Departmental Database

Part B: Relational Database Implementation

[Name]

[Student Id]

**Document Name**: DatabaseTemplatePartB.docx

Document History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Version | Revision | Description | Author | Role |
| 31/08/2022 | 1 | 1 | First version document |  | System Analyst |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Table of Contents**

[1. Revised Scope and Boundaries 3](#_Toc110947633)

[2. Revised Business Rules 3](#_Toc110947634)

[3. Final Entity Relationship Diagram 3](#_Toc110947635)

[4. SQL to Create Database and Tables 3](#_Toc110947636)

[5. SQL Queries 0](#_Toc110947637)

[6. Views 0](#_Toc110947638)

[6.1. Creating View 0](#_Toc110947639)

[6.2. Running view 0](#_Toc110947640)

[6.3. Dropping View 0](#_Toc110947641)

[7. Stored Procedures 0](#_Toc110947642)

[7.1. Retrieving Data Using a Stored Procedure 0](#_Toc110947643)

[7.2. Inserting Data Using a Stored Procedure 0](#_Toc110947644)

[7.3. Modifying Data Using a Stored Procedure 0](#_Toc110947645)

[7.4. Stored Procedure Using a Parameter 0](#_Toc110947646)

[7.5. Dropping a Stored Procedure 0](#_Toc110947647)

[8. Triggers 0](#_Toc110947648)

[8.1. Trigger 1 0](#_Toc110947649)

[8.2. Trigger 2 0](#_Toc110947650)

[9. Appendix 1](#_Toc110947651)

# Revised Scope and Boundaries

# Revised Business Rules

# Final Entity Relationship Diagram

# SQL to Create Database and Tables

**SELECT \* FROM mydb\_UPDATE.CLASS;**

INSERT INTO mydb\_UPDATE.CLASS(CLASS\_ID, TEACHER\_ID, SUBJECT\_ID, START\_TIME, STUDENT\_ID, NUMBER\_OF\_STUDENTS, CLASS\_NAME)

VALUES (1, 1, 2,'Wed 9am', 3, 2, 'COMPUTER SCIENCE');

INSERT INTO mydb\_UPDATE.CLASS(CLASS\_ID, TEACHER\_ID, SUBJECT\_ID, START\_TIME, STUDENT\_ID, NUMBER\_OF\_STUDENTS, CLASS\_NAME)

VALUES (2, 3, 2,'Tues 11am', 2, 3, 'PYTHON LANGUAGE');

INSERT INTO mydb\_UPDATE.CLASS(CLASS\_ID, TEACHER\_ID, SUBJECT\_ID, START\_TIME, STUDENT\_ID, NUMBER\_OF\_STUDENTS, CLASS\_NAME)

VALUES (3, 2, 1,'Fri 9am', 1, 3, 'COMMERCIAL LAW');

**SELECT \* FROM mydb\_UPDATE.DEPARTMENT;**

INSERT INTO mydb\_UPDATE.DEPARTMENT (DEPARTMENT\_NAME, DEPARTMENT\_ID)

VALUES ('BUSINESS DEPARTMENT', 1);

INSERT INTO mydb\_UPDATE.DEPARTMENT (DEPARTMENT\_NAME, DEPARTMENT\_ID)

VALUES ('COMPUTER SCIENCE DEPARTMENT', 2);

INSERT INTO mydb\_UPDATE.DEPARTMENT (DEPARTMENT\_NAME, DEPARTMENT\_ID)

VALUES ('COMMERCE DEPARTMENT',3);

**SELECT \* FROM mydb\_UPDATE.COURSE;**

INSERT INTO mydb\_UPDATE.COURSE (COURSE\_ID, DEPARTMENT\_ID, COURSE\_NAME)

VALUES (1, 2,'ICT40220') ;

INSERT INTO mydb\_UPDATE.COURSE (COURSE\_ID,DEPARTMENT\_ID, COURSE\_NAME)

VALUES(2, 2,'ICT30220');

INSERT INTO mydb\_UPDATE.COURSE (COURSE\_ID,DEPARTMENT\_ID, COURSE\_NAME)

VALUES (3, 2, 'ICT50220');

**SELECT \* FROM mydb\_UPDATE.PERSON;**

INSERT INTO mydb\_UPDATE.PERSON(PERSON\_ID,FIRST\_NAME, LAST\_NAME, EMAIL, MOBILE, AGE, DOB)

VALUES(1,'ALICE', 'ROBERSTON', 'ROBERTSON.ALICE@HOTMAIL.COM' , 470894685, 34, '1988-09-02');

INSERT INTO mydb\_UPDATE.PERSON(PERSON\_ID,FIRST\_NAME, LAST\_NAME, EMAIL, MOBILE, AGE, DOB)

VALUES(2, 'BOB', 'JOHNSON', 'JOHNSON.BOB@HOTMAIL.COM', 403598764, 26, '1996-03-15');

INSERT INTO mydb\_UPDATE.PERSON(PERSON\_ID,FIRST\_NAME, LAST\_NAME, EMAIL, MOBILE, AGE, DOB)

VALUES(3,'CHRIS', 'WONG', 'WONG.CHRIS@HOTMAIL.COM', 428976874, 18, '2004-08-12' );

INSERT INTO mydb\_UPDATE.PERSON(PERSON\_ID,FIRST\_NAME, LAST\_NAME, EMAIL, MOBILE,AGE, DOB)

