CS/SE 6360. 005 DATABASE DESIGN

ONLINE EDUCATION SYSTEM

GUIDE : Prof. Nurcan Yuruk

TEAM MEMBERS:

Panchami Rudrakshi- pgr150030

Tushar Bhatia- txb150630

Aruksha Grover- axg151031

INDEX

INTRODUCTION

An Online Education system is a software application designed to take educational content and move it to the web. The term Online Education System is representative of a foundational technology whose sole purpose is to aggregate and reposit these materials on the web. The Online Education System at its core is what prompted technological advancements in eBooks, Learning Management and Learning Record Management. These systems enable learners to access content from anywhere with access to the internet. It supports teaching and learning using a computer web technology and bridges the gap between a teacher and a student in different ways.

Online education system   is essentially the computer and network enabled transfer of skills and knowledge. It refers to using electronic applications and processes to learn. Online education systems include Web-based learning, computer-based learning, virtual classrooms and digital collaboration. Content is delivered via the Internet, intranet/extranet, audio or video tape, satellite TV, and CD-ROM. It is used by the educational Institutions to enhance and support the class room teaching and offering courses to a larger population of learners across the Globe. It can be self-paced or instructor led and includes media in the form of text, image, animation, streaming video and audio.

Online education system has created new markets for teaching and learning material and equipment, attracting the attention of academic institutions as well as companies supplying them in different sectors – computer manufacturers, software producers, publishing houses and special training providers. It has also led to the reorientation of government policy, in particular, towards encouraging the spread of online education techniques and developing the skills and know-how required for their use.

Objectives

Online education system represents an innovative shift in the field of learning, providing rapid access to specific knowledge and information. It offers online instruction that can be delivered anytime and anywhere through a wide range of electronic learning solutions such as Web-based courseware, online discussion groups, live virtual classes, video and audio streaming, Web chat, online simulations, and virtual mentoring.

Online education system enables organizations to transcend distance and other organizational gaps by providing a cohesive virtual learning environment. Companies must educate and train vendors, employees, partners, and clients to stay competitive, and the online education systems can provide such just-in-time training in a cost-effective way.

Benefits of Online Education System

- Convenient
- Private learning
- Self paced
- Flexibility
- Easy to understand and more engaging
- Repeatable
- Easier to monitor progress
- Consistent

DATA REQUIREMENTS FOR THE SYSTEM

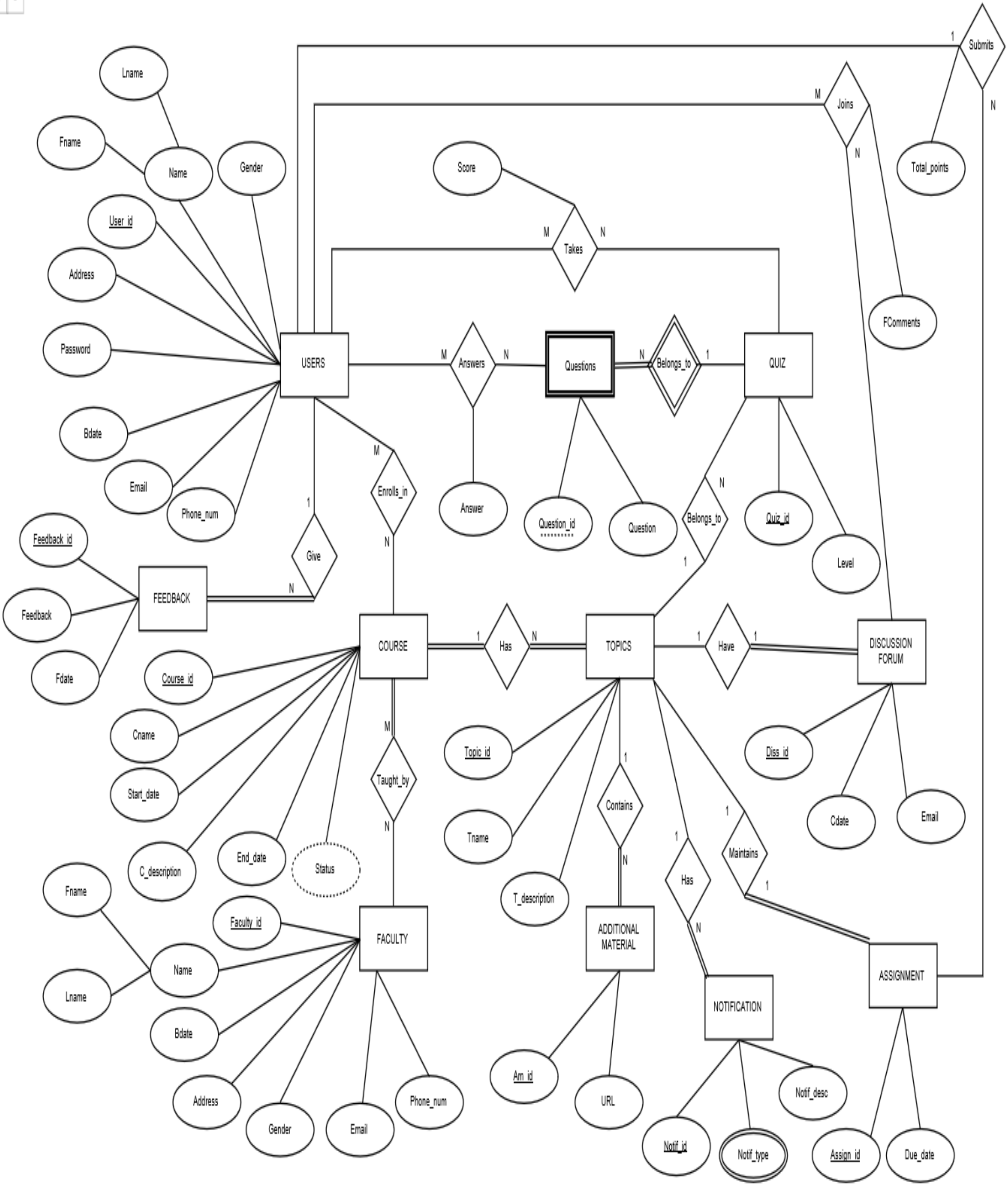
- The system consists of two types of users: Faculty & Subscribers. Details of each like USER ID, NAME, BDATE, PHONE NUMBER, EMAIL, ADDRESS and GENDER are maintained by the system. A user has to register himself to access the online education system. He is authenticated by his login credentials (PASSWORD) that he creates during registration.
- There are multiple COURSES provided by the system. Each course is identified by a unique ID and NAME. It has a DESCRIPTION to help the subscribers get an overview of the course and its syllabus. Each course shows it's START DATE and END DATE based on which its STATUS is determined. A course can be taken by any number of users at a time.
- Each COURSE has multiple TOPICS identified by a unique ID and has attributes as NAME and DESCRIPTION. A user can subscribe to any number of courses and the details are stored as STUDENT COURSE.
- Courses can be taught by different FACULTIES who are assigned ID for identification. Further, each faculty has the flexibility to handle multiple courses. Details about course taught by each is maintained as TAUGHT_BY.
- For effective teaching, STUDY MATERIALS are posted for each topic timely with an URL. Study materials can be accessed based on its ID and TOPIC to which it belongs.
- An ASSIGNMENT is also given for every topic to continuously evaluate the subscriber. Each has a DUE DATE mentioned before which it has to be submitted. After evaluation the SCORES are posted by the system for each subscriber (UID) and stored as ASSIGNMENT MARKS.
- Also, a DISCUSSION FORUM is maintained for each topic separately by the system to help the user better understand the topics. Each forum has a unique ID, DATE on which it takes place, belonging TOPIC and EMAIL of the participant. A subscriber can participate in any number of discussions i.e. USER DISCUSSIONS at a particular time and give COMMENTS.
- During the course each subscriber can give FEEDBACK about the course and staff to improve the system. Feedbacks are identified by their ID, DATE on which it is posted, person posting it(USER ID) and the FEEDBACK.

