Online Education System

CS 6360.003- Database Design Project Report

Asok, Anagha AXA151631
Muniswamappa, Nandish NXM160630
Shankar, Madhumitha MXS162530

December 2017

Contents

INTRODUCTION

Online Educational system is a web based application where courses are offered online, and subscribers can register for the course and learn. In the modern world where computers in general, and internet in particular, have permeated the day to day life, such an application finds vivid influence. The technology lets learners access the lessons at their ease and convenience and bridges the gap between teacher and learner. There are so many existing applications in the web like ‘coursera’, ‘udemy’ etc. The application can be a simple one where a user subscribes and takes selects a course from the list of courses available to the ones where there are complex features like one-one lectures, certification for courses, paid courses etc. The whole idea is to integrate resources available across the globe and make it available to the learner. It includes Web-based learning, computer-based learning, virtual classrooms, satellite classrooms, video conferencing lectures etc. Content is delivered via the Internet, intranet/extranet, audio or video tape, satellite TV, CD-ROM, flash drive, video calls etc. It is used by the Universities and Schools to provide support their class room teaching. Many institutions maintain their own online education services and offers courses to a larger population of learners. It can be self-paced, or instructor led; includes media in the form of text, image, animation, streaming video and audio; has quizzes and exams with clear grading criteria etc.

Growth of online education services have opened a whole new market in the internet and opportunity for web developers, software developers and academic institutions. It made knowledge accessible to anyone who is interested and enthusiastic about learning.

The objective of this project is to design a database for an Online Education System. A research was conducted on the existing databases on such systems and it was found out that the fundamental requirements of such a system will be entities namely, Students, Instructors, Courses, Chapters, and Feedback. Each entity should have a primary key and relations to other entities. Chapters and Instructors should be connected to the respective course. There should be also entities like Exams, Assignments, Learning Resources, Discussion Forums related to chapters and students who are learning the chapters. Feedback should relate both a user and a course. The aim is to design a system which offers free access to all its users and provides no certification. The purpose is just to teach the learners in a systematic way, provide assignments and exams and grade them and get an effective feedback. Hence purpose of the design is as follows:

- A user can register to a system. User has so many attributes like name, email ID etc. and it relates to a course.
- A course should have instructor and chapters.
- Each instructor has attributes.
- Each course has number of chapters related to it and chapters have assignments, exams, discussions forums, learning resources and alerts.
- Each assignment and exam are related to the user as well as have a unique ID of their own.
- Discussion forums are also related to users and they can post different threads related to the course in there.
- Exams should have questions and it should also be related to students to mark which student answered which system.

Hence the design considered in mind initially and following sections discuss the detailed design and implementation.

1. DATA REQUIREMENTS

1. The fundamental element of the system is a user who is named as STUDENT_USERS who subscribe to the facility. While registering to the system the user need to have a NAME (FNAME, LNAME), EMAIL, BDATE, ADDRESS, PHONE_NUM and GENDER. He also needs to choose a unique USER_NAME and authenticate his credentials using a PASSWORD every time he logs in.

2. Another important element is course offered in the system and is names as COURSES. Each course has a CNAME, START_DATE and END_DATE. If end_date is past the system date the course has been completed, else the course is still going on. Hence STATUS is also an attribute of course which can be derived from the end_date. Each course is assigned a unique COURSE_ID. It has an attribute C_DESCRIPTION which gives a brief description about what the course is. Student_users can enroll in any number of courses they prefer. Also, more than one student can enroll to the same course. There can be courses within which no student has enrolled. And, a user can register in the system and prefer not to enroll in the courses.

3. INSTRUCTOR who teaches the courses offered is an integral part of the system. Each Instructor has NAME (FNAME, LNAME), EMAIL, BDATE, ADDRESS, PHONE_NUM and GENDER. Also, each of them is assigned a unique INSTRUCTOR_ID. An instructor can teach any number of courses and more than one instructor can teach the same course too. But if a course is listed, there must be at least one instructor who teaches that course. But an instructor who is not teaching a course right now can also exist in the system.

4. Each Course contains CHAPTERS under it. Chapters have CHNAME and CH_DESCRIPTION. Also, each chapter is assigned a unique CHAPTERID. Courses can have multiple chapters, but a chapter can only be part of one and only one course. Also, a course should have at-least one chapter.

5. Chapters have LEARNING RESOURCES under them. Learning resources are additional course materials provided to the students other than the contents for the chapter for self-study. These are mainly links to useful materials. Hence each of them have a unique LR_ID and a URL to the resource link. Each chapter can have multiple learning resources, but a learning resource can be and must be part of one and only one chapter. It is possible to have chapters without learning resources too.

6. Chapters maintain DISCUSSION FORUMS which are identified by unique DISS_ID. Discussion forums is also related to the participating student user and stores his EMAIL. It has a CDATE which the date on which the discussion started. It is related to student user via a relation JOIN which saves all the threads of that user as DTHREADS. One chapter can have only one discussion forum and a discussion forum can belong to only one chapter. Also, there can be chapters without discussion forums but if a discussion forum exists it must be related one and only one Chapter. A student can participate in multiple forums and one forum can have multiple students. Also, a student can decide not to participate in any forum and a forum can be empty without any discussion threads.

7. Chapters have HOMEWORK which are identified by unique HW_ID and it also has DUE_DATE as its attribute. Students can submit the homework and get points before due date. It is updated in TOTAL_POINTS. Each chapter can have only one homework and each home work must belong to one and only one chapter. There can be chapters without homework. Also, Homework is related to student user who can submit the homework. A student can opt not to submit a homework and homework need not be submitted by a student. A student can submit multiple homework but if a homework is submitted it must belong to only one student.

8. Similar to homework, chapters have EXAMS which are identifies by EXAM_ID and each exam has TERM it belongs to. A chapter can have multiple exams and if an exam exists it has to be and can be part of only one chapter. There can be chapters without exams too. Student users can take the exams and update the score every time they take it. Students can

take exams multiple times or opt not to take it. Also, there can be exams which are not taken by students and an exam can be taken by multiple students.

9. If exam exists, they have QUESTIONS. Each question is identified with a unique QUESTION_ID and it has a QUESTION in it. Each question must be part of one and only one exam, but an exam should have at-least one or more questions. Each question can be answered by a user. The ANSWER attribute has the answer given by student. A question can be answered by many students or can remain unanswered. Also, a student can opt not to answer any questions or answer multiple questions.

10. ALERTS are posted by the system under chapter at times. The alerts can be about the home works, exams or new learning materials. It is identified by a unique ALERT_ID. ALERT_DESCRIPTION contains the content of the alert and ALERT_TYPE is a multivalued attribute which contains the type of alert. i.e. The nature of alert and what it is about. An alert can be part of one and only one Chapter. But a chapter can have no or multiple alerts.

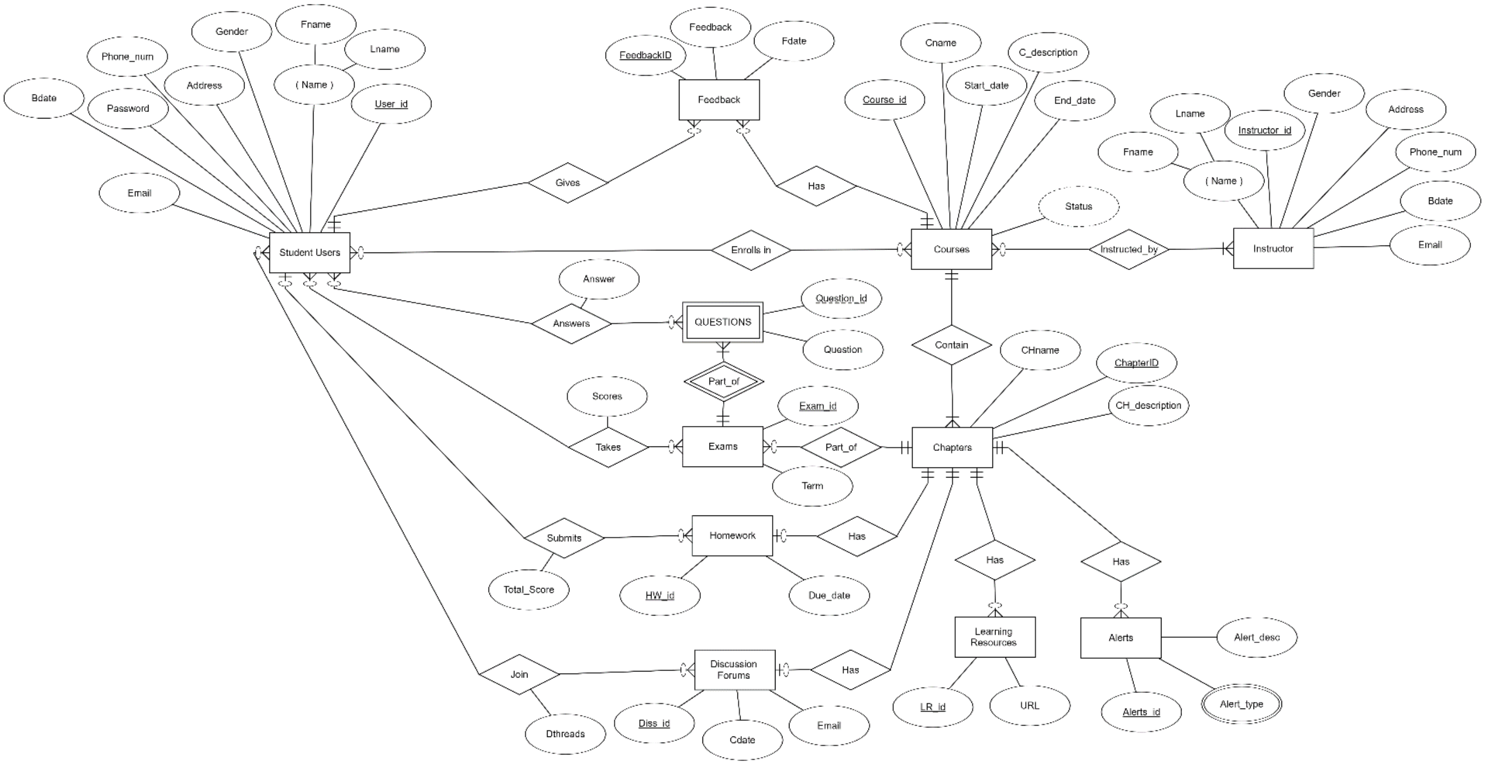
11. Each user can give a FEEDBACK about a course he is enrolled in. Each feedback has a unique FEEDBACK_ID with which it is identified. It has an attribute FEEDBACK which has the content of the feedback and FDATE which has the date on which the feedback was given. A student can give multiple feedbacks about a course or opt not to give a feedback. But, if a feedback exists it must be given my one and only one student.