VALUES(4,'DAVE', 'CHARMCHY', 'CHARMCHY.DAVE@HOTMAIL.COM', 412937952, 45, '1977-03-17' );

INSERT INTO mydb\_UPDATE.PERSON(PERSON\_ID,FIRST\_NAME, LAST\_NAME, EMAIL, MOBILE, AGE, DOB)

VALUES(5,'EVE', 'MORRISON', 'EVE.MORRISON@HOTMAIL.COM', 412937952,23, '1999-05-16' );

INSERT INTO mydb\_UPDATE.PERSON(PERSON\_ID,FIRST\_NAME, LAST\_NAME, EMAIL, MOBILE, AGE, DOB)

VALUES(6,'COCO', 'WILLAM', 'COCO.WILLAM@HOTMAIL.COM', 412937952, 34, '1988-06-07');

**SELECT \* FROM mydb\_UPDATE.TEACHERS;**

INSERT INTO mydb\_UPDATE.TEACHERS(FIRST\_NAME, LAST\_NAME, TEACHER\_EMAIL, TEACHER\_ID, PAY\_CODE)

VALUES('DAVE', 'CHARMCHY', 'CHARMCHY.DAVE@HOTMAIL.COM',1,'3.1' );

INSERT INTO mydb\_UPDATE.TEACHERS(FIRST\_NAME, LAST\_NAME, TEACHER\_EMAIL, TEACHER\_ID, PAY\_CODE)

VALUES('EVE', 'MORRISON', 'EVE.MORRISON@HOTMAIL.COM', 2,'3.2');

INSERT INTO mydb\_UPDATE.TEACHERS(FIRST\_NAME, LAST\_NAME, TEACHER\_EMAIL, TEACHER\_ID, PAY\_CODE)

VALUES('COCO', 'WILLAM', 'COCO.WILLAM@HOTMAIL.COM', 3,'3.3');

**SELECT \* FROM mydb\_UPDATE.SUBJECTS;**

INSERT INTO mydb\_UPDATE.SUBJECTS (SUBJECT\_ID, SUBJECT\_NAME, COURSE\_ID, AVERAGE\_SCORE)

VALUES (1,'DIGITAL IMAGES', 1, 78);

INSERT INTO mydb\_UPDATE.SUBJECTS (SUBJECT\_ID, SUBJECT\_NAME, COURSE\_ID, AVERAGE\_SCORE)

VALUES (2,'WEB DEV', 2, 80);

INSERT INTO mydb\_UPDATE.SUBJECTS (SUBJECT\_ID, SUBJECT\_NAME, COURSE\_ID, AVERAGE\_SCORE)

VALUES (3,'DATABASE', 2, 82);

# SQL Queries

|  |  |  |
| --- | --- | --- |
| **Purpose** | **Tables Involved** | **Command(s) Involved and**  **Result** |
| *A:,List all students in alphabetical order* | *Students* | *SELECT \* FROM mydb\_UPDATE.STUDENTS;*  *SELECT STUDENT\_ID, FIRST\_NAME, LAST\_NAME*  *FROM STUDENTS*  *ORDER BY FIRST\_NAME ASC;* |
| B:Retrieve student age records | Students, person | *LECT AGE, S. FIRST\_NAME, S. LAST\_NAME, S. STUDENT\_EMAIL,S.STUDENT\_NUMBER*  *FROM mydb\_UPDATE.STUDENTS S*  *JOIN mydb\_UPDATE.PERSON P*  *ON S.FIRST\_NAME = P.FIRST\_NAME AND S.LAST\_NAME = P.LAST\_NAME;* |
| C: the number of persons | person | SELECT COUNT(PERSON\_ID)  FROM mydb\_UPDATE.PERSON; |
| D:student\_id, who select which couse | Class, subject | SELECT C.CLASS\_ID, S.AVERAGE\_SCORE  FROM mydb\_UPDATE.CLASS C, mydb\_UPDATE.SUBJECTS S  WHERE C.SUBJECT\_ID = S.SUBJECT\_ID  AND S.AVERAGE\_SCORE > 79; |
| E:the class whose subject average score is higher than 20 | course and subject | SELECT \* FROM COURSE  WHERE COURSE\_ID IN  (SELECT COURSE\_ID FROM SUBJECTS where AVERAGE\_SCORE>80) |
| F:square root (average\_score )\*10 | Subjects | SELECT (SQRT(AVERAGE\_SCORE)\* 10) FROM mydb\_UPDATE.SUBJECTS; |
| G:DOB is greater than 1990-0101 |  | SELECT \* FROM mydb\_UPDATE.PERSON  WHERE DOB > '1990-01-01'  ORDER BY DOB DESC |