- Timely NOTIFICATIONS are posted by the system for each topic and its description is given. These notifications can be about exams, assignments, additional material, change of course timings etc and are differentiated by NOTIFICATION TYPE.
- QUIZZES are conducted for each topic to evaluate the performance of the subscriber. Each quiz has its unique ID and shows the topic to which it belongs. The subscriber has the choice to take the quiz at different LEVELS, but he has to complete all the quizzes before course end date.
- Each QUIZ consists of multiple QUESTIONS identified by a unique QUESTION ID. The subscriber's ANSWERS are stored by the system for each QUESTION. Based on his/her performance, SCORE for each QUIZ is maintained by the system as USER QUIZ details and notified to the user.

- QUIZZES are conducted for each topic to evaluate the performance of the subscriber. Each quiz has its unique ID and shows the topic to which it belongs. The subscriber has the choice to take the quiz at different LEVELS, but he has to complete all the quizzes before course end date.

- Each QUIZ consists of multiple QUESTIONS identified by a unique QUESTION ID and the quiz to which it belongs. The subscriber's ANSWERS are stored by the system for each QUESTION. Based on his/her performance, SCORE for each QUIZ is maintained by the system as USER QUIZ details and notified to the user.

ER DIAGRAM


Submits
1
N
Joins
M
N
Lname
Fname
Name
Gender
Total_points
Score
M
Takes
N
User_id
Address
FComments
Password
USERS
M
Answers
N
Questions
N
Belongs_to
1
QUIZ
Bdate
Email
Phone_num
M
Enrolls_in
N
Answer
Question_id
Question
N
Belongs_to
1
Quiz_id
Level
1
Give
N
Feedback_id
Feedback
FEEDBACK
Fdate
COURSE
1
Has
N
TOPICS
1
Have
1
DISCUSSION FORUM
Course_id
Cname
Start_date
M
Taught_by
N
Topic_id
Tname
1
Contains
N
Diss_id
Cdate
Email
C_description
End_date
Status
1
Has
N
1
Maintains
1
T_description
Fname
Faculty_id
FACULTY
Name
Lname
Bdate
ADDITIONAL MATERIAL
ASSIGNMENT
NOTIFICATION
Address
Gender
Email
Phone_num
Am_id
URL
Notif_desc
Notif_id
Notif_type
Assign_id
Due_date

ASSUMPTIONS

- A user can only register in a COURSE before the start date of the course.
- Status of the COURSE will only be 1 when the present date is between the course start date & end date.
- If there exists a COURSE then it has to be taught by some faculty member.
- If there exists some ADDITIONAL_MATERIAL then it must belong to some topic.
- If there exists some NOTIFICATION then it must belong to some topic.
- If there exists some ASSIGNMENT then it must belong to some topic.
- If there exists some DISCUSSION_FORUM then it must belong to some topic.
- One USER can submit multiple assignments belonging to different topics but he cannot submit the same assignment multiple times.
- For one TOPIC user can take QUIZ only once.
- One QUESTION belong to only one QUIZ at a time.

The system includes the following relationships:

1) One-to-one binary relationships.

- Each topic has an assignment. One assign8ment belongs to only one topic.
- Each topic has a discussion forum. One discussion forum belongs to only one topic.