12. The details of users who are enrolled in a course are stored in STUDENT_COURSE.

13. The details of marks obtained by users in exams are stored in USER_EXAM.

14. The details of marks obtained by users in homework are stored in HOMEWORK_MARKS.

2 ER DIAGRAM



Assumptions

- A STUDENT_USER can register in a COURSE only before the start date of the course.
- The status of the COURSE must be 1 when the system date is between the course start date & end date. (Implemented as a stored procedure)
- If there exists a COURSE, then it must be taught by at least one INSTRUCTOR.
- If there exists a LEARNING_RESOURCE, it must belong to a CHAPTER.
- If there exists an ALERT, it must belong to a CHAPTER.
- If there exists a HOMEWORK, it must belong to a CHAPTER.
- If there exists a DISCUSSION_FORUM then it must belong to a CHAPTER.
- QUESTIONS need to exist only if EXAMS exist.
- FEEDBACK is for a given COURSE by a given STUDENT_USER.

- A SCORES for EXAM cannot be more than 100. (Implemented as trigger)
- The ALERT with ALERT_ID one contains the detailed structure of the COURSE and it cannot be DELETED. (Implemented as Trigger)
- If CH_DESCRIPTION is “null” it should be shown as “Chapter description not provided”. (Implemented as stored procedure).

2.1 One-to-one binary relationships

- Each CHAPTER has an HOMEWORK. One HOMEWORK belongs to only one CHAPTER.
- Each CHAPTER has a DISCUSSION_FORUM. One DISCUSSION_FORUM belongs to only one CHAPTER.

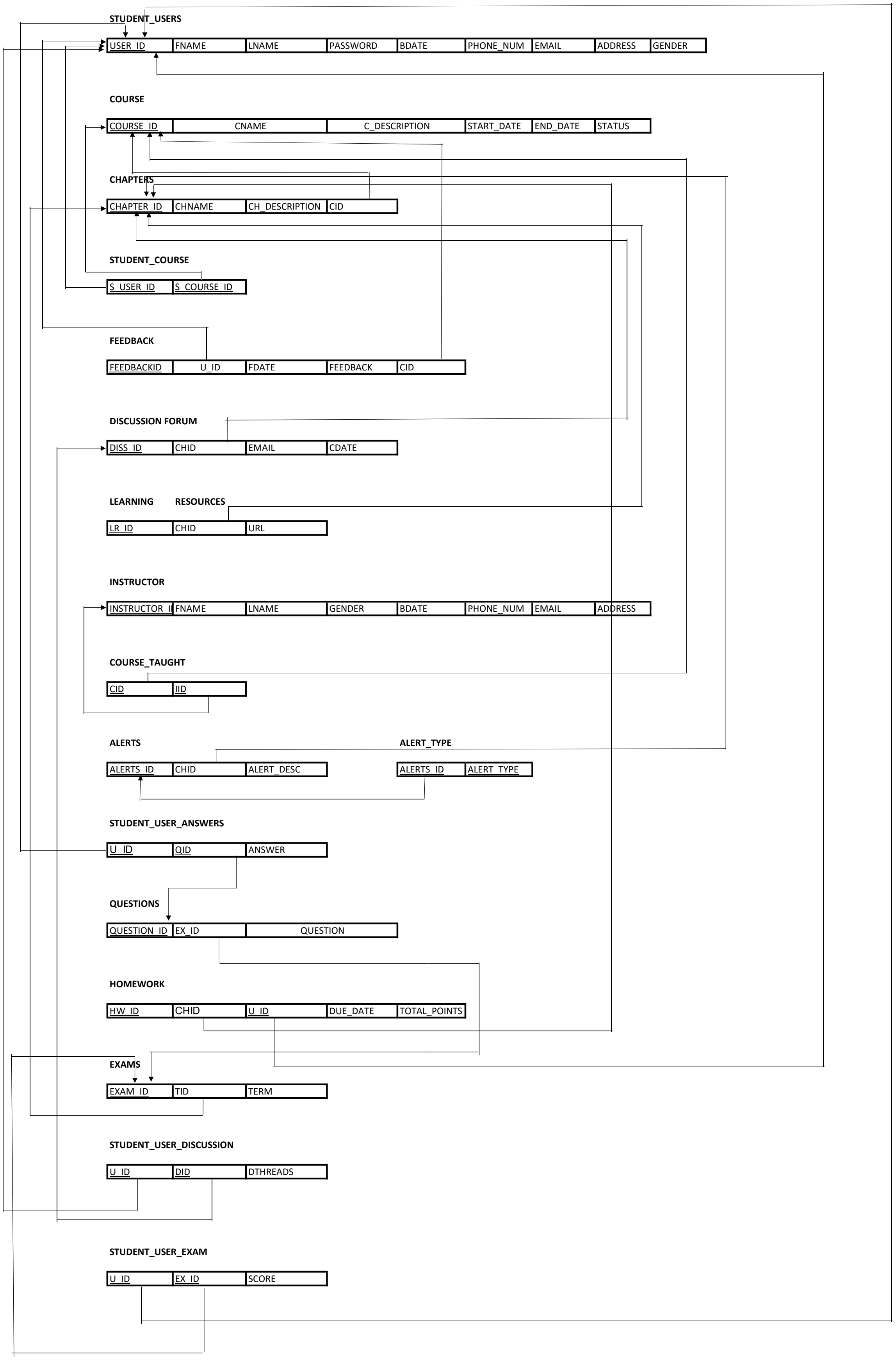
One-to-many binary relationships

- Each EXAM has multiple questions. A QUESTION can belong only one EXAM.
- Each EXAM belongs to a single CHAPTER. There can be multiple EXAMS conducted for a CHAPTER.
- Each COURSE has multiple CHAPTERS. A CHAPTER can belong to only one COURSE.
- There can be many LEARNING_RESOURCES for a CHAPTER. A LEARNING_RESOURCE can belong to only a single CHAPTER.
- There can be many ALERTS posted for a single CHAPTER. An ALERT belongs to a single CHAPTER.
- Each STUDENT_USER can submit multiple HOMEWORK. If submitted a HOMEWORK should belong only to one STUDENT_USER
- A STUDENT_USER can give multiple FEEDBACKS. A FEEDBACK is given by only one STUDENT_USER.
- A COURSE can have multiple FEEDBACKS. A FEEDBACK can belong to only one COURSE.

Many-to-many binary relationships

- A STUDENT_USER can enroll in multiple COURSES. A COURSE can be taken by multiple STUDENT_USERs.
- A COURSE can be taught by multiple number of INSTRUCTORS. An INSTRUCTOR can teach multiple COURSES.
- Each STUDENT_USER answers multiple QUESTIONS. A QUESTION can be answered by multiple STUDENT_USERS.
- A STUDENT_USER can take multiple EXAMS. Each EXAM can be taken by multiple STUDENT_USER.
- Each STUDENT_USER can participate in multiple DISCUSSION_FORUMs at the same time. A DISCUSSION_FORUM has multiple STUDENT_USERS.