# Views

## Creating View, Running view, Dropping View

## 

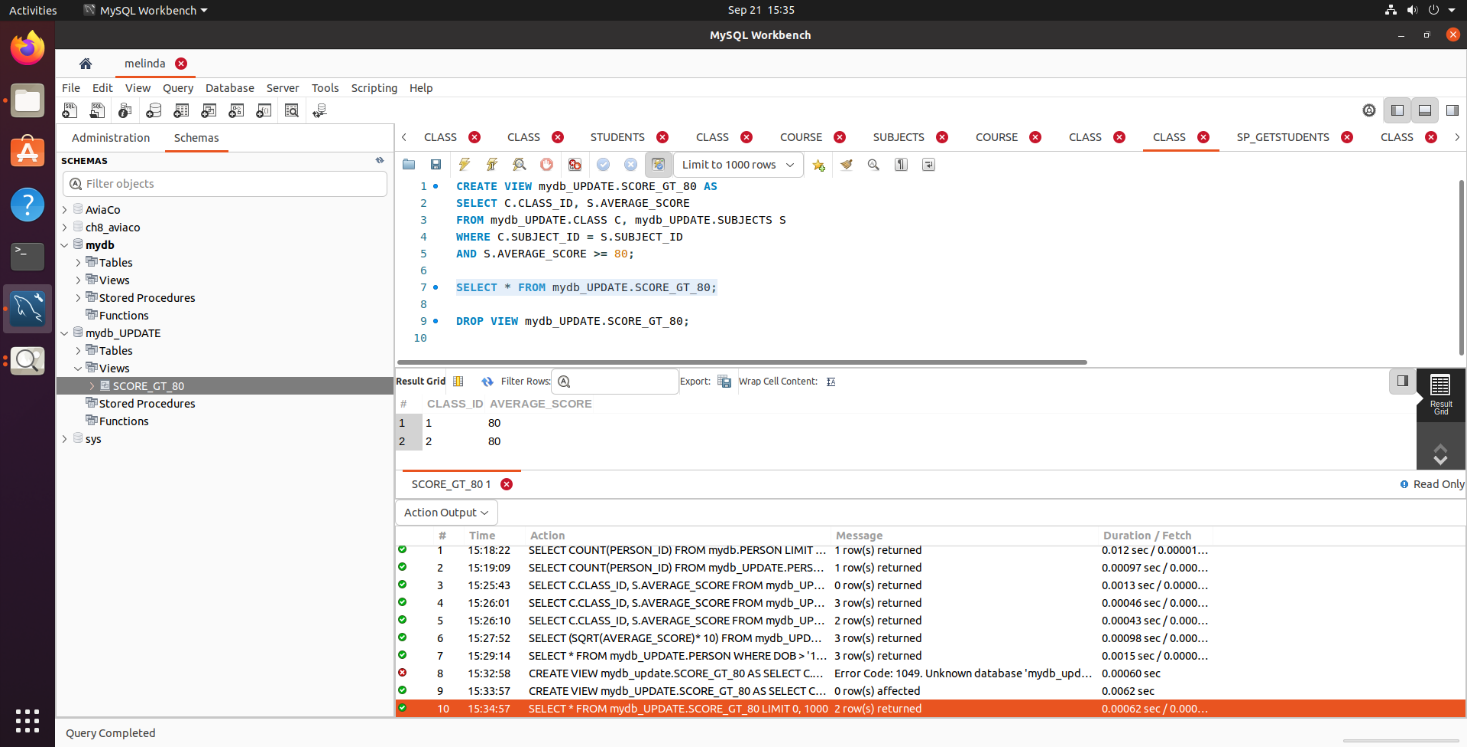
CREATE VIEW mydb.SCORE\_GT\_82 AS

SELECT C.CLASS\_ID, S.AVERAGE\_SCORE

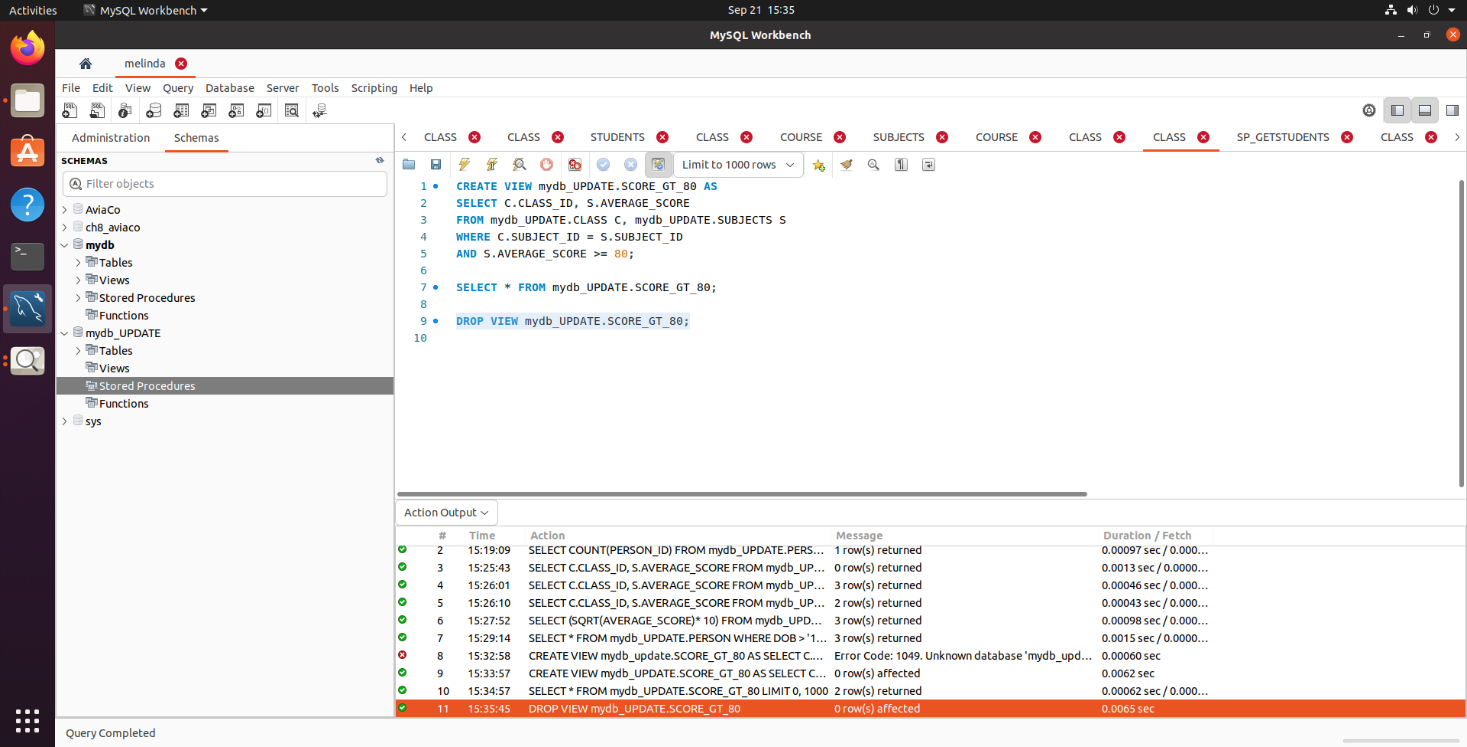