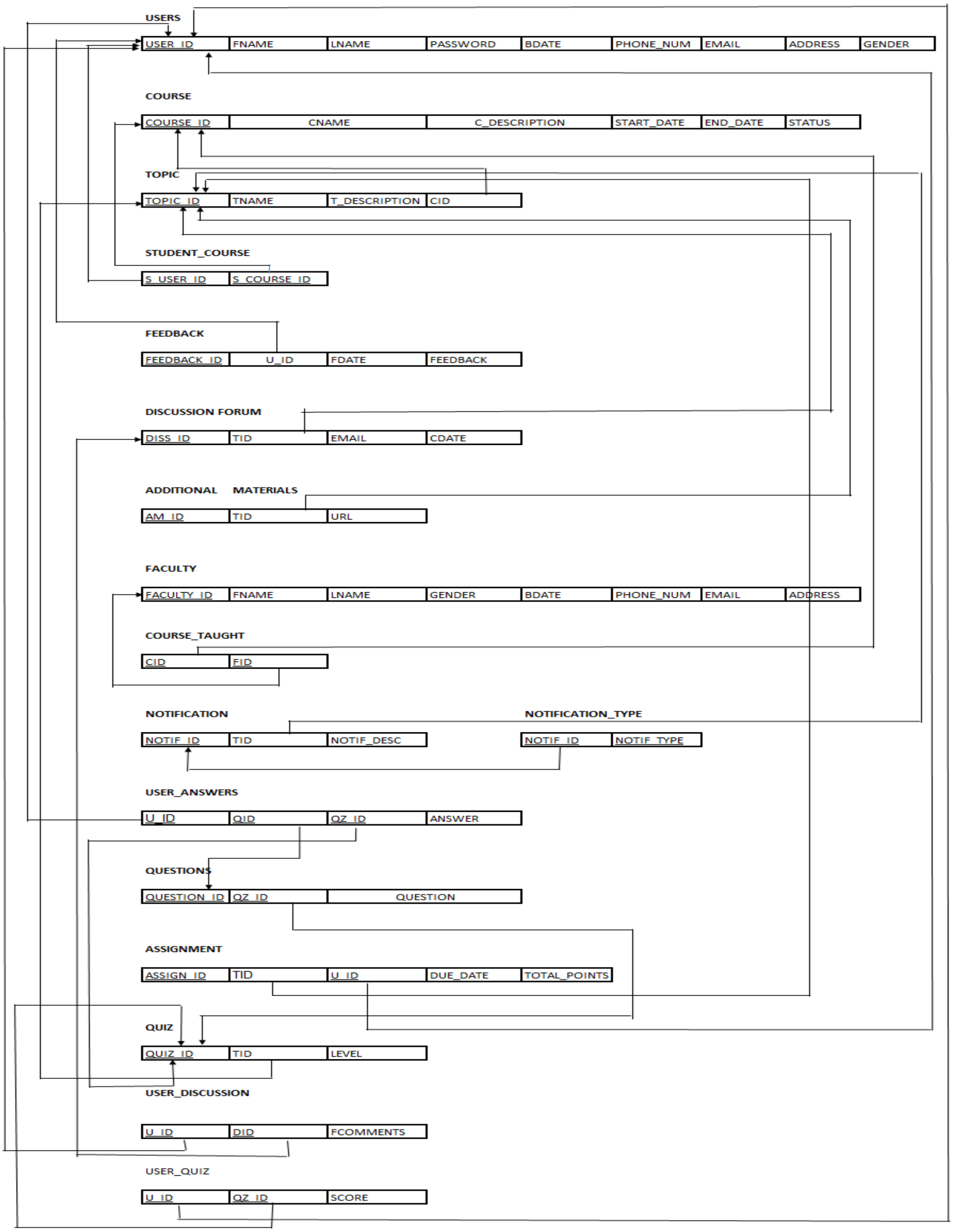
2) One-to-many binary relationships.

- Each quiz has multiple questions. A question can belong only one quiz.
- Each quiz belongs to a single topic. There can be multiple quizzes conducted for a particular topic.
- Each course has multiple topics. A topic can belong to only one course.
- There can be many additional materials in support of a particular topic. An additional material can belong to a single topic.
- There can be many notifications posted for a particular topic. A notification belongs to a single topic.
- Each user can submit multiple assignments. An assignment is submitted by only one user.
- A user can give multiple feedbacks. A feedback is given by a single user.

c. **Many-to-many binary relationships.**

- A user can take multiple courses. A particular course can be taken by multiple users.
- A course can be taught by any no. of faculties. A particular faculty can teach multiple courses.
- Each user answers multiple questions. A question is answered by multiple users.
- A user can take multiple quizzes. Each quiz can be taken by multiple users.
- Each user can participate in multiple discussion forums at the same time. A discussion forum has multiple users.