3. MAPPING OF ER DIAGRAM INTO RELATIONAL SCHEMA



STUDENT_USERS

USER_ID	FNAME	LNAME	PASSWORD	BDATE	PHONE_NUM	EMAIL	ADDRESS	GENDER

COURSE

COURSE_ID	CNAME	C_DESCRIPTION	START_DATE	END_DATE	STATUS

CHAPTERS

CHAPTER_ID	CHNAME	CH_DESCRIPTION	CID

STUDENT_COURSE

S_USER_ID	S_COURSE_ID

FEEDBACK

FEEDBACKID	U_ID	FDATE	FEEDBACK	CID

DISCUSSION FORUM

DISS_ID	CHID	EMAIL	CDATE

LEARNING RESOURCES

LR_ID	CHID	URL

INSTRUCTOR

INSTRUCTOR_I	FNAME	LNAME	GENDER	BDATE	PHONE_NUM	EMAIL	ADDRESS

COURSE_TAUGHT

CID	IID

ALERTS

ALERTS_ID	CHID	ALERT_DESC

ALERT_TYPE

ALERTS_ID	ALERT_TYPE

STUDENT_USER_ANSWERS

U_ID	QID	ANSWER

QUESTIONS

QUESTION_ID	EX_ID	QUESTION

HOMEWORK

HW_ID	CHID	U_ID	DUE_DATE	TOTAL_POINTS

EXAMS

EXAM_ID	TID	TERM

STUDENT_USER_DISCUSSION

U_ID	DID	DTHREADS

STUDENT_USER_EXAM

U_ID	EX_ID	SCORE

4 DATABASE NORMALIZATION

Functional Dependencies for the System

STUDENT_USERS

USER_ID→FNAME, LNAME, PASSWORD, BDATE, PHONE_NUM, EMAIL, ADDRESS, GENDER

COURSE

COURSE_ID →CNAME, C_DESCRIPTION, START_DATE, END_DATE

CHAPTERS

CHAPTER_ID → CHNAME, CH_DESCRIPTION, CID

FEEDBACK

FEEDBACK_ID → U_ID, FDATE, FEEDBACK, F_C_ID

DISCUSSION_FORUM

DISS_ID → CHID, EMAIL, CDATE

LEARNING_RESOURCES

LR_ID → CHID, URL

INSTRUCTOR

INSTRUCTOR_ID →FNAME, LNAME, GENDER, BDATE, PHONE_NUM, EMAIL, ADDRESS

ALERTS

ALERTS_ID → CHID, ALERTS _DESC

USER_ANSWERS

U_ID, QID → ANSWER

QUESTIONS

QUESTION_ID → EX_ID, QUESTION

HOMEWORK

HW_ID, U_ID→ CHID, DUE_DATE, TOTAL_POINTS

HW_ID→CHID, DUE_DATE

EXAMS

EXAM_ID → CHID, TERM

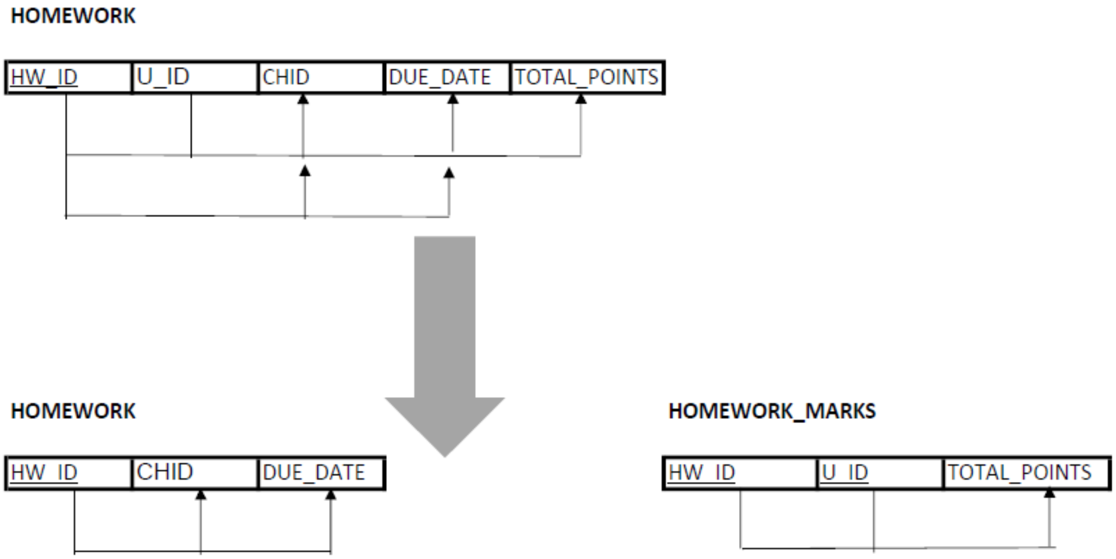
STUDENT_USER_DISCUSSION

U_ID, DID → DTHREADS

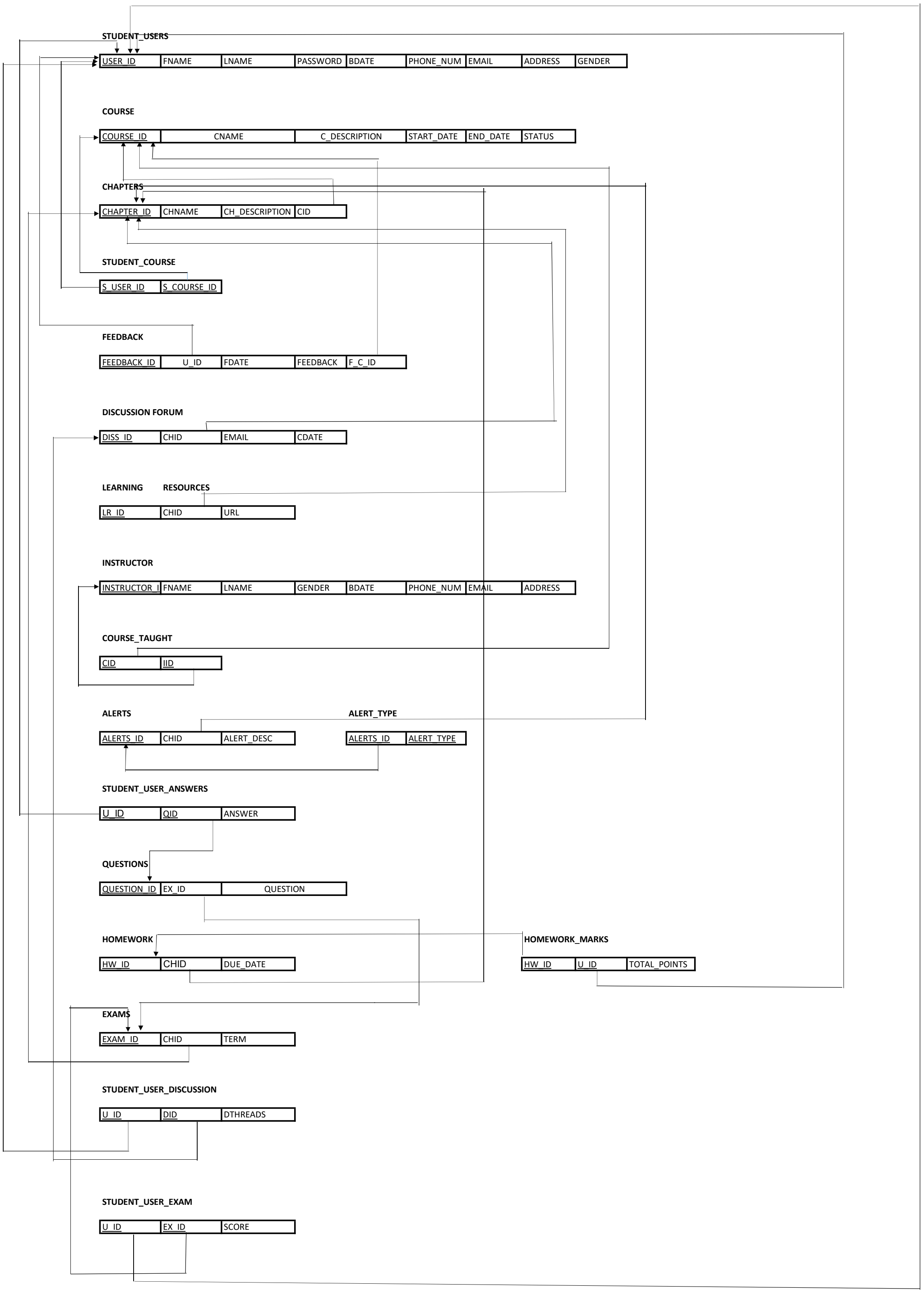
STUDENT_USER_EXAM

U_ID, EX_ID → SCORE

Normalization



5.FINAL RELATIONAL SCHEMA AFTER NORMALIZATION



STUDENT_USERS

USER_ID	FNAME	LNAME	PASSWORD	BDATE	PHONE_NUM	EMAIL	ADDRESS	GENDER

COURSE

COURSE_ID	CNAME	C_DESCRIPTION	START_DATE	END_DATE	STATUS

CHAPTERS

CHAPTER_ID	CHNAME	CH_DESCRIPTION	CID

STUDENT_COURSE

S_USER_ID	S_COURSE_ID

FEEDBACK

FEEDBACK_ID	U_ID	FDATE	FEEDBACK	F_C_ID

DISCUSSION FORUM

DISS_ID	CHID	EMAIL	CDATE

LEARNING RESOURCES

LR_ID	CHID	URL

INSTRUCTOR

INSTRUCTOR_I	FNAME	LNAME	GENDER	BDATE	PHONE_NUM	EMAIL	ADDRESS

COURSE_TAUGHT

CID	IID

ALERTS

ALERTS_ID	CHID	ALERT_DESC

ALERT_TYPE

ALERTS_ID	ALERT_TYPE

STUDENT_USER_ANSWERS

U_ID	QID	ANSWER

QUESTIONS

QUESTION_ID	EX_ID	QUESTION

HOMEWORK

HW_ID	CHID	DUE_DATE

HOMEWORK_MARKS

HW_ID	U_ID	TOTAL_POINTS

EXAMS

EXAM_ID	CHID	TERM

STUDENT_USER_DISCUSSION

U_ID	DID	DTHREADS

STUDENT_USER_EXAM

U_ID	EX_ID	SCORE

6 SQL CODE

DROP TABLE :

```
DROP TABLE USER_EXAM;
DROP TABLE USER_DISCUSSION;
DROP TABLE USER_ANSWERS;
DROP TABLE QUESTIONS;
DROP TABLE EXAM;
DROP TABLE HOMEWORK_MARKS;
DROP TABLE HOMEWORK;
DROP TABLE ALERT_TYPE;
DROP TABLE ALERT;
DROP TABLE COURSE_INSTRUCTED;
DROP TABLE INSTRUCTOR;
DROP TABLE LEARNING_RESOURCES;
DROP TABLE DISCUSSION_FORUM;
DROP TABLE FEEDBACK;
DROP TABLE STUDENT_COURSE;
DROP TABLE CHAPTER;
DROP TABLE COURSE;
DROP TABLE STUDENTUSERS;
```

```
insert into STUDENTUSERS (USER_ID, FNAME, LNAME, PASSWORD, BDATE, PHONE_NUM, EMAIL, ADDRESS, GENDER) values (1,'madhumitha','shankar','111','29-JAN-94','4696567878','mxs@gmail.com','1412 Marquis Apts','f');

insert into STUDENTUSERS (USER_ID, FNAME, LNAME, PASSWORD, BDATE, PHONE_NUM, EMAIL, ADDRESS, GENDER) values (2,'anagha','asok','121','03-MAR-93','8989876576','axa@gmail.com','7774 Mcclum blvd','f');

insert into STUDENTUSERS (USER_ID, FNAME, LNAME, PASSWORD, BDATE, PHONE_NUM, EMAIL, ADDRESS, GENDER) values (3,'nandish','kumar','125','16-MAR-92','8787765576','nxk@gmail.com','7825 Mcclum blvd','f');
insert into STUDENTUSERS (USER_ID, FNAME, LNAME, PASSWORD, BDATE, PHONE_NUM,
EMAIL, ADDRESS, GENDER) values (4,'kavitha','raman','443','19-NOV-
93','8988801176','kxr@gmail.com','7904 Mcclum blvd','m');

insert into STUDENTUSERS (USER_ID, FNAME, LNAME, PASSWORD, BDATE, PHONE_NUM,
EMAIL, ADDRESS, GENDER) values (5,'spandana','bellamkonda','187','18-DEC-
92','9011909576','sxb@gmail.com','7000Mcclum blvd','f');


insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (1,'database design','database design and storage analysis','01-OCT-
17','01-DEC-17',0);

insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (2,'microprocessor','computer processor that incopoerates computer CPU
on a single intergrated circuit','01-OCT-17','01-DEC-17',0);

insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (3,'ML','pattern recognition and computational learning theory in artificial
intelligence','01-NOV-17','01-JAN-18',0);

insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (4,'ACN','digital telecommunication network which allows nodes to share
resources','01-NOV-17','01-FEB-18',0);

insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (5,'DSP','processing of analog signals to digital signals to make it
compatible with digital computer used today','01-MAY-18','01-AUG-18',0);

insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (6,'ADL','implementation of logic gates in digital system','01-JAN-18','01-
MAR-18',0);

insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (8,'sdn','transferring control components to software','05-DEC-17','01-
JAN-18',0);

insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (7,'OS','study of software that supports computers basic functions','01-
DEC-17','01-JAN-18',0);


insert into CHAPTER (CHAPTER_ID,CHNAME, CH_DESCRIPTION,CID) values
(6,'DB','Introduction, actors, advantages and disadvantages',1);

insert into CHAPTER (CHAPTER_ID,CHNAME, CH_DESCRIPTION,CID) values (7 ,'protocols','OSI
layers and TCP protocols',4);

insert into CHAPTER (CHAPTER_ID,CHNAME, CH_DESCRIPTION,CID) values (3,'IO
operations','Introduction to digital logic systems',2);
```

```
insert into CHAPTER (CHAPTER_ID,CHNAME, CH_DESCRIPTION,CID) values (5,'SDN Intro','Evolution of switches and control planes',8);

insert into CHAPTER (CHAPTER_ID,CHNAME, CH_DESCRIPTION,CID) values (1,'ACN Intro','communication,switching techniques in computer networks',4);

insert into CHAPTER (CHAPTER_ID,CHNAME, CH_DESCRIPTION,CID) values (4,'ML Intro','statistical methods in ML',3);

insert into CHAPTER (CHAPTER_ID,CHNAME, CH_DESCRIPTION,CID) values (2,'Frequency','fourier transforms,complex signals',5);

insert into CHAPTER (CHAPTER_ID,CHNAME, CH_DESCRIPTION,CID) values (9,'System',null,7);


insert into STUDENT_COURSE (S_USER_ID, S_COURSE_ID) values (1,1);

insert into STUDENT_COURSE (S_USER_ID, S_COURSE_ID) values (1,4);

insert into STUDENT_COURSE (S_USER_ID, S_COURSE_ID) values (2,2);

insert into STUDENT_COURSE (S_USER_ID, S_COURSE_ID) values (2,4);

insert into STUDENT_COURSE (S_USER_ID, S_COURSE_ID) values (2,1);

insert into STUDENT_COURSE (S_USER_ID, S_COURSE_ID) values (3,3);

insert into STUDENT_COURSE (S_USER_ID, S_COURSE_ID) values (3,2);

insert into STUDENT_COURSE (S_USER_ID, S_COURSE_ID) values (4,1);

insert into STUDENT_COURSE (S_USER_ID, S_COURSE_ID) values (4,2);

insert into STUDENT_COURSE (S_USER_ID, S_COURSE_ID) values (5,2);


insert into FEEDBACK (FEEDBACK_ID,U_ID, FDATE,CID,FEEDBACK) values (1,2,'15-JAN-17',3,'very useful study material');

insert into FEEDBACK (FEEDBACK_ID,U_ID, FDATE,CID,FEEDBACK) values (2,3,'20-DEC-17',4,'lectures need to be more detailed');

insert into FEEDBACK (FEEDBACK_ID,U_ID, FDATE,CID,FEEDBACK) values (3,4,'09-NOV-17',1,'good assignments');

insert into FEEDBACK (FEEDBACK_ID,U_ID, FDATE,CID,FEEDBACK) values (4,5,'22-JAN-17',5,'very tough quizzes');

insert into FEEDBACK (FEEDBACK_ID,U_ID, FDATE,CID,FEEDBACK) values (5,1,'11-OCT-17',2,'very poor video quality');


insert into DISCUSSION_FORUM(DISS_ID, CHID,EMAIL,CDATE) values (1,1,'mxs@gmail.com','01-NOV-17');
insert into DISCUSSION_FORUM(DISS_ID, CHID, EMAIL,CDATE) values (2,2,'axa@gmail.com','15-NOV-17');

insert into DISCUSSION_FORUM(DISS_ID, CHID,EMAIL, CDATE) values (3,3,'nxk@gmail.com','01-FEB-17');

insert into DISCUSSION_FORUM(DISS_ID, CHID,EMAIL, CDATE) values (4,4,'kxr@gmail.com','29-JAN-17');

insert into DISCUSSION_FORUM(DISS_ID, CHID,EMAIL, CDATE) values (5,5,'sxb@gmail.com','11-NOV-17');

insert into DISCUSSION_FORUM(DISS_ID, CHID,EMAIL, CDATE) values (6,6,'ropd@gmail.com','01-DEC-17');


insert into USER_DISCUSSION(U_ID,DID ,DTHREADS) values (2,1,'What is frequency analysis');

insert into USER_DISCUSSION(U_ID,DID ,DTHREADS) values (3,2, 'What is sdn');

insert into USER_DISCUSSION(U_ID,DID ,DTHREADS) values (4,3, 'What is nosql');

insert into USER_DISCUSSION(U_ID,DID ,DTHREADS) values (2,4, 'What is sql');

insert into USER_DISCUSSION(U_ID,DID ,DTHREADS) values (3,5, 'What is networking');

insert into USER_DISCUSSION(U_ID,DID ,DTHREADS) values (1,6, 'What is eigen function');



insert into LEARNING_RESOURCES(LR_ID, CHID, URL) values (1,1,'https://www.google.com');

insert into LEARNING_RESOURCES(LR_ID, CHID, URL) values (2,3,' https://www.eecs.umich.edu/courses/eecs373/labs/refs/M3%20Guide.pdf');

insert into LEARNING_RESOURCES(LR_ID, CHID, URL) values (4,2,' https://en.wikipedia.org/wiki/frequency');

insert into LEARNING_RESOURCES(LR_ID, CHID, URL) values (5,7,' https://en.wikipedia.org/wiki/protocols');

insert into LEARNING_RESOURCES(LR_ID, CHID, URL) values (6,7,' http://intronetworks.cs.luc.edu/current/ComputerNetworks.pdf');

insert into LEARNING_RESOURCES(LR_ID, CHID, URL) values (7,4,' https://en.wikipedia.org/wiki/ml');

insert into LEARNING_RESOURCES(LR_ID, CHID, URL) values (8,5,' https://en.wikipedia.org/wiki/sdn');
insert into
INSTRUCTOR(INSTRUCTOR_ID,FNAME,LNAME,GENDER,BDATE,PHONE_NUM,EMAIL,ADD
RESS)values (1,'prem','kumar','m','23-MAY-1963','8780076576','prem@gmail.com','794 Marquis Apts');

insert into
INSTRUCTOR(INSTRUCTOR_ID,FNAME,LNAME,GENDER,BDATE,PHONE_NUM,EMAIL,ADD
RESS)values (2,'sharvani','reddy','f','10-JAN-1973','8119876576','sharu@gmail.com','7829 Mcclum
blvd');

insert into
INSTRUCTOR(INSTRUCTOR_ID,FNAME,LNAME,GENDER,BDATE,PHONE_NUM,EMAIL,ADD
RESS)values (3,'roy','bing','m','22-JAN-1979','8789876076','roy@gmail.com','7177 chattam courts');

insert into
INSTRUCTOR(INSTRUCTOR_ID,FNAME,LNAME,GENDER,BDATE,PHONE_NUM,EMAIL,ADD
RESS)values (4,'nandi','prasad','m','11-FEB-1980','9279876576','nandi@gmail.com','9824 Mcclum blvd');

insert into
INSTRUCTOR(INSTRUCTOR_ID,FNAME,LNAME,GENDER,BDATE,PHONE_NUM,EMAIL,ADD
RESS)values (5,'neha','sharma','f','20-OCT-1982','8789877894','neha@gmail.com','7884 Mcclum blvd');

insert into
INSTRUCTOR(INSTRUCTOR_ID,FNAME,LNAME,GENDER,BDATE,PHONE_NUM,EMAIL,ADD
RESS)values (6,'joey','tribiani','m','11-OCT-1983','9729870576','joey@gmail.com','7824 Mcclum blvd');


insert into COURSE_INSTRUCTED(CID ,IID ) values (1,1);

insert into COURSE_INSTRUCTED(CID ,IID ) values (2,2);

insert into COURSE_INSTRUCTED(CID ,IID ) values (3,3);

insert into COURSE_INSTRUCTED(CID ,IID ) values (4,4);

insert into COURSE_INSTRUCTED(CID ,IID ) values (5,5);

insert into COURSE_INSTRUCTED(CID ,IID ) values (6,1);

insert into COURSE_INSTRUCTED(CID ,IID ) values (8,6);

insert into COURSE_INSTRUCTED(CID ,IID ) values (5,2);


insert into ALERT(ALERT_ID,CHID,ALERT_DESC)values (1,1,'the following is the detailed course
structure of the course');

insert into ALERT(ALERT_ID,CHID,ALERT_DESC)values (2,2,'Exams dates are finalised and posted');

insert into ALERT(ALERT_ID,CHID,ALERT_DESC)values (3,3,'Assignment posted');

insert into ALERT(ALERT_ID,CHID,ALERT_DESC)values (4,4,'Project posted');

insert into ALERT(ALERT_ID,CHID,ALERT_DESC)values (5,5,'Additional practice problems posted');
insert into ALERT(ALERT_ID,CHID,ALERT_DESC)values (6,6,'Scores are updated');
insert into ALERT(ALERT_ID,CHID,ALERT_DESC)values (8,1,'Project topics posted');
insert into ALERT(ALERT_ID,CHID,ALERT_DESC)values (7,1,'Class postponed');

insert into ALERT_TYPE(ALERT_ID,ALERT_TYPE)values (1,'Class');
insert into ALERT_TYPE(ALERT_ID,ALERT_TYPE)values (2,'Exam');
insert into ALERT_TYPE(ALERT_ID,ALERT_TYPE)values (3,'Assignment');
insert into ALERT_TYPE(ALERT_ID,ALERT_TYPE)values (4,'Project');
insert into ALERT_TYPE(ALERT_ID,ALERT_TYPE)values (5,'HW');
insert into ALERT_TYPE(ALERT_ID,ALERT_TYPE)values (6,'Scores');
insert into ALERT_TYPE(ALERT_ID,ALERT_TYPE)values (7,'Class');
insert into ALERT_TYPE(ALERT_ID,ALERT_TYPE)values (8,'Project');

insert into EXAM(EXAM_ID,CHID,TERM) values (1,1,2);
insert into EXAM(EXAM_ID,CHID,TERM) values (2,2,3);
insert into EXAM(EXAM_ID,CHID,TERM) values (3,2,3);
insert into EXAM(EXAM_ID,CHID,TERM) values (4,1,1);
insert into EXAM(EXAM_ID,CHID,TERM) values (5,3,2);
insert into EXAM(EXAM_ID,CHID,TERM) values (6,4,3);
insert into EXAM(EXAM_ID,CHID,TERM) values (7,1,1);
insert into EXAM(EXAM_ID,CHID,TERM) values (8,5,2);
insert into EXAM(EXAM_ID,CHID,TERM) values (9,6,1);
insert into EXAM(EXAM_ID,CHID,TERM) values (10,7,1);
insert into EXAM(EXAM_ID,CHID,TERM) values (11,2,2);
insert into EXAM(EXAM_ID,CHID,TERM) values (12,1,1);
insert into EXAM(EXAM_ID,CHID,TERM) values (13,6,2);

insert into QUESTIONS(QUESTION_ID, EX_ID,QUESTION) values (1,2,'What is FourierTransform ');
insert into QUESTIONS(QUESTION_ID, EX_ID,QUESTION) values (2,1,'What is SQL');
insert into QUESTIONS(QUESTION_ID, EX_ID,QUESTION) values (3,1,'What is NOSQL');
insert into QUESTIONS(QUESTION_ID, EX_ID,QUESTION) values (4,2,'What is Ztransform');

insert into QUESTIONS(QUESTION_ID, EX_ID,QUESTION) values (5,5,'WHat is digital logic');

insert into QUESTIONS(QUESTION_ID, EX_ID,QUESTION) values (6,6,'What is Statiscal methods');

insert into QUESTIONS(QUESTION_ID, EX_ID,QUESTION) values (7,7,'What is DATA
ANALYSIS');

insert into QUESTIONS(QUESTION_ID, EX_ID,QUESTION) values (8,8,'What is UNIVARIATION');

insert into QUESTIONS(QUESTION_ID, EX_ID,QUESTION) values (9,9,'What is MULTI
TASKING');


insert into USER_ANSWERS(QID, U_ID, ANSWER) values (1,1,'decomposes function of time into
frequency ');

insert into USER_ANSWERS(QID, U_ID, ANSWER) values (1,2,'mnopq ');

insert into USER_ANSWERS(QID, U_ID, ANSWER) values (9,4,'abcd');

insert into USER_ANSWERS(QID, U_ID, ANSWER) values (5,5,'efgh ');

insert into USER_ANSWERS(QID, U_ID, ANSWER) values (8,2,'ghij');

insert into USER_ANSWERS(QID, U_ID, ANSWER) values (6,3,'klmn');

insert into USER_ANSWERS(QID, U_ID, ANSWER) values (3,4,'uvwx');



insert into USER_EXAM(u_id,EX_ID,score) values(1,1,90);

insert into USER_EXAM (u_id,EX_ID,score) values(2,1,80);

insert into USER_EXAM(u_id,EX_ID,score) values(3,10,95);

insert into USER_EXAM(u_id,EX_ID,score) values(4,9,70);

insert into USER_EXAM (u_id,EX_ID,score) values(5,5,80);

insert into USER_EXAM(u_id,EX_ID,score) values(1,12,60);

insert into USER_EXAM(u_id,EX_ID,score) values(2,8,89);

insert into USER_EXAM (u_id,EX_ID,score) values(3,6,90);

insert into USER_EXAM (u_id,EX_ID,score) values(4,3,100);


insert into HOMEWORK (HW_ID,CHID,DUE_DATE)values (1,1,'10-OCT-17');

insert into HOMEWORK (HW_ID,CHID,DUE_DATE)values (2,2,'15-NOV-17');
insert into HOMEWORK (HW_ID,CHID,DUE_DATE)values (3,3,'12-DEC-17');
insert into HOMEWORK (HW_ID,CHID,DUE_DATE)values (4,4,'20-FEB-17');
insert into HOMEWORK (HW_ID,CHID,DUE_DATE)values (5,5,'10-DEC-17');

insert into HOMEWORK_MARKS(HW_ID,U_ID,TOTAL_SCORE) values (1,2,100);
insert into HOMEWORK_MARKS(HW_ID,U_ID,TOTAL_SCORE) values (2,5,90);
insert into HOMEWORK_MARKS(HW_ID,U_ID,TOTAL_SCORE) values (3,1,80);
insert into HOMEWORK_MARKS(HW_ID,U_ID,TOTAL_SCORE) values (4,4,95);
insert into HOMEWORK_MARKS(HW_ID,U_ID,TOTAL_SCORE) values (5,3,85);
```

CREATE TABLE:

```
CREATE TABLE STUDENTUSERS
(
USER_ID         INT                      NOT NULL,
FNAME                    VARCHAR (15)                NOT NULL,
LNAME                    VARCHAR (15)                NOT NULL,
PASSWORD        VARCHAR (15)                NOT NULL,
BDATE                    DATE,
PHONE_NUM       CHAR (10),
EMAIL           VARCHAR (30)                NOT NULL,
ADDRESS         VARCHAR (30),
GENDER          CHAR,
PRIMARY KEY (USER_ID)
);

CREATE TABLE COURSE
(
COURSE_ID       INT                      NOT NULL,
CNAME                    VARCHAR (15)                NOT NULL,
C_DESCRIPTION VARCHAR (500),
START_DATE      DATE                     NOT NULL,
END_DATE        DATE                    NOT NULL,
STATUS CHAR NOT NULL,
PRIMARY KEY (COURSE_ID)
);

CREATE TABLE CHAPTER
(
CHAPTER_ID      INT                     NOT NULL,
CHNAME                  VARCHAR (15)                    NOT NULL,
CH_DESCRIPTION VARCHAR (500),
CID             INT                     NOT NULL,
PRIMARY KEY (CHAPTER_ID),
FOREIGN KEY (CID) REFERENCES COURSE (COURSE_ID) ON DELETE CASCADE
);

CREATE TABLE STUDENT_COURSE
(
S_USER_ID       INT                     NOT NULL,
S_COURSE_ID             INT                     NOT NULL,
PRIMARY KEY (S_USER_ID, S_COURSE_ID),
FOREIGN KEY (S_USER_ID) REFERENCES STUDENTUSERS (USER_ID) ON DELETE CASCADE,
FOREIGN KEY (S_COURSE_ID) REFERENCES COURSE (COURSE_ID) ON DELETE CASCADE
);

CREATE TABLE FEEDBACK
(
FEEDBACK_ID             INT                     NOT NULL,
U_ID            INT                     NOT NULL,
FDATE           DATE,
CID  INT NOT NULL,
FEEDBACK        VARCHAR (500)                   NOT NULL,
PRIMARY KEY (FEEDBACK_ID),
FOREIGN KEY (U_ID) REFERENCES STUDENTUSERS (USER_ID) ON DELETE CASCADE,
FOREIGN KEY (CID) REFERENCES COURSE (COURSE_ID) ON DELETE CASCADE
);

CREATE TABLE DISCUSSION_FORUM
(
DISS_ID                 INT                     NOT NULL,
CHID            INT                     NOT NULL,
EMAIL           VARCHAR (30)                    NOT NULL,
CDATE                   DATE                    NOT NULL,
PRIMARY KEY (DISS_ID),
FOREIGN KEY (CHID) REFERENCES CHAPTER (CHAPTER_ID) ON DELETE CASCADE
);

CREATE TABLE LEARNING_RESOURCES
(
LR_ID           INT                     NOT NULL,
CHID            INT                     NOT NULL,
URL             VARCHAR (200)                   NOT NULL,
PRIMARY KEY (LR_ID),
FOREIGN KEY (CHID) REFERENCES CHAPTER (CHAPTER_ID) ON DELETE CASCADE
);

CREATE TABLE INSTRUCTOR
(
INSTRUCTOR_ID           INT                     NOT NULL,
FNAME                   VARCHAR (15)                    NOT NULL,
LNAME          VARCHAR (15)        NOT NULL,
GENDER    CHAR,
BDATE          DATE,
PHONE_NUM CHAR (10),
EMAIL     VARCHAR (30)        NOT NULL,
ADDRESS   VARCHAR (30),
PRIMARY KEY (INSTRUCTOR_ID)
);

CREATE TABLE COURSE_INSTRUCTED
(
CID            INT            NOT NULL,
IID            INT            NOT NULL,
PRIMARY KEY (CID,IID),
FOREIGN KEY (CID) REFERENCES COURSE (COURSE_ID) ON DELETE CASCADE,
FOREIGN KEY (IID) REFERENCES INSTRUCTOR (INSTRUCTOR_ID) ON DELETE CASCADE
);

CREATE TABLE ALERT
(
ALERT_ID            INT            NOT NULL,
CHID                INT           NOT NULL,
ALERT_DESC         VARCHAR(500)        NOT NULL,
PRIMARY KEY (ALERT_ID),
FOREIGN KEY (CHID) REFERENCES CHAPTER (CHAPTER_ID) ON DELETE CASCADE
);

CREATE TABLE ALERT_TYPE
(
ALERT_ID            INT            NOT NULL,
ALERT_TYPE           VARCHAR(500)         NOT NULL,
PRIMARY KEY (ALERT_ID, ALERT_TYPE),
FOREIGN KEY (ALERT_ID) REFERENCES  ALERT (ALERT_ID) ON DELETE CASCADE
);

CREATE TABLE HOMEWORK
(
HW_ID             INT          NOT NULL,
CHID                 INT             NOT NULL,
DUE_DATE DATE,
PRIMARY KEY (HW_ID),
FOREIGN KEY (CHID) REFERENCES CHAPTER(CHAPTER_ID) ON DELETE CASCADE
);

CREATE TABLE HOMEWORK_MARKS
(
HW_ID       INT             NOT NULL,
U_ID            INT            NOT NULL,
TOTAL_SCORE FLOAT,
PRIMARY KEY (HW_ID, U_ID),
FOREIGN KEY (HW_ID) REFERENCES HOMEWORK (HW_ID),
FOREIGN KEY (U_ID) REFERENCES STUDENTUSERS (USER_ID) ON DELETE CASCADE
);

CREATE TABLE EXAM
(
EXAM_ID              INT       NOT NULL,
CHID                 INT       NOT NULL,
TERM INT NOT NULL,
PRIMARY KEY (EXAM_ID),
FOREIGN KEY (CHID) REFERENCES CHAPTER (CHAPTER_ID) ON DELETE CASCADE
);

CREATE TABLE QUESTIONS
(
QUESTION_ID             INT                 NOT NULL,
EX_ID             INT                 NOT NULL,
QUESTION          VARCHAR(500)        NOT NULL,
PRIMARY KEY (QUESTION_ID),
FOREIGN KEY (EX_ID) REFERENCES EXAM(EXAM_ID)
);

CREATE TABLE USER_ANSWERS
(
QID               INT                 NOT NULL,
U_ID              INT                 NOT NULL,
ANSWER            VARCHAR(100),
PRIMARY KEY (U_ID, QID),
FOREIGN KEY (U_ID) REFERENCES STUDENTUSERS (USER_ID) ON DELETE CASCADE,
FOREIGN KEY (QID) REFERENCES QUESTIONS (QUESTION_ID) ON DELETE CASCADE
);

CREATE TABLE USER_DISCUSSION
(
U_ID               INT                 NOT NULL,
DID               INT                 NOT NULL,
DTHREADS          VARCHAR(400),
PRIMARY KEY (U_ID, DID),
FOREIGN KEY (U_ID) REFERENCES STUDENTUSERS (USER_ID) ON DELETE CASCADE,
FOREIGN KEY (DID) REFERENCES DISCUSSION_FORUM (DISS_ID) ON DELETE CASCADE
```

```
);

CREATE TABLE USER_EXAM
(
U_ID                INT                 NOT NULL,
EX_ID               INT                 NOT NULL,
SCORE         INT                 NOT NULL,
PRIMARY KEY (U_ID, EX_ID),
FOREIGN KEY (U_ID) REFERENCES STUDENTUSERS (USER_ID) ON DELETE CASCADE,
FOREIGN KEY (EX_ID) REFERENCES EXAM (EXAM_ID) ON DELETE CASCADE
);
```

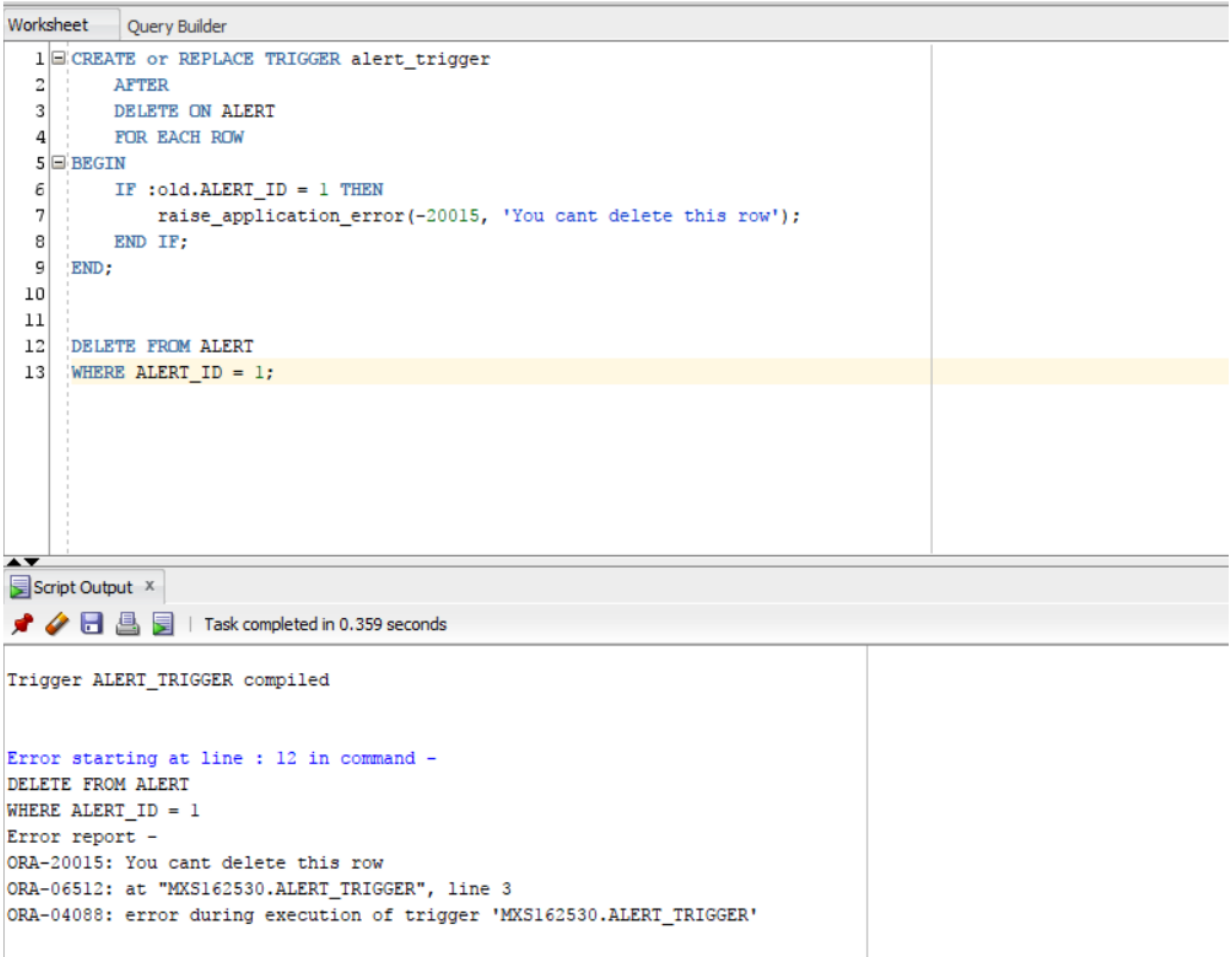
7 PL/SQL

TRIGGERS

1. Restriction to deleting trigger.

Function: The ALERT with ALERT_ID one contains the detailed structure of the COURSE and it cannot be DELETED.

Delete Trigger Code:



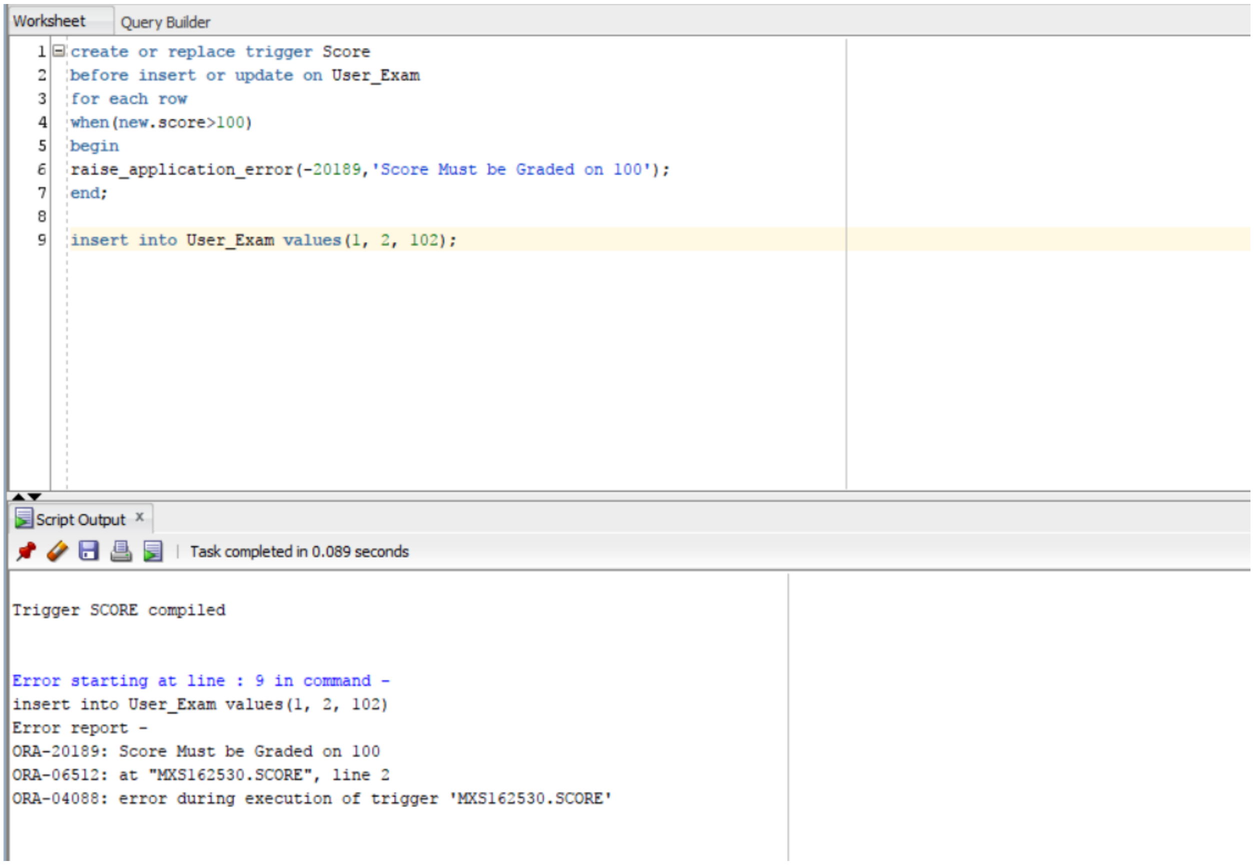
```
CREATE or REPLACE TRIGGER alert_trigger
    AFTER
    DELETE ON ALERT
    FOR EACH ROW
BEGIN
    IF :old.ALERT_ID = 1 THEN
        raise_application_error(-20015, 'You cant delete this row');
    END IF;
END;


DELETE FROM ALERT
WHERE ALERT_ID = 1;
```

```
Trigger ALERT_TRIGGER compiled


Error starting at line : 12 in command -
DELETE FROM ALERT
WHERE ALERT_ID = 1
Error report -
ORA-20015: You cant delete this row
ORA-06512: at "MXS162530.ALERT_TRIGGER", line 3
ORA-04088: error during execution of trigger 'MXS162530.ALERT_TRIGGER'
```

2. Restriction to updating a trigger

Function: A SCORES for EXAM cannot be more than 100

The trigger is added here to show that a maximum grade is 100.



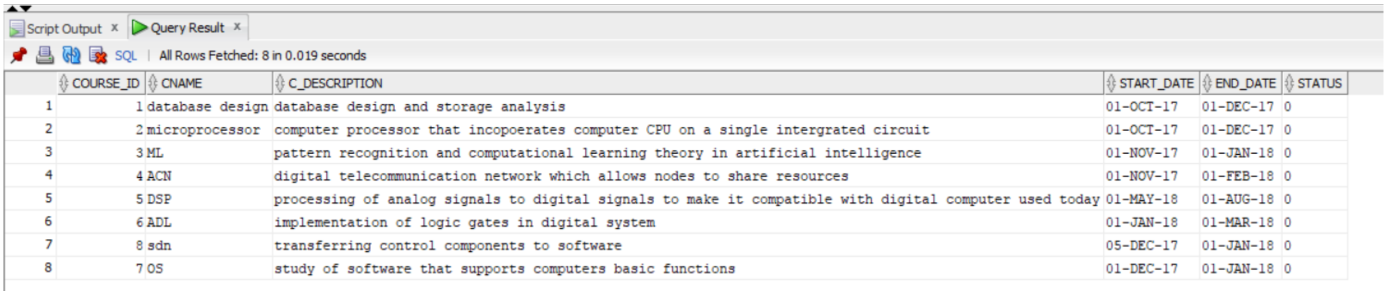
```
1 create or replace trigger Score
2 before insert or update on User_Exam
3 for each row
4 when(new.score>100)
5 begin
6 raise_application_error(-20189,'Score Must be Graded on 100');
7 end;
8
9 insert into User_Exam values(1, 2, 102);
```

```
Trigger SCORE compiled


Error starting at line : 9 in command -
insert into User_Exam values(1, 2, 102)
Error report -
ORA-20189: Score Must be Graded on 100
ORA-06512: at "MXS162530.SCORE", line 2
ORA-04088: error during execution of trigger 'MXS162530.SCORE'
```

PROCEDURES

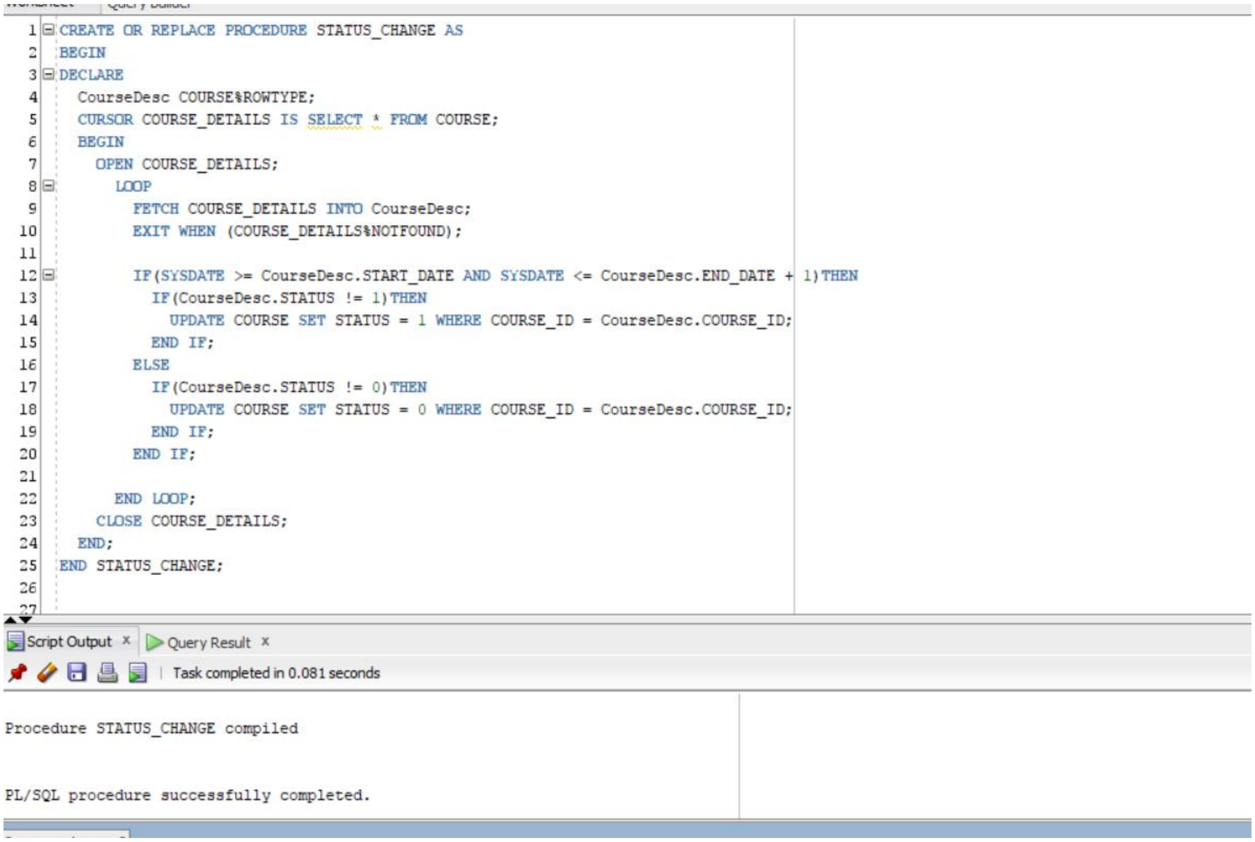
1. Function: PL/SQL stored procedure to change the status of a course to 1 if the current date is between the start & end date of the course and make it zero if it is not.

Course table before stored procedure



	COURSE_ID	CNAME	C_DESCRIPTION	START_DATE	END_DATE	STATUS
1	1	database design	database design and storage analysis	01-OCT-17	01-DEC-17	0
2	2	microprocessor	computer processor that incopoerates computer CPU on a single intergrated circuit	01-OCT-17	01-DEC-17	0
3	3	ML	pattern recognition and computational learning theory in artificial intelligence	01-NOV-17	01-JAN-18	0
4	4	ACN	digital telecommunication network which allows nodes to share resources	01-NOV-17	01-FEB-18	0
5	5	DSP	processing of analog signals to digital signals to make it compatible with digital computer used today	01-MAY-18	01-AUG-18	0
6	6	ADL	implementation of logic gates in digital system	01-JAN-18	01-MAR-18	0
7	8	sdn	transferring control components to software	05-DEC-17	01-JAN-18	0
8	7	OS	study of software that supports computers basic functions	01-DEC-17	01-JAN-18	0

Stored Procedure

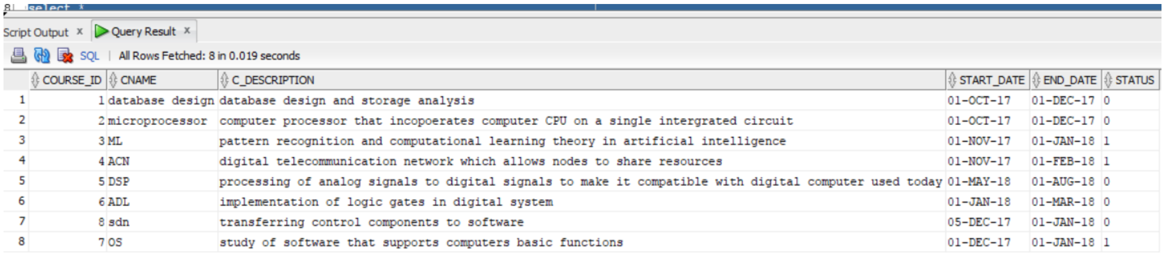


```
CREATE OR REPLACE PROCEDURE STATUS_CHANGE AS
BEGIN
DECLARE
  CourseDesc COURSE%ROWTYPE;
  CURSOR COURSE_DETAILS IS SELECT * FROM COURSE;
  BEGIN
    OPEN COURSE_DETAILS;
      LOOP
        FETCH COURSE_DETAILS INTO CourseDesc;
        EXIT WHEN (COURSE_DETAILS%NOTFOUND);

        IF(SYSDATE >= CourseDesc.START_DATE AND SYSDATE <= CourseDesc.END_DATE + 1)THEN
          IF(CourseDesc.STATUS != 1)THEN
            UPDATE COURSE SET STATUS = 1 WHERE COURSE_ID = CourseDesc.COURSE_ID;
          END IF;
        ELSE
          IF(CourseDesc.STATUS != 0)THEN
            UPDATE COURSE SET STATUS = 0 WHERE COURSE_ID = CourseDesc.COURSE_ID;
          END IF;
        END IF;

      END LOOP;
    CLOSE COURSE_DETAILS;
  END;
END STATUS_CHANGE;
```

Task completed in 0.081 seconds

```
Procedure STATUS_CHANGE compiled


PL/SQL procedure successfully completed.
```

Course Table after applying store procedure

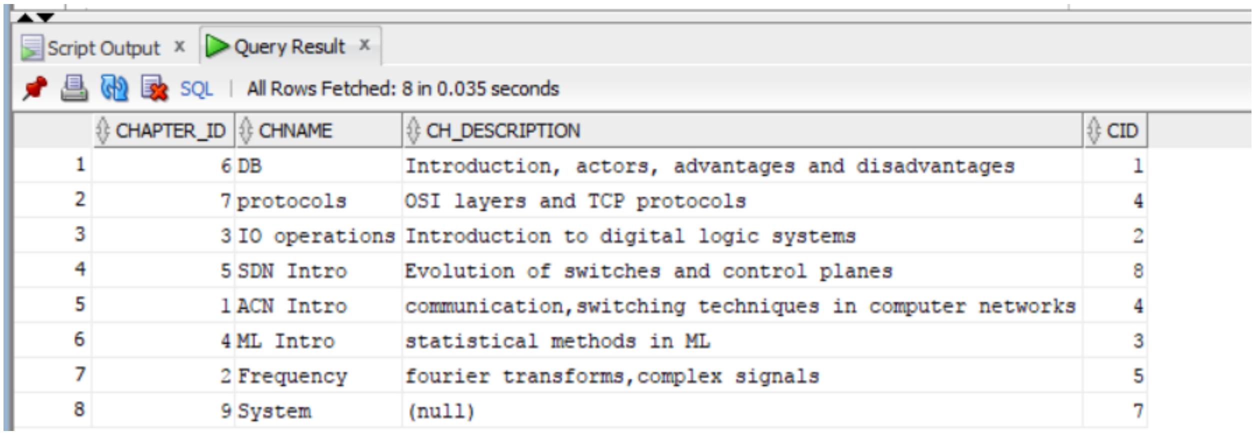


All Rows Fetched: 8 in 0.019 seconds

	COURSE_ID	CNAME	C_DESCRIPTION	START_DATE	END_DATE	STATUS
1	1	database design	database design and storage analysis	01-OCT-17	01-DEC-17	0
2	2	microprocessor	computer processor that incopoerates computer CPU on a single intergrated circuit	01-OCT-17	01-DEC-17	0
3	3	ML	pattern recognition and computational learning theory in artificial intelligence	01-NOV-17	01-JAN-18	1
4	4	ACN	digital telecommunication network which allows nodes to share resources	01-NOV-17	01-FEB-18	1
5	5	DSP	processing of analog signals to digital signals to make it compatible with digital computer used today	01-MAY-18	01-AUG-18	0
6	6	ADL	implementation of logic gates in digital system	01-JAN-18	01-MAR-18	0
7	8	sdn	transferring control components to software	05-DEC-17	01-JAN-18	0
8	7	OS	study of software that supports computers basic functions	01-DEC-17	01-JAN-18	1

2. Function: If CH_DESCRIPTION is “null” it should be shown as “Chapter description not provided”.

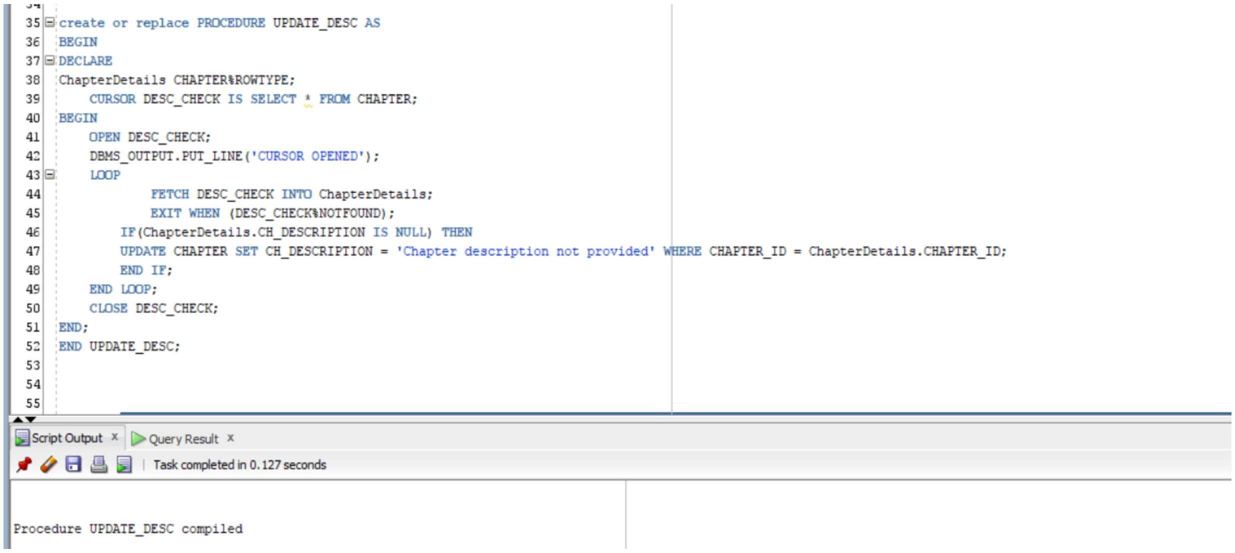
Chapter table before applying stored procedure



All Rows Fetched: 8 in 0.035 seconds

	CHAPTER_ID	CHNAME	CH_DESCRIPTION	CID
1	6	DB	Introduction, actors, advantages and disadvantages	1
2	7	protocols	OSI layers and TCP protocols	4
3	3	IO operations	Introduction to digital logic systems	2
4	5	SDN Intro	Evolution of switches and control planes	8
5	1	ACN Intro	communication,switching techniques in computer networks	4
6	4	ML Intro	statistical methods in ML	3
7	2	Frequency	fourier transforms,complex signals	5
8	9	System	(null)	7

Stored Procedure

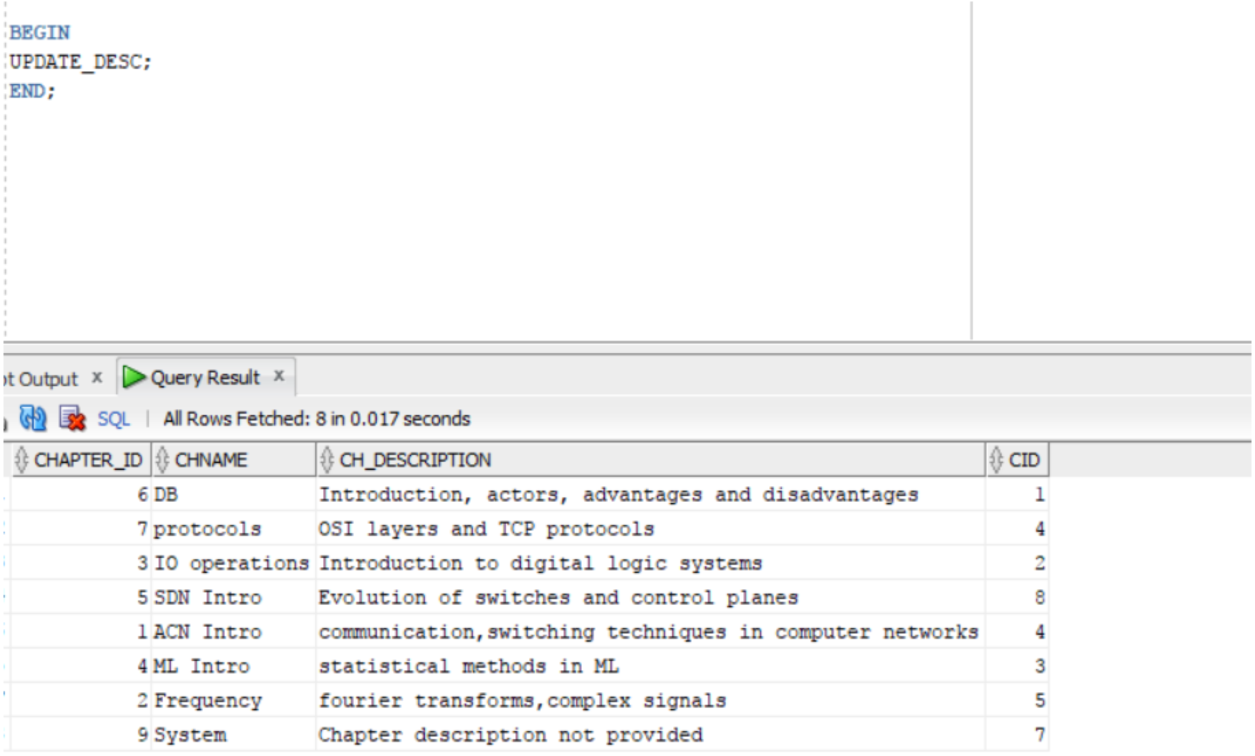


```
create or replace PROCEDURE UPDATE_DESC AS
BEGIN
DECLARE
ChapterDetails CHAPTER%ROWTYPE;
    CURSOR DESC_CHECK IS SELECT * FROM CHAPTER;
BEGIN
    OPEN DESC_CHECK;
    DBMS_OUTPUT.PUT_LINE('CURSOR OPENED');
    LOOP
            FETCH DESC_CHECK INTO ChapterDetails;
            EXIT WHEN (DESC_CHECK%NOTFOUND);
        IF(ChapterDetails.CH_DESCRIPTION IS NULL) THEN
        UPDATE CHAPTER SET CH_DESCRIPTION = 'Chapter description not provided' WHERE CHAPTER_ID = ChapterDetails.CHAPTER_ID;
        END IF;
    END LOOP;
    CLOSE DESC_CHECK;
END;
END UPDATE_DESC;
```

Task completed in 0.127 seconds

```
Procedure UPDATE_DESC compiled
```

Chapter Table after applying stored procedure



```
BEGIN
UPDATE_DESC;
END;
```

All Rows Fetched: 8 in 0.017 seconds

CHAPTER_ID	CHNAME	CH_DESCRIPTION	CID
6	DB	Introduction, actors, advantages and disadvantages	1
7	protocols	OSI layers and TCP protocols	4
3	IO operations	Introduction to digital logic systems	2
5	SDN Intro	Evolution of switches and control planes	8
1	ACN Intro	communication,switching techniques in computer networks	4
4	ML Intro	statistical methods in ML	3
2	Frequency	fourier transforms,complex signals	5
9	System	Chapter description not provided	7