FROM mydb.CLASS C, mydb.SUBJECTS S

WHERE C.SUBJECT\_ID = S.SUBJECT\_ID

AND S.AVERAGE\_SCORE >= 82;

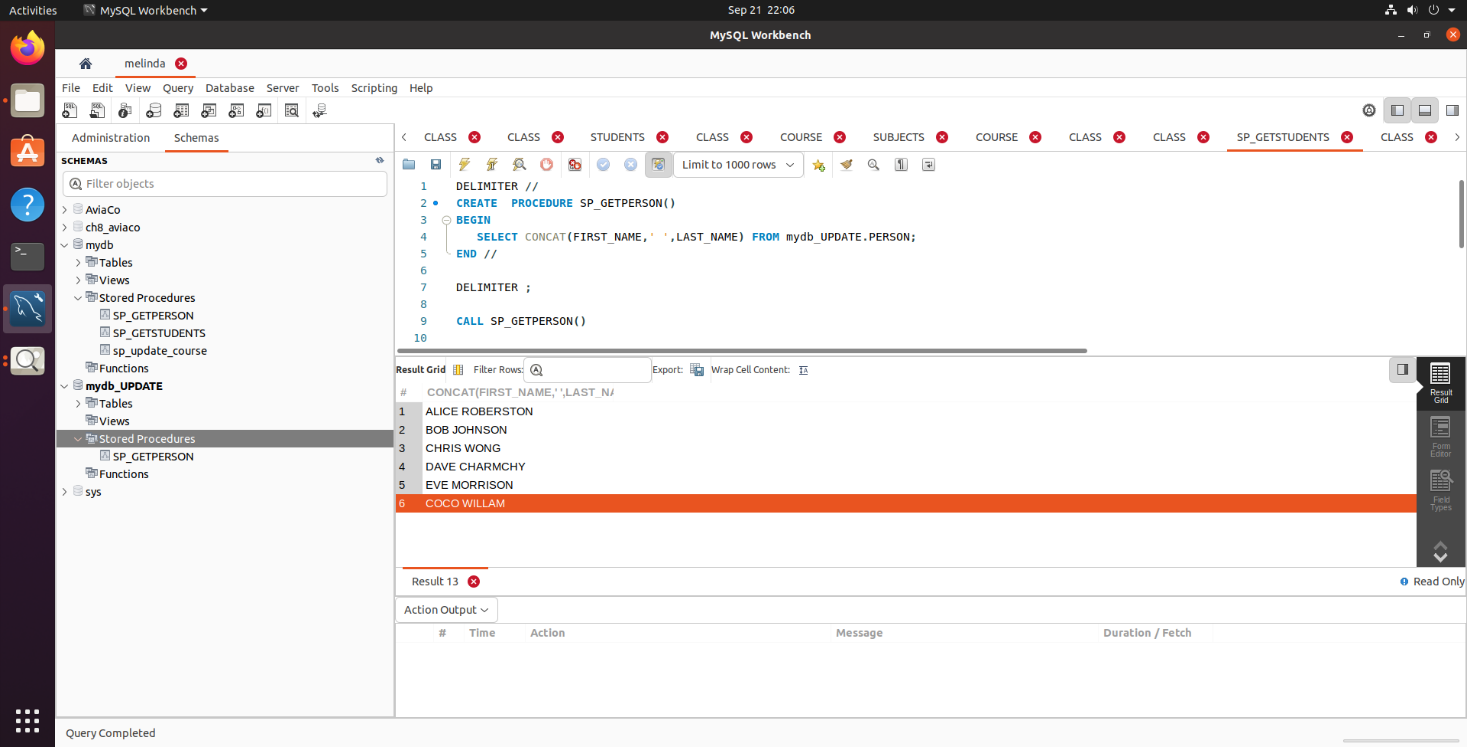


SELECT \* FROM mydb.SCORE\_GT\_82;

DROP VIEW mydb.SCORE\_GT\_82;

# Stored Procedures

## Retrieving Data Using a Stored Procedure



DELIMITER //

CREATE PROCEDURE SP\_GETPERSON()

BEGIN

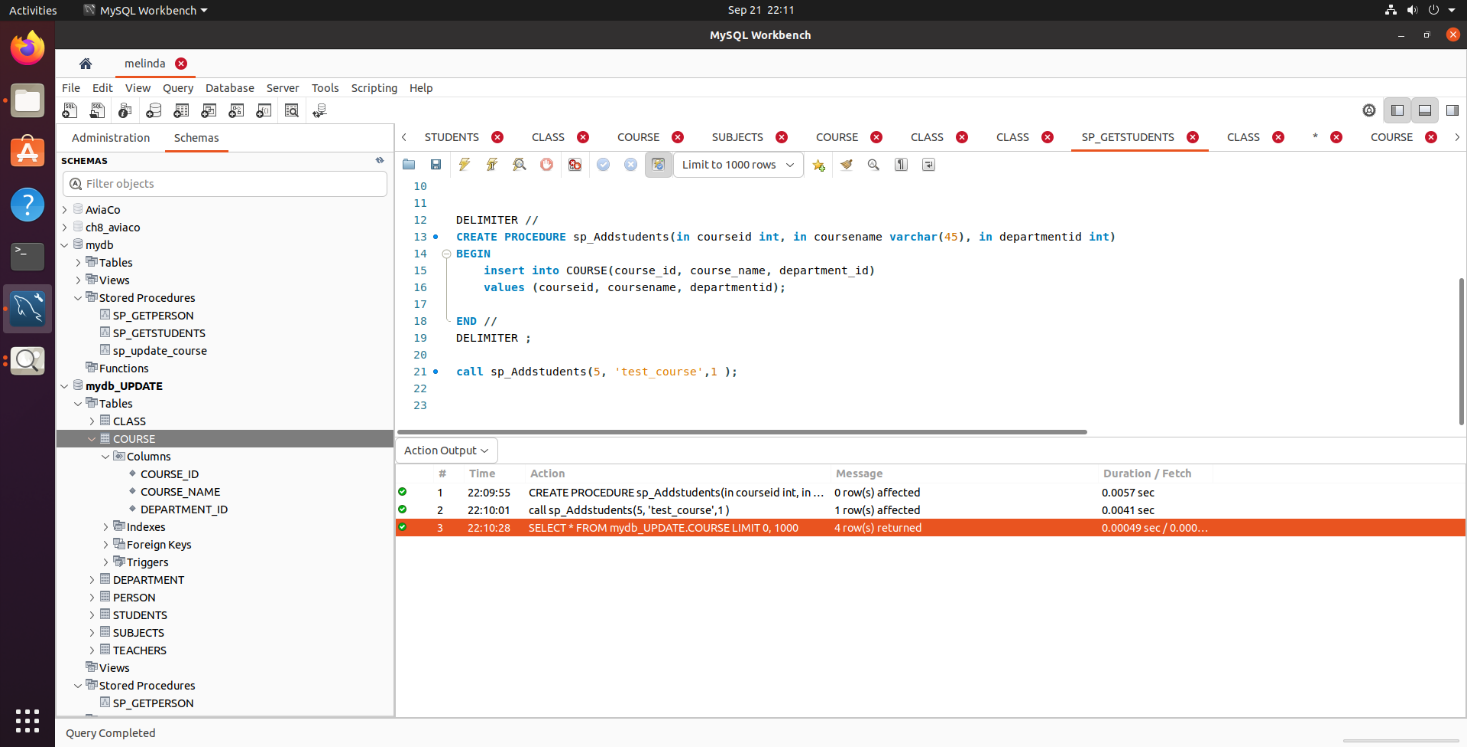
SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) FROM mydb\_UPDATE.PERSON;

END //

DELIMITER ;

CALL SP\_GETPERSON()

## Inserting Data Using a Stored Procedure



DELIMITER //

CREATE PROCEDURE sp\_Addstudents(in courseid int, in coursename varchar(45), in departmentid int)

BEGIN

insert into COURSE(course\_id, course\_name, department\_id)

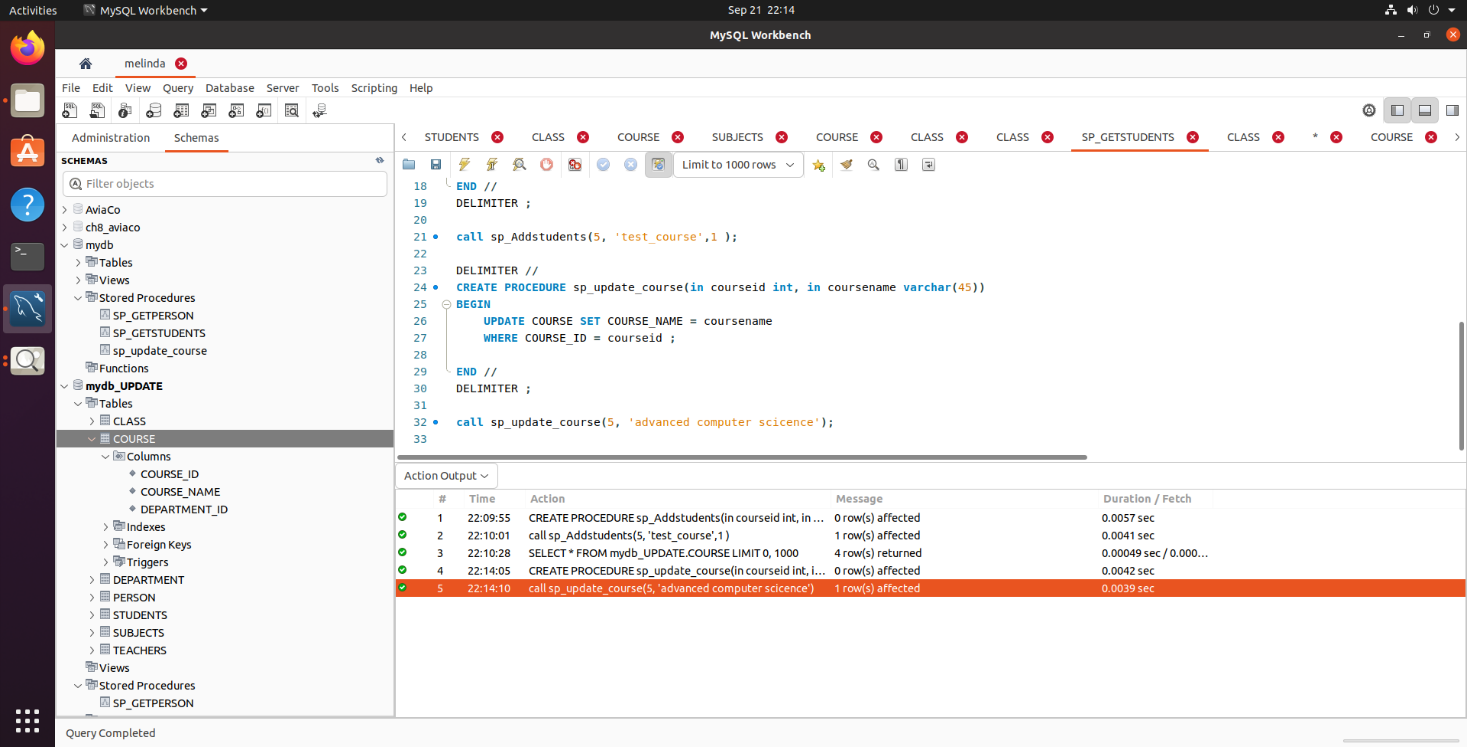
values (courseid, coursename, departmentid);

END //

DELIMITER ;

call sp\_Addstudents(5, 'test\_course',1 );

## Modifying Data Using a Stored Procedure



DELIMITER //

CREATE PROCEDURE sp\_update\_course(in courseid int, in coursename varchar(45))

BEGIN

UPDATE COURSE SET COURSE\_NAME = coursename

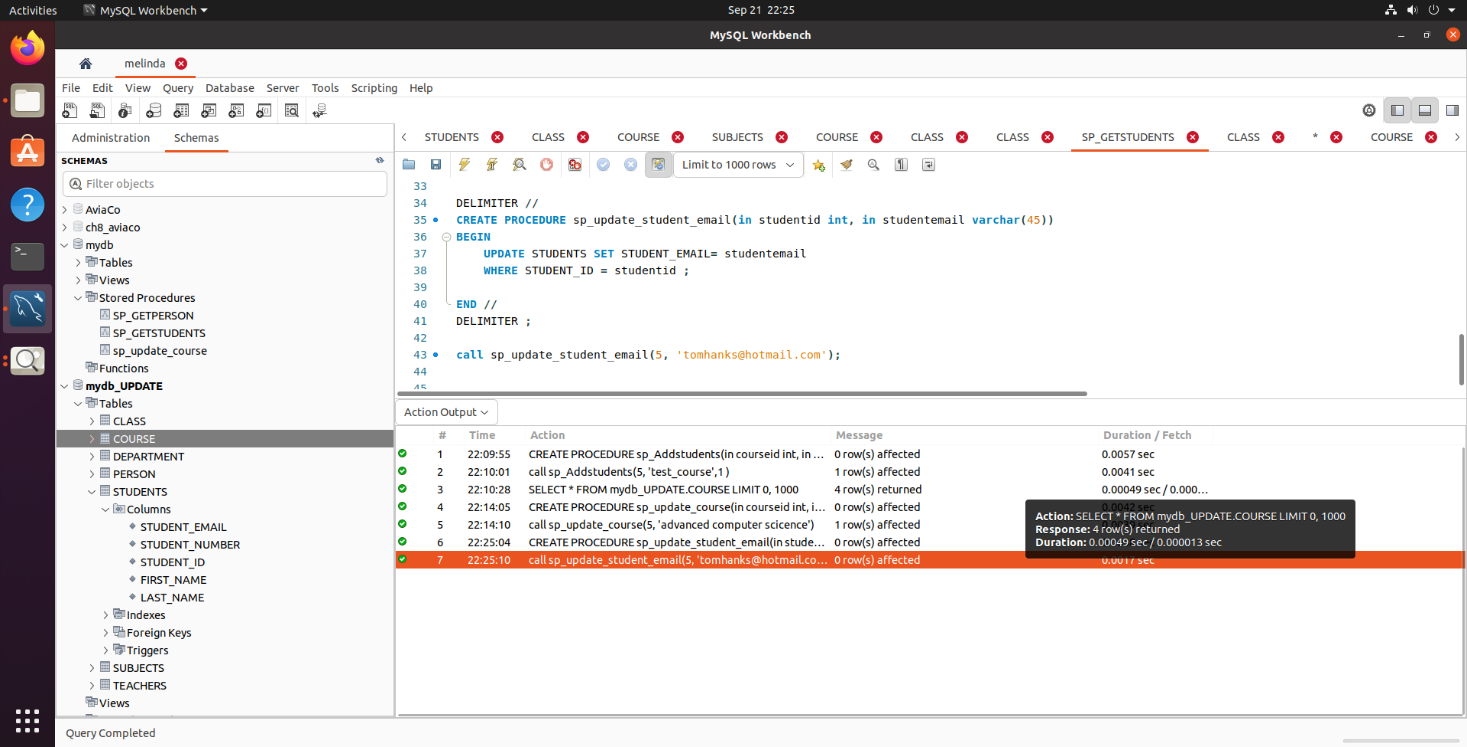
WHERE COURSE\_ID = courseid ;

END //

DELIMITER ;

call sp\_update\_course(5, 'advanced computer scicence');

## Stored Procedure Using a Parameter



DELIMITER //

CREATE PROCEDURE sp\_update\_student\_email(in studentid int, in studentemail varchar(45))

BEGIN

UPDATE STUDENTS SET STUDENT\_EMAIL= studentemail

WHERE STUDENT\_ID = studentid ;

END //

DELIMITER ;

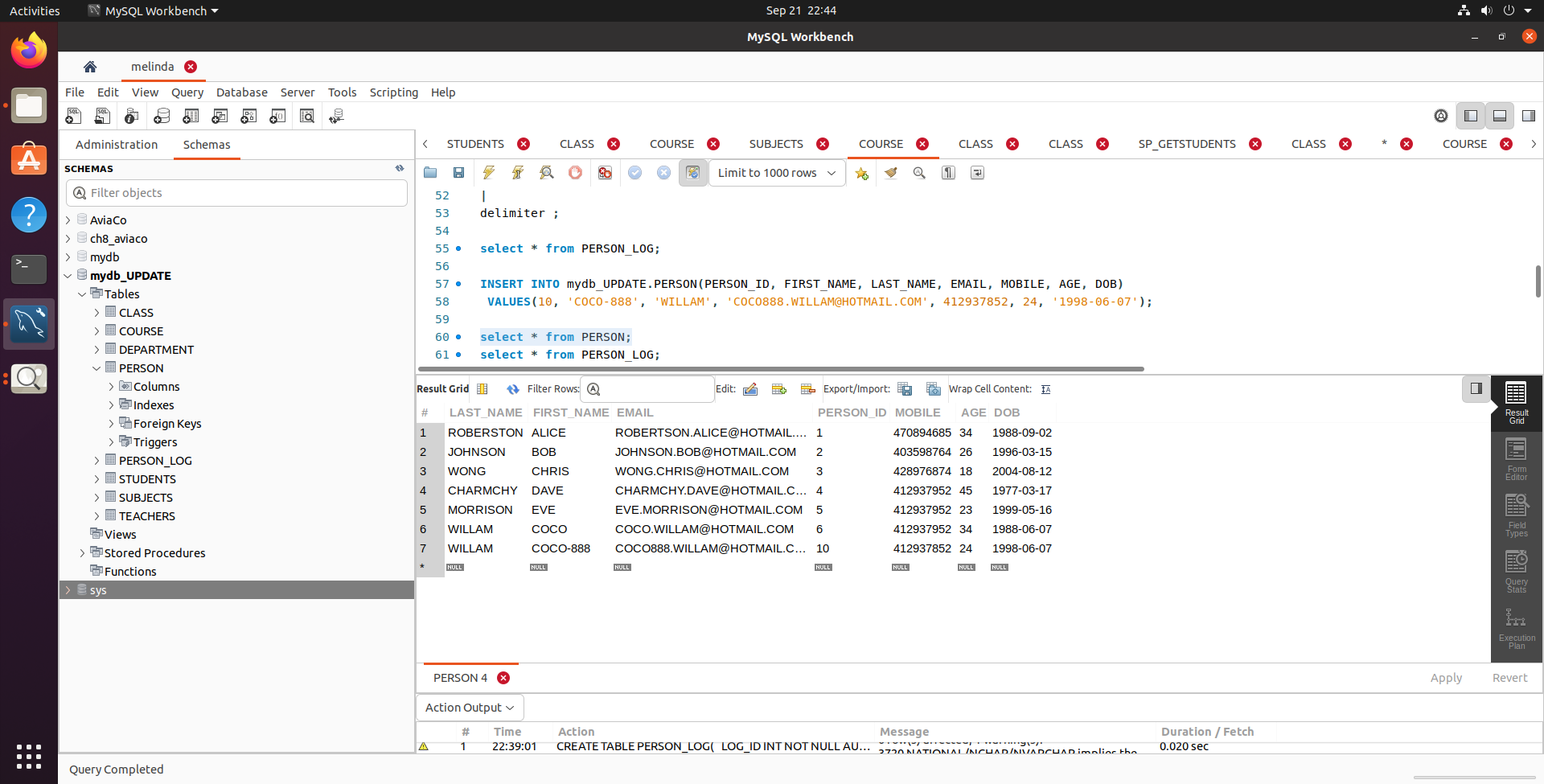
call sp\_update\_student\_email(5, 'tomhanks@hotmail.com');

## Dropping a Stored Procedure

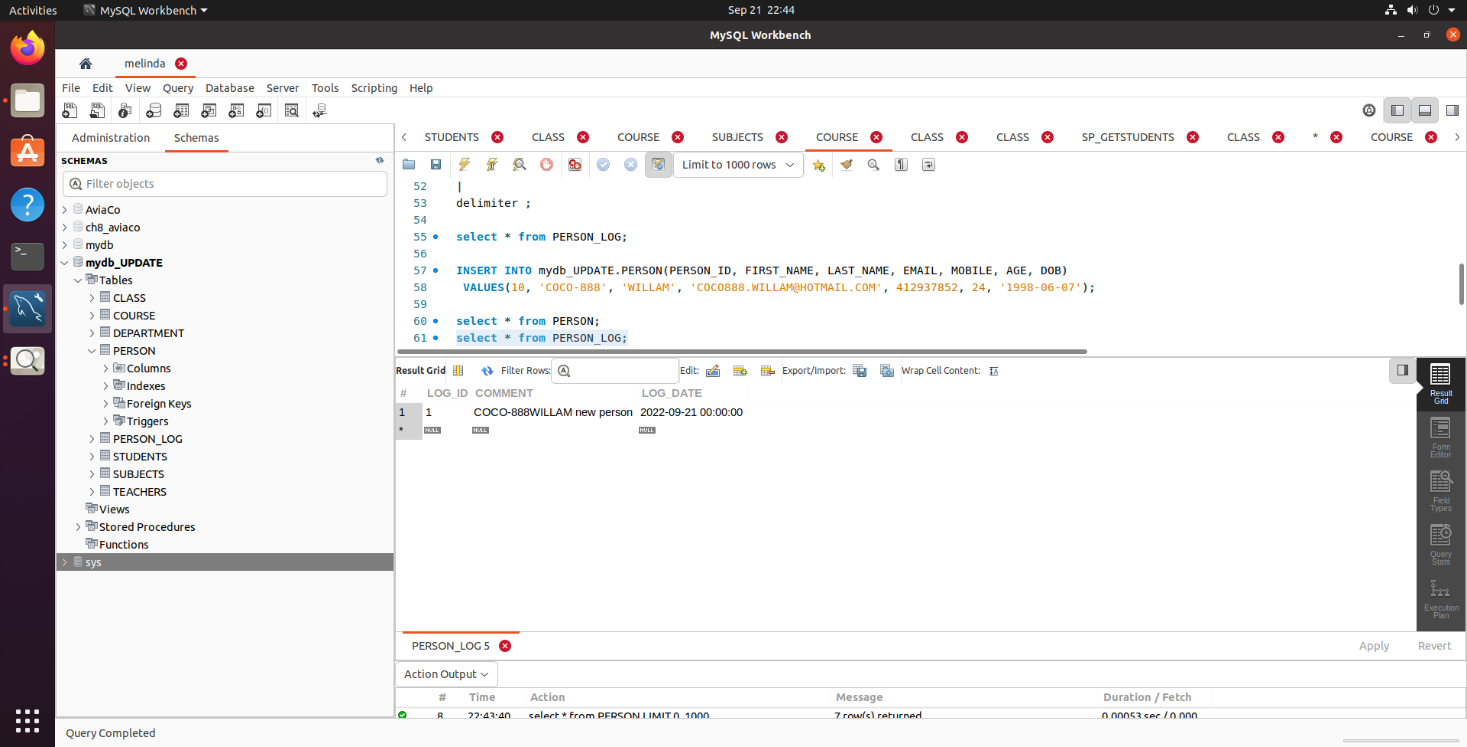
DROP PROCEDURE SP\_GETSTUDENTS

# Triggers

## Trigger 1



## Trigger 2



CREATE TABLE PERSON\_LOG(

LOG\_ID INT NOT NULL AUTO\_INCREMENT,

COMMENT nvarchar(500) NOT NULL,

LOG\_DATE datetime NOT NULL ,

primary key(LOG\_ID)

);

delimiter |

CREATE TRIGGER TRIG\_PERSON\_LOG

AFTER INSERT ON PERSON

FOR EACH ROW

BEGIN

INSERT INTO PERSON\_LOG

(COMMENT,LOG\_DATE)

VALUES

(concat(NEW.FIRST\_NAME, '', NEW.LAST\_NAME, ' new person'), curdate());

END;

|

delimiter ;

select \* from PERSON\_LOG;

INSERT INTO mydb\_UPDATE.PERSON(PERSON\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, MOBILE, AGE, DOB)

VALUES(10, 'COCO-888', 'WILLAM', 'COCO888.WILLAM@HOTMAIL.COM', 412937852, 24, '1998-06-07');

select \* from PERSON;

select \* from PERSON\_LOG;

my explanation about my trigger:

1. specify the name of the trigger is TRIG\_PERSON\_LOG
2. specify the trigger action time is AFTER, which indicates that the trigger is invoked after each row is modified.
3. The operation of activating the trigger is INSERT.
4. The name of table to which the trigger belongs PERSON
5. When the trigger invokes, the statement will be executed: insert a record including ‘COMMENT’ and ‘LOG\_DATE’ into table PERSON\_LOG

# Appendix