MAPPING OF ER DIAGRAM INTO RELATIONAL SCHEMA



USERS

USER_ID	FNAME	LNAME	PASSWORD	BDATE	PHONE_NUM	EMAIL	ADDRESS	GENDER

COURSE

COURSE_ID	CNAME	C_DESCRIPTION	START_DATE	END_DATE	STATUS

TOPIC

TOPIC_ID	TNAME	T_DESCRIPTION	CID

STUDENT_COURSE

S_USER_ID	S_COURSE_ID

FEEDBACK

FEEDBACK_ID	U_ID	FDATE	FEEDBACK

DISCUSSION FORUM

DISS_ID	TID	EMAIL	CDATE

ADDITIONAL MATERIALS

AM_ID	TID	URL

FACULTY

FACULTY_ID	FNAME	LNAME	GENDER	BDATE	PHONE_NUM	EMAIL	ADDRESS

COURSE_TAUGHT

CID	FID

NOTIFICATION

NOTIF_ID	TID	NOTIF_DESC

NOTIFICATION_TYPE

NOTIF_ID	NOTIF_TYPE

USER_ANSWERS

U_ID	QID	QZ_ID	ANSWER

QUESTIONS

QUESTION_ID	QZ_ID	QUESTION

ASSIGNMENT

ASSIGN_ID	TID	U_ID	DUE_DATE	TOTAL_POINTS

QUIZ

QUIZ_ID	TID	LEVEL

USER_DISCUSSION

U_ID	DID	FCOMMENTS

USER_QUIZ

U_ID	QZ_ID	SCORE

FUNCTIONAL DEPENDENCIES FOR THE SYSTEM

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

COURSE
COURSE_ID →CNAME, C_DESCRIPTION, START_DATE, END_DATE

TOPIC
TOPIC_ID → TNAME, T_DESCRIPTION, CID

FEEDBACK
FEEDBACK_ID → U_ID, FDATE, FEEDBACK

DISCUSSION_FORUM
DISS_ID → TID, EMAIL, CDATE

ADDITIONAL_MATERIAL
AM_ID → TID, URL

FACULTY
FACULTY_ID →FNAME, LNAME, GENDER, BDATE, PHONE_NUM, EMAIL, ADDRESS

NOTIFICATION
NOTIF_ID → TID, NOTIF_DESC

USER_ANSWERS
U_ID, QID,QZ_ID→ ANSWER

QUESTIONS
QUESTION_ID, QZ_ID → QUESTION

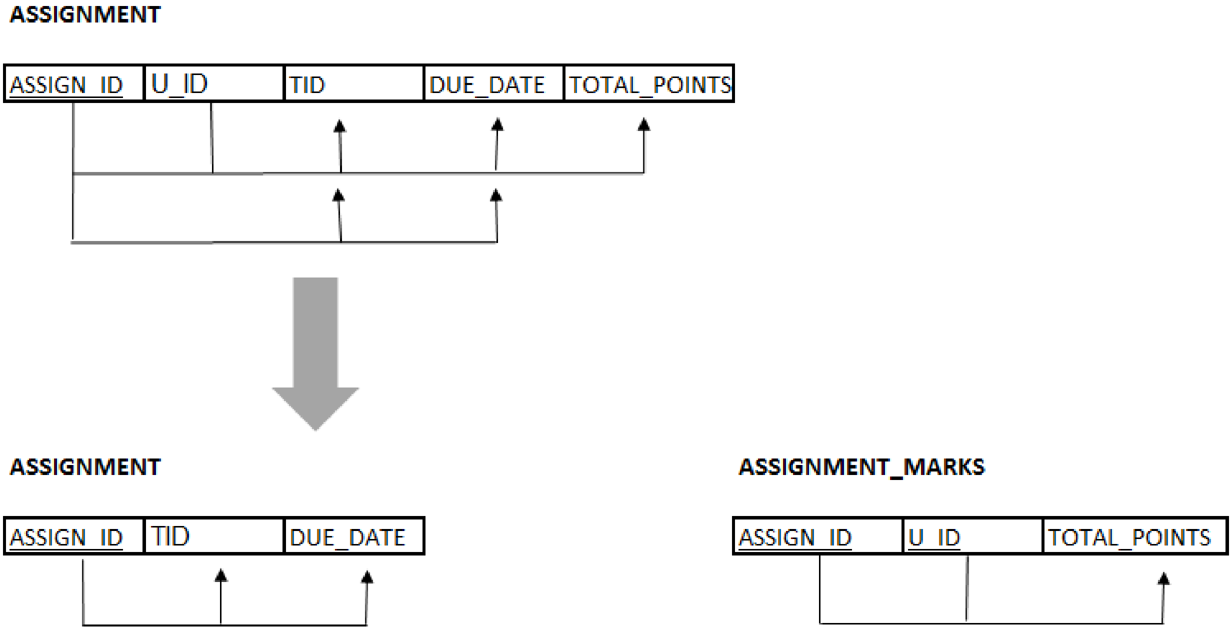
ASSIGNMENT
ASSIGN_ID, U_ID→ TID, DUE_DATE, TOTAL_POINTS
ASSIGN_ID→TID, DUE_DATE

QUIZ
QUIZ_ID → TID,LEVEL

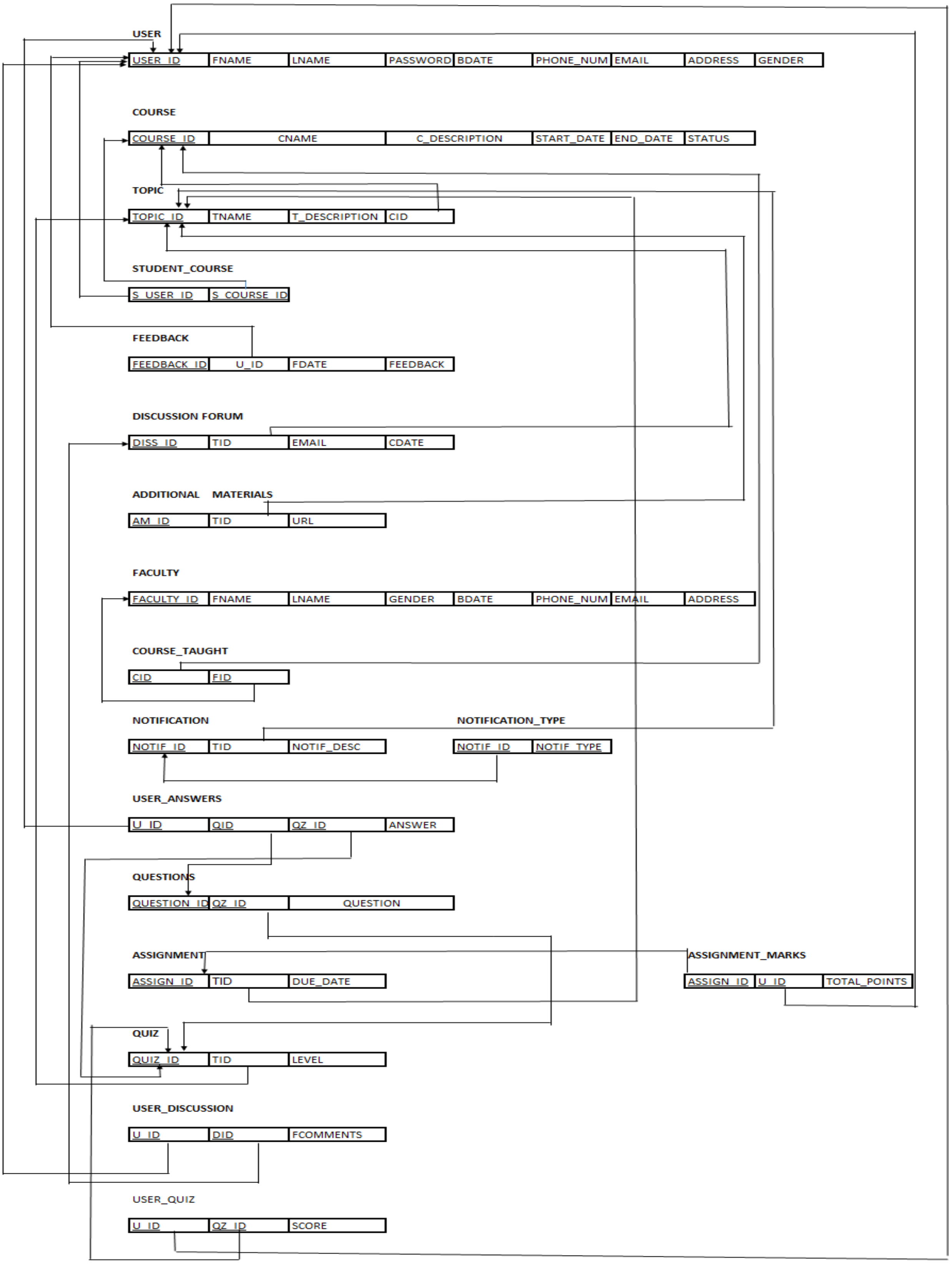
USER_DISCUSSION
U_ID, DID → FCOMMENTS

USER_QUIZ
U_ID, QZ_ID → SCORE

NORMALIZATION



FINAL RELATIONAL SCHEMA AFTER NORMALIZATION


USER
USER_ID | FNAME | LNAME | PASSWORD | BDATE | PHONE_NUM | EMAIL | ADDRESS | GENDER
COURSE
COURSE_ID | CNAME | C_DESCRIPTION | START_DATE | END_DATE | STATUS
TOPIC
TOPIC_ID | TNAME | T_DESCRIPTION | CID
STUDENT_COURSE
S_USER_ID | S_COURSE_ID
FEEDBACK
FEEDBACK_ID | U_ID | FDATE | FEEDBACK
DISCUSSION FORUM
DISS_ID | TID | EMAIL | CDATE
ADDITIONAL MATERIALS
AM_ID | TID | URL
FACULTY
FACULTY_ID | FNAME | LNAME | GENDER | BDATE | PHONE_NUM | EMAIL | ADDRESS
COURSE_TAUGHT
CID | FID
NOTIFICATION
NOTIF_ID | TID | NOTIF_DESC
NOTIFICATION_TYPE
NOTIF_ID | NOTIF_TYPE
USER_ANSWERS
U_ID | QID | QZ_ID | ANSWER
QUESTIONS
QUESTION_ID | QZ_ID | QUESTION
ASSIGNMENT
ASSIGN_ID | TID | DUE_DATE
ASSIGNMENT_MARKS
ASSIGN_ID | U_ID | TOTAL_POINTS
QUIZ
QUIZ_ID | TID | LEVEL
USER_DISCUSSION
U_ID | DID | FCOMMENTS
USER_QUIZ
U_ID | QZ_ID | SCORE

SQL STATEMENTS

DROP STATEMENTS

```
DROP TABLE USER_QUIZ;
DROP TABLE USER_DISCUSSION;
DROP TABLE USER_ANSWERS;
DROP TABLE QUESTIONS;
DROP TABLE QUIZ;
DROP TABLE ASSIGNMENT_MARKS;
DROP TABLE ASSIGNMENT;
DROP TABLE NOTIFICATION_TYPE;
DROP TABLE NOTIFICATION;
DROP TABLE COURSE_TAUGHT;
DROP TABLE FACULTY;
DROP TABLE ADDITIONAL_MATERIAL;
DROP TABLE DISCUSSION_FORUM;
DROP TABLE FEEDBACK;
DROP TABLE STUDENT_COURSE;
DROP TABLE TOPIC;
DROP TABLE COURSE;
DROP TABLE USERS;
```

SQL CREATE STATEMENTS

```
CREATE TABLE USERS
(
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 TOPIC
(
TOPIC_ID  INT            NOT NULL,
TNAME          VARCHAR (15)        NOT NULL,
T_DESCRIPTION VARCHAR (500),
CID       INT            NOT NULL,
PRIMARY KEY (TOPIC_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 USERS (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,
FEEDBACK  VARCHAR (500)       NOT NULL,
PRIMARY KEY (FEEDBACK_ID),
FOREIGN KEY (U_ID) REFERENCES USERS (USER_ID) ON DELETE CASCADE
);

CREATE TABLE DISCUSSION_FORUM
(
DISS_ID   INT            NOT NULL,
TID       INT            NOT NULL,
EMAIL     VARCHAR (30)        NOT NULL,
CDATE     DATE           NOT NULL,
PRIMARY KEY (DISS_ID),
FOREIGN KEY (TID) REFERENCES TOPIC (TOPIC_ID) ON DELETE CASCADE
);

CREATE TABLE ADDITIONAL_MATERIAL
```

```
(
AM_ID     INT            NOT NULL,
TID       INT            NOT NULL,
URL       VARCHAR (200)       NOT NULL,
PRIMARY KEY (AM_ID),
FOREIGN KEY (TID) REFERENCES TOPIC (TOPIC_ID) ON DELETE CASCADE
);

CREATE TABLE FACULTY
(
FACULTY_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 (FACULTY_ID)
);

CREATE TABLE COURSE_TAUGHT
(
CID            INT             NOT NULL,
FID            INT             NOT NULL,
PRIMARY KEY (CID,FID),
FOREIGN KEY (CID) REFERENCES COURSE (COURSE_ID) ON DELETE CASCADE,
FOREIGN KEY (FID) REFERENCES FACULTY (FACULTY_ID) ON DELETE CASCADE
);

CREATE TABLE NOTIFICATION
(
NOTIF_ID             INT            NOT NULL,
TID                  INT                NOT NULL,
NOTIF_DESC          VARCHAR(500)         NOT NULL,
PRIMARY KEY (NOTIF_ID),
FOREIGN KEY (TID) REFERENCES TOPIC (TOPIC_ID) ON DELETE CASCADE
);

CREATE TABLE NOTIFICATION_TYPE
(
NOTIF_ID             INT            NOT NULL,
NOTIF_TYPE          VARCHAR(500)         NOT NULL,
PRIMARY KEY (NOTIF_ID, NOTIF_TYPE),
FOREIGN KEY (NOTIF_ID) REFERENCES  NOTIFICATION (NOTIF_ID) ON
DELETE CASCADE
);

CREATE TABLE ASSIGNMENT
(
ASSIGN_ID            INT                NOT NULL,
TID                  INT                NOT NULL,
DUE_DATE DATE,
PRIMARY KEY (ASSIGN_ID),
FOREIGN KEY (TID) REFERENCES TOPIC (TOPIC_ID) ON DELETE CASCADE
);

CREATE TABLE ASSIGNMENT_MARKS
(
ASSIGN_ID      INT             NOT NULL,
U_ID                 INT            NOT NULL,
TOTAL_POINS FLOAT,
PRIMARY KEY (ASSIGN_ID, U_ID),
FOREIGN KEY (ASSIGN_ID) REFERENCES ASSIGNMENT (ASSIGN_ID),
FOREIGN KEY (U_ID) REFERENCES USERS (USER_ID) ON DELETE CASCADE
);

CREATE TABLE QUIZ
```

```
(
QUIZ_ID              INT           NOT NULL,
TID                  INT           NOT NULL,
LVLS INT NOT NULL,
PRIMARY KEY (QUIZ_ID),
FOREIGN KEY (TID) REFERENCES TOPIC (TOPIC_ID) ON DELETE CASCADE
);

CREATE TABLE QUESTIONS
(
QUESTION_ID         INT            NOT NULL,
QZ_ID               INT            NOT NULL,
QUESTION       VARCHAR(500)   NOT NULL,
PRIMARY KEY (QUESTION_ID,QZ_ID),
FOREIGN KEY (QZ_ID) REFERENCES QUIZ (QUIZ_ID)
);

CREATE TABLE USER_ANSWERS
(
QID            INT            NOT NULL,
U_ID           INT            NOT NULL,
ANSWER         VARCHAR(100),
QZ_ID          INT            NOT NULL,
PRIMARY KEY (U_ID,QID,QZ_ID),
FOREIGN KEY (U_ID) REFERENCES USERS (USER_ID) ON DELETE CASCADE,
FOREIGN KEY (QID,QZ_ID) REFERENCES QUESTIONS (QUESTION_ID,QZ_ID) ON
DELETE CASCADE
);
CREATE TABLE USER_DISCUSSION
(
U_ID            INT            NOT NULL,
DID            INT            NOT NULL,
FCOMMENTS           VARCHAR(400),
PRIMARY KEY (U_ID, DID),
FOREIGN KEY (U_ID) REFERENCES USERS (USER_ID) ON DELETE CASCADE,
FOREIGN KEY (DID) REFERENCES DISCUSSION_FORUM (DISS_ID) ON DELETE
CASCADE
);

CREATE TABLE USER_QUIZ
(
U_ID           INT             NOT NULL,
QZ_ID          INT            NOT NULL,
SCORE          INT            NOT NULL,
PRIMARY KEY (U_ID, QZ_ID),
FOREIGN KEY (U_ID) REFERENCES USERS (USER_ID) ON DELETE CASCADE,
FOREIGN KEY (QZ_ID) REFERENCES QUIZ (QUIZ_ID) ON DELETE CASCADE
);
```

INSERT STATEMENTS

```
insert into USERS (USER_ID, FNAME, LNAME, PASSWORD, BDATE,
PHONE_NUM, EMAIL, ADDRESS, GENDER) values
(1,'tushar','bhatia','123','10-JAN-
93','8789876576','txb@gmail.com','7824 Mcclum blvd','m');
insert into USERS (USER_ID, FNAME, LNAME, PASSWORD, BDATE,
PHONE_NUM, EMAIL, ADDRESS, GENDER) values
(2,'aruksha','singh','124','12-FEB-
93','8989876576','axg@gmail.com','7774 Mcclum blvd','f');
insert into USERS (USER_ID, FNAME, LNAME, PASSWORD, BDATE,
PHONE_NUM, EMAIL, ADDRESS, GENDER) values
(3,'panchami','rudrakshi','125','16-MAR-
92','8787765576','pgr@gmail.com','7825 Mcclum blvd','f');
insert into USERS (USER_ID, FNAME, LNAME, PASSWORD, BDATE,
PHONE_NUM, EMAIL, ADDRESS, GENDER) values
(4,'jai','arora','111','12-FEB-
90','8988909576','jai@gmail.com','7904 Mcclum blvd','m');
```

```
insert into USERS (USER_ID, FNAME, LNAME, PASSWORD, BDATE,
PHONE_NUM, EMAIL, ADDRESS, GENDER) values
(5,'raghu','NR','100','06-FEB-
91','8000909576','raghu@gmail.com','7000Mcclum blvd','m');
insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (1,'statistics','analysis of statistical
methods for data scientist','01-JAN-15','01-MAR-15',0);
insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (2,'database design','database design and
storage analysis','01-OCT-15','01-DEC-15',0);
insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (3,'sdn','transferring control components
to software','01-NOV-15','01-JAN-16',0);
insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (4,'ML','pattern recognition and
computational learning theory in artificial intelligence','01-NOV-
15','01-FEB-16',0);
insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (5,'bigdata','sets so large or complex that
traditional data processing applications','01-MAY-16','01-AUG-
16',0);
insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (6,'statistics','analysis of statistical
methods for data scientist','01-JAN-16','01-MAR-16',0);
insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (7,'database design','database design and
storage analysis','01-DEC-15','01-JAN-16',0);
insert into COURSE (COURSE_ID, CNAME, C_DESCRIPTION, START_DATE,
END_DATE,STATUS) values (8,'sdn','transferring control components
to software','05-DEC-15','01-JAN-16',0);

insert into TOPIC (TOPIC_ID,TNAME, T_DESCRIPTION,CID) values
(3,'probability','Events and their probabilities',1);
```

```
insert into TOPIC (TOPIC_ID, TNAME, T_DESCRIPTION,CID) values
(4,'Distributions','',1);
insert into TOPIC (TOPIC_ID, TNAME, T_DESCRIPTION,CID) values
(1,'DB','Introduction, actors, advantages and disadvantages',2);
insert into TOPIC (TOPIC_ID,TNAME, T_DESCRIPTION,CID) values
(2,'Architecture','System concepts,data models,schemas',2);
insert into TOPIC (TOPIC_ID, TNAME, T_DESCRIPTION,CID) values
(5,'SDN Intro','',3);
insert into TOPIC (TOPIC_ID, TNAME, T_DESCRIPTION,CID) values
(6,'ML Intro','statistical methods in ML',4);
insert into TOPIC (TOPIC_ID, TNAME, T_DESCRIPTION,CID) values (7
,'bigdata Intro','data classification and analysis',5);

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 STUDENT_COURSE (S_USER_ID, S_COURSE_ID) values (1,6);
insert into STUDENT_COURSE (S_USER_ID, S_COURSE_ID) values (2,6);
insert into STUDENT_COURSE (S_USER_ID, S_COURSE_ID) values (3,6);

insert into FEEDBACK (FEEDBACK_ID,U_ID, FDATE, FEEDBACK) values
(1,1,'15-JAN-15','very useful study material');
insert into FEEDBACK (FEEDBACK_ID,U_ID, FDATE, FEEDBACK) values
(2,2,'20-DEC-15','lectures need to be more detailed');
insert into FEEDBACK (FEEDBACK_ID,U_ID, FDATE, FEEDBACK) values
(3,3,'09-NOV-15','good assignments');
```

```
insert into FEEDBACK (FEEDBACK_ID,U_ID, FDATE, FEEDBACK) values
(4,4,'22-JAN-15','very tough quizzes');
insert into FEEDBACK (FEEDBACK_ID,U_ID, FDATE, FEEDBACK) values
(5,5,'11-OCT-15','very poor video quality');

insert into DISCUSSION_FORUM(DISS_ID, TID,EMAIL,CDATE) values
(1,1,'txb@gmail.com','01-NOV-15');
insert into DISCUSSION_FORUM(DISS_ID, TID, EMAIL,CDATE) values
(2,2,'too@gmail.com','15-NOV-15');
insert into DISCUSSION_FORUM(DISS_ID, TID,EMAIL, CDATE) values
(3,3,'asd@gmail.com','01-FEB-15');
insert into DISCUSSION_FORUM(DISS_ID, TID,EMAIL, CDATE) values
(4,4,'rtd@gmail.com','29-JAN-15');
insert into DISCUSSION_FORUM(DISS_ID, TID,EMAIL, CDATE) values
(5,5,'ryui@gmail.com','11-NOV-15');
insert into DISCUSSION_FORUM(DISS_ID, TID,EMAIL, CDATE) values
(6,6,'ropd@gmail.com','01-DEC-15');

insert into USER_DISCUSSION(U_ID,DID ,FCOMMENTS) values (2,1,'What
is sql');
insert into USER_DISCUSSION(U_ID,DID ,FCOMMENTS) values (3,2, 'What
is nosql');
insert into USER_DISCUSSION(U_ID,DID ,FCOMMENTS) values (4,3, 'What
is statistics');
insert into USER_DISCUSSION(U_ID,DID ,FCOMMENTS) values (2,4, 'What
is sdn');
insert into USER_DISCUSSION(U_ID,DID ,FCOMMENTS) values (3,5, 'What
is ml');
insert into USER_DISCUSSION(U_ID,DID ,FCOMMENTS) values (1,6, 'What
is bigdata');

insert into ADDITIONAL_MATERIAL (AM_ID, TID, URL) values
(1,1,'https://drive.google.com/file/d/0B_CXBnxbGzHYR1RuXzM5dlhsbTQ/
view?ts=565a2752');
```

```
insert into ADDITIONAL_MATERIAL (AM_ID, TID, URL) values (2,1,'
https://www.google.come');
insert into ADDITIONAL_MATERIAL (AM_ID, TID, URL) values (4,2,'
https://en.wikipedia.org/wiki/Statistics');
insert into ADDITIONAL_MATERIAL (AM_ID, TID, URL) values (5,3,'
https://en.wikipedia.org/wiki/Stat');
insert into ADDITIONAL_MATERIAL (AM_ID, TID, URL) values (6,4,'
https://en.wikipedia.org/wiki/Software-defined_networking');
insert into ADDITIONAL_MATERIAL (AM_ID, TID, URL) values (7,5,'
https://en.wikipedia.org/wiki/ml');
insert into ADDITIONAL_MATERIAL (AM_ID, TID, URL) values (8,6,'
https://en.wikipedia.org/wiki/Software-
defined_networking/bigdata');

insert into
FACULTY(FACULTY_ID,FNAME,LNAME,GENDER,BDATE,PHONE_NUM,EMAIL,ADDRESS
)values (1,'sammy','johnson','m','10-JAN-
1963','8780076576','sammy@gmail.com','794 Mcclum blvd');
insert into
FACULTY(FACULTY_ID,FNAME,LNAME,GENDER,BDATE,PHONE_NUM,EMAIL,ADDRESS
)values (2,'joy','culver','m','10-JAN-
1973','8009876576','joy@gmail.com','7829 Mcclum blvd');
insert into
FACULTY(FACULTY_ID,FNAME,LNAME,GENDER,BDATE,PHONE_NUM,EMAIL,ADDRESS
)values (3,'pallavi','bhat','f','10-JAN-
1979','8789876076','pallavi@gmail.com','7894 Mcclum blvd');
insert into
FACULTY(FACULTY_ID,FNAME,LNAME,GENDER,BDATE,PHONE_NUM,EMAIL,ADDRESS
)values (4,'kavya','raman','f','10-JAN-
1980','8709876576','kavya@gmail.com','9824 Mcclum blvd');
insert into
FACULTY(FACULTY_ID,FNAME,LNAME,GENDER,BDATE,PHONE_NUM,EMAIL,ADDRESS
)values (5,'lahari','ganesh','f','10-JAN-
1982','8789870576','lahari@gmail.com','7884 Mcclum blvd');
```

```
insert into
FACULTY(FACULTY_ID,FNAME,LNAME,GENDER,BDATE,PHONE_NUM,EMAIL,ADDRESS
)values (6,'ragini','kumar','f','10-JAN-
1983','8739870576','ragini@gmail.com','7824 Mcclum blvd');

insert into COURSE_TAUGHT(CID ,FID ) values (1,1);
insert into COURSE_TAUGHT(CID ,FID ) values (2,2);
insert into COURSE_TAUGHT(CID ,FID ) values (3,3);
insert into COURSE_TAUGHT(CID ,FID ) values (4,4);
insert into COURSE_TAUGHT(CID ,FID ) values (5,5);
insert into COURSE_TAUGHT(CID ,FID ) values (6,1);
insert into COURSE_TAUGHT(CID ,FID ) values (7,2);
insert into COURSE_TAUGHT(CID ,FID ) values (8,6);

insert into NOTIFICATION(NOTIF_ID,TID,NOTIF_DESC)values
(1,1,'Assignment submission due today');
insert into NOTIFICATION(NOTIF_ID,TID,NOTIF_DESC)values
(2,2,'Reserve seat for Exam 2');
insert into NOTIFICATION(NOTIF_ID,TID,NOTIF_DESC)values
(3,3,'Negative marking for late submission');
insert into NOTIFICATION(NOTIF_ID,TID,NOTIF_DESC)values
(4,4,'Todays classes cancelled');
insert into NOTIFICATION(NOTIF_ID,TID,NOTIF_DESC)values
(5,5,'Project posted');
insert into NOTIFICATION(NOTIF_ID,TID,NOTIF_DESC)values (6,6,'Exam
marks posted');
insert into NOTIFICATION(NOTIF_ID,TID,NOTIF_DESC)values
(7,1,'Classes postponed');
insert into NOTIFICATION(NOTIF_ID,TID,NOTIF_DESC)values (8,1,'HW
posted');
```

```
insert into NOTIFICATION_TYPE(NOTIF_ID,NOTIF_TYPE)values
(1,'Assignment');
insert into NOTIFICATION_TYPE(NOTIF_ID,NOTIF_TYPE)values
(2,'Exam');
insert into NOTIFICATION_TYPE(NOTIF_ID,NOTIF_TYPE)values
(3,'Assignment');
insert into NOTIFICATION_TYPE(NOTIF_ID,NOTIF_TYPE)values
(4,'Class');
insert into NOTIFICATION_TYPE(NOTIF_ID,NOTIF_TYPE)values
(5,'Project');
insert into NOTIFICATION_TYPE(NOTIF_ID,NOTIF_TYPE)values
(6,'Exam');
insert into NOTIFICATION_TYPE(NOTIF_ID,NOTIF_TYPE)values
(7,'Class');
insert into NOTIFICATION_TYPE(NOTIF_ID,NOTIF_TYPE)values (8,'HW');

insert into QUIZ(QUIZ_ID,TID,LVLS) values (1,1,1);
insert into QUIZ(QUIZ_ID,TID, LVLS) values (2,1,2);
insert into QUIZ(QUIZ_ID,TID, LVLS) values (3,1,3);
insert into QUIZ(QUIZ_ID,TID, LVLS) values (4,2,1);
insert into QUIZ(QUIZ_ID,TID, LVLS) values (5,2,2);
insert into QUIZ(QUIZ_ID,TID, LVLS) values (6,2,3);
insert into QUIZ(QUIZ_ID,TID, LVLS) values (7,3,1);
insert into QUIZ(QUIZ_ID,TID, LVLS) values (8,3,2);
insert into QUIZ(QUIZ_ID,TID, LVLS) values (9,4,1);
insert into QUIZ(QUIZ_ID,TID, LVLS) values (10,5,1);
insert into QUIZ(QUIZ_ID,TID, LVLS) values (11,5,2);
insert into QUIZ(QUIZ_ID,TID, LVLS) values (12,6,1);
insert into QUIZ(QUIZ_ID,TID, LVLS) values (13,6,2);

insert into QUESTIONS(QUESTION_ID, QZ_ID,QUESTION) values
(1,1,'What is DB ');
insert into QUESTIONS(QUESTION_ID, QZ_ID,QUESTION) values
(2,2,'What is SQL');
```

```
insert into QUESTIONS(QUESTION_ID, QZ_ID,QUESTION) values
(3,3,'What is MONGODB');
insert into QUESTIONS(QUESTION_ID, QZ_ID,QUESTION) values
(4,4,'What is STATISTICS');
insert into QUESTIONS(QUESTION_ID, QZ_ID,QUESTION) values
(5,5,'What is REGRESSION');
insert into QUESTIONS(QUESTION_ID, QZ_ID,QUESTION) values
(6,6,'What is MONTE CARLO');
insert into QUESTIONS(QUESTION_ID, QZ_ID,QUESTION) values
(7,7,'What is DATA ANALYSIS');
insert into QUESTIONS(QUESTION_ID, QZ_ID,QUESTION) values
(8,8,'What is UNIVARIATION');
insert into QUESTIONS(QUESTION_ID, QZ_ID,QUESTION) values
(9,9,'What is MULTI TASKING');

insert into USER_ANSWERS(QID, U_ID, ANSWER,QZ_ID) values (7,1,'a
structured set of data ',7);
insert into USER_ANSWERS(QID, U_ID, ANSWER,QZ_ID) values (1,2,'abc
',1);
insert into USER_ANSWERS(QID, U_ID, ANSWER,QZ_ID) values
(9,4,'def',9);
insert into USER_ANSWERS(QID, U_ID, ANSWER,QZ_ID) values (5,5,'ghi
',5);
insert into USER_ANSWERS(QID, U_ID, ANSWER,QZ_ID) values (8,2,'jkl
',8);
insert into USER_ANSWERS(QID, U_ID, ANSWER,QZ_ID) values (6,3,'mno
',6);
insert into USER_ANSWERS(QID, U_ID, ANSWER,QZ_ID) values (3,4,'pqr
',3);

insert into USER_QUIZ (u_id,QZ_ID,score) values(1,7,90);
insert into USER_QUIZ (u_id,QZ_ID,score) values(2,1,80);
insert into USER_QUIZ (u_id,QZ_ID,score) values(3,10,95);
insert into USER_QUIZ (u_id,QZ_ID,score) values(4,9,70);
insert into USER_QUIZ (u_id,QZ_ID,score) values(5,5,80);
insert into USER_QUIZ (u_id,QZ_ID,score) values(1,12,60);
insert into USER_QUIZ (u_id,QZ_ID,score) values(2,13,76);
insert into USER_QUIZ (u_id,QZ_ID,score) values(2,8,89);
insert into USER_QUIZ (u_id,QZ_ID,score) values(3,6,90);
insert into USER_QUIZ (u_id,QZ_ID,score) values(4,3,100);

insert into ASSIGNMENT (ASSIGN_ID,TID,DUE_DATE)values (1,1,'10-OCT-15');
insert into ASSIGNMENT (ASSIGN_ID,TID,DUE_DATE)values (2,2,'15-NOV-15');
insert into ASSIGNMENT (ASSIGN_ID,TID,DUE_DATE)values (3,3,'12-DEC-15');
insert into ASSIGNMENT (ASSIGN_ID,TID,DUE_DATE)values (4,4,'20-FEB-15');
insert into ASSIGNMENT (ASSIGN_ID,TID,DUE_DATE)values (5,5,'10-DEC-15');
insert into ASSIGNMENT_MARKS(ASSIGN_ID,U_ID,TOTAL_POINS) values (1,2,100);
insert into ASSIGNMENT_MARKS(ASSIGN_ID,U_ID,TOTAL_POINS) values (2,5,90);
insert into ASSIGNMENT_MARKS(ASSIGN_ID,U_ID,TOTAL_POINS) values (3,1,80);
insert into ASSIGNMENT_MARKS(ASSIGN_ID,U_ID,TOTAL_POINS) values (4,4,95);
insert into ASSIGNMENT_MARKS(ASSIGN_ID,U_ID,TOTAL_POINS) values (5,3,85);
```

STORED PROCEDURES BY USING PL/SQL

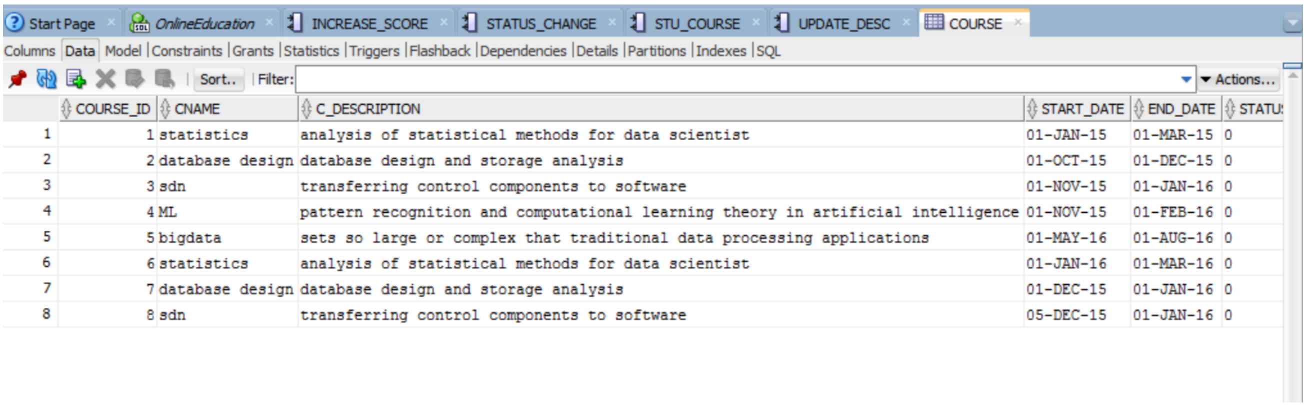
1. **PL/SQL stored procedure to change the status of a course to 1 if the current date is between the start & end date of the course and make it zero if it is not.**

```
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;
```

Procedure Call:

