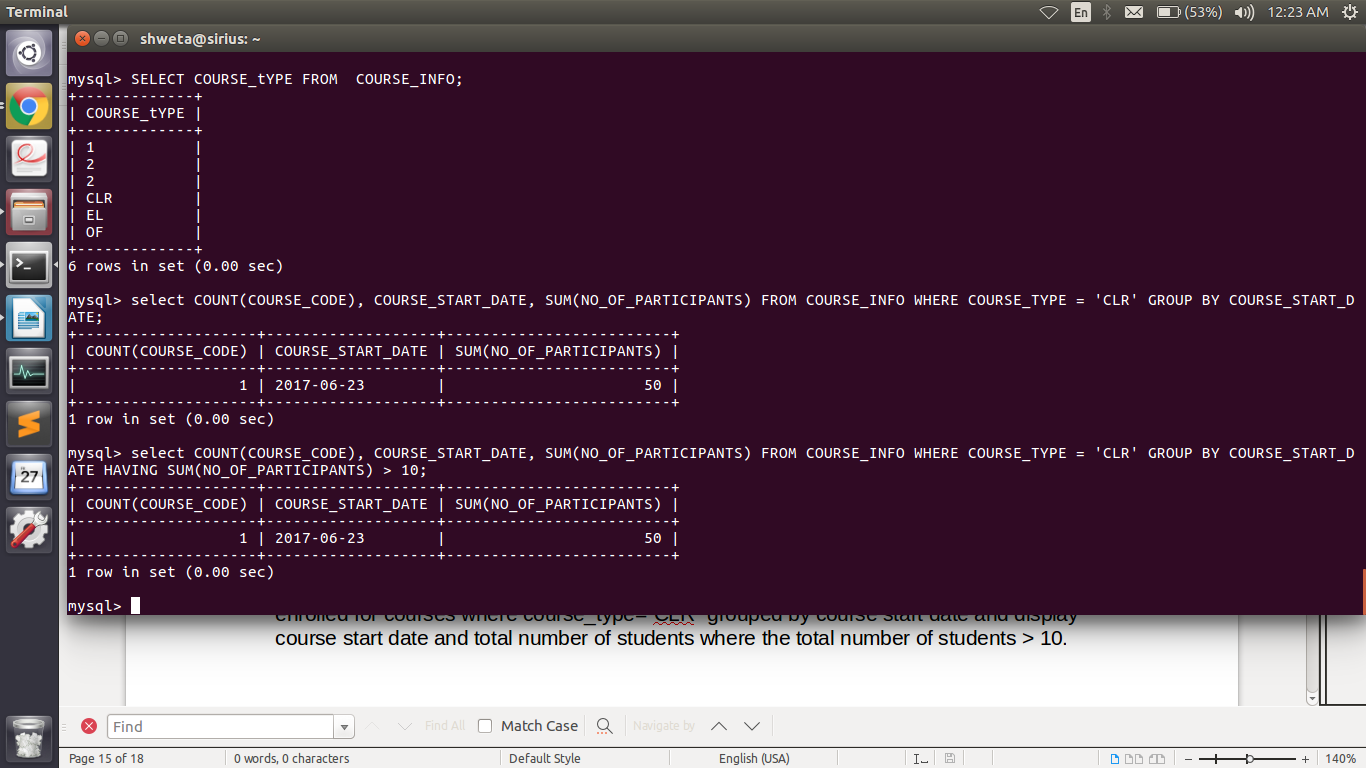
Output:



Exercise 20

Problem Statement: Develop a query which would retrieve the total number of students enrolled for courses where course\_type=“CLR” grouped by course start date and display course start date and total number of students where the total number of students > 10.

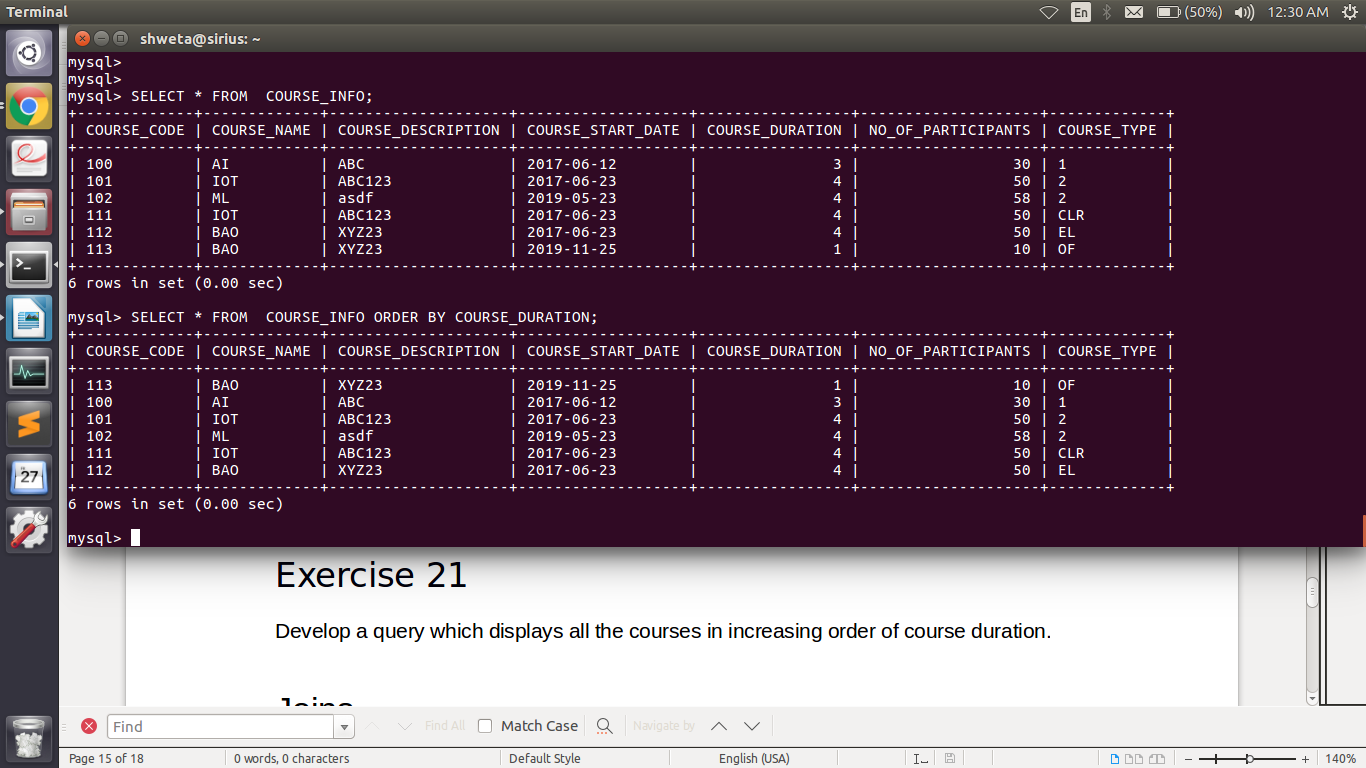
Output:



Exercise 21

Develop a query which displays all the courses in increasing order of course duration.

Output:



Joins

Case Study Scenario:

This case study is to develop a Course Management System (CMS) for ABC University. The following are the two use cases for which the database needs to be designed.

Add Course:

To add the course details into the course management system.

Retrieve Course:

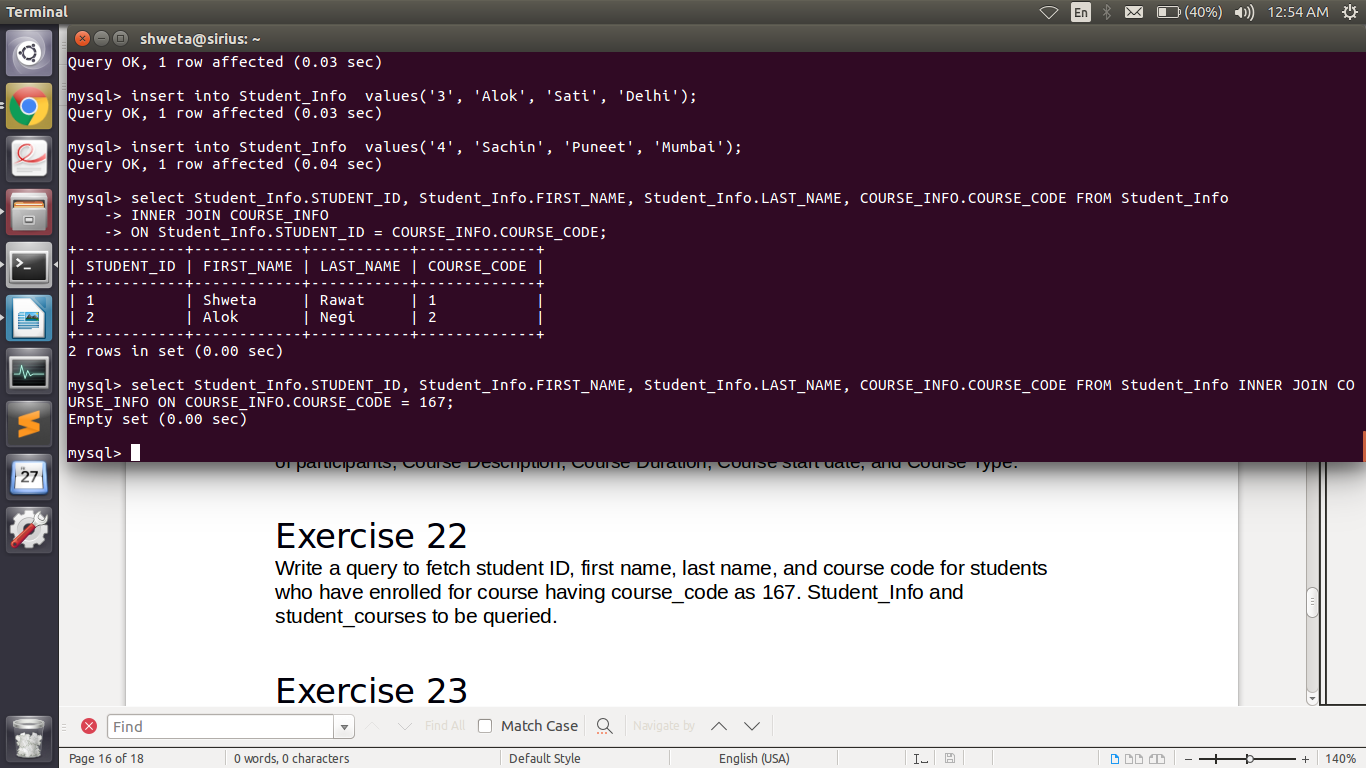
Retrieve the courses stored in the system and display it.

The courses to be added will have the following attributes: Course Code, Course Name, Number of participants, Course Description, Course Duration, Course start date, and Course Type.

Exercise 22

Write a query to fetch student ID, first name, last name, and course code for students who have enrolled for course having course\_code as 167. Student\_Info and student\_courses to be queried.

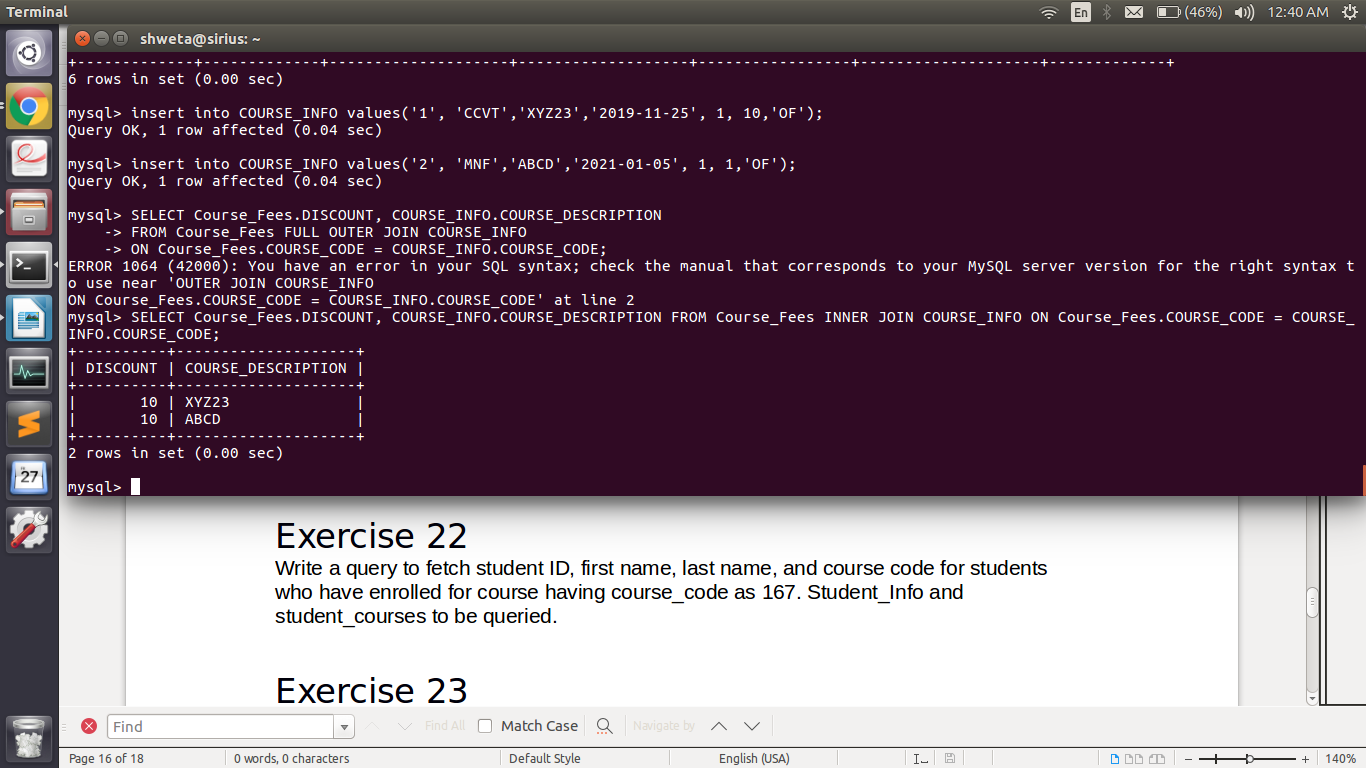
Output:



Exercise 23

Write a query to display the discount offered on the courses along with course descriptions.

Output:



Exercise 24

Pre-requisite: Create a student record (student\_info\_ <employee id>).

Note: Do not create any other detail regarding this student in any other table.

Problem Statement: Write a query to fetch first names of the students along with the course codes of the courses they have enrolled in.

Note: Even if the course\_code does not exist for a student, the record needs to be fetched.

Get the same output as per the requirement above using:

* + Left Join
  + Right Join

Output:

Subqueries

Case Study Scenario:

This case study is to develop a Course Management System (CMS) for ABC University. The following are the two use cases for which the database needs to be designed.

Add Course:

To add the course details into the course management system.

Retrieve Course:

Retrieve the courses stored in the system and display it.

The courses to be added will have the following attributes: Course Code, Course Name, Number of participants, Course Description, Course Duration, Course start date, and Course Type.

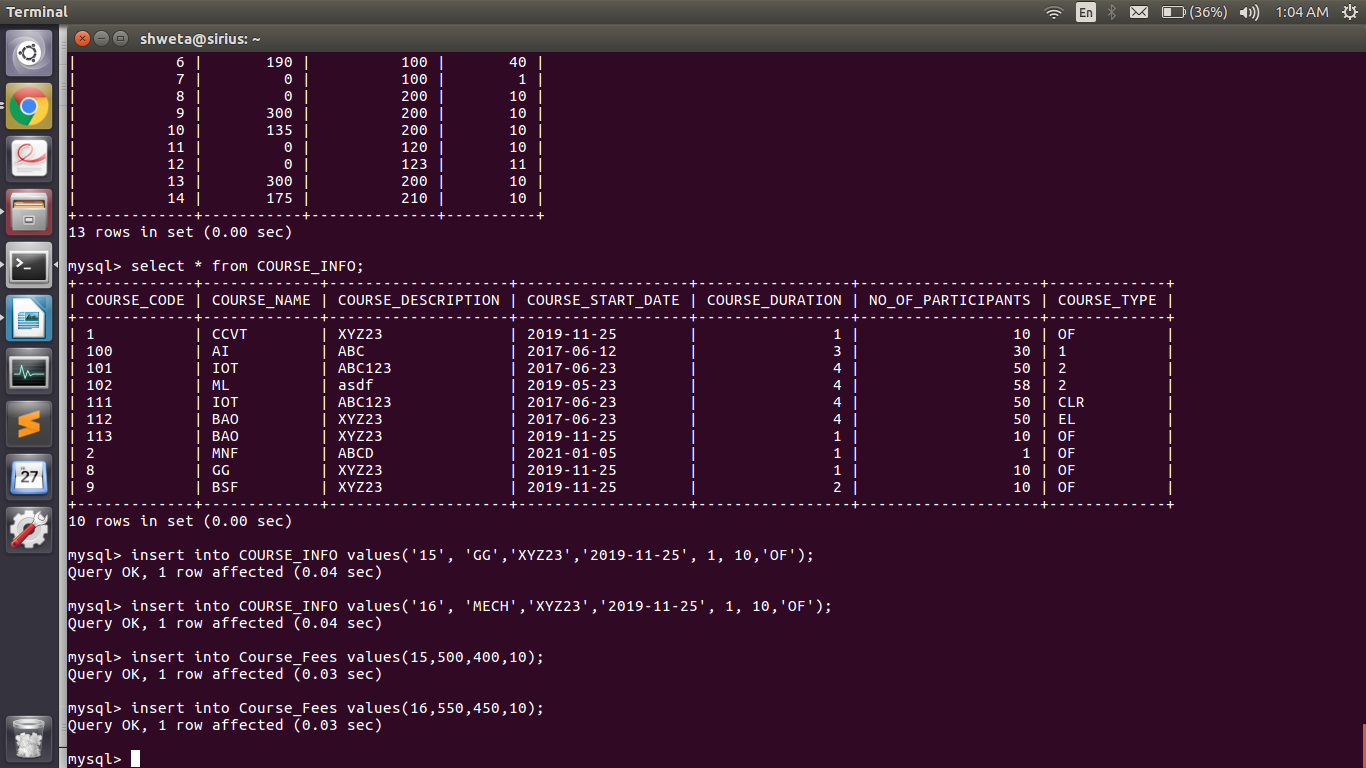
Exercise 25

Pre-Requisite:

* + Insert the following records
  + Add two new courses in course\_info table
  + Add the course fees for the two courses in course\_Fees with fees amount < 1500
  + Enroll two students to the newly added courses

Problem Statement: Write a query which fetches the student id for students who have enrolled for at least one course whose fees is less than 1500.

Output:



Exercise 26

Write a query which fetches the student id and student name for students who have enrolled for at least one course whose fees is less than 1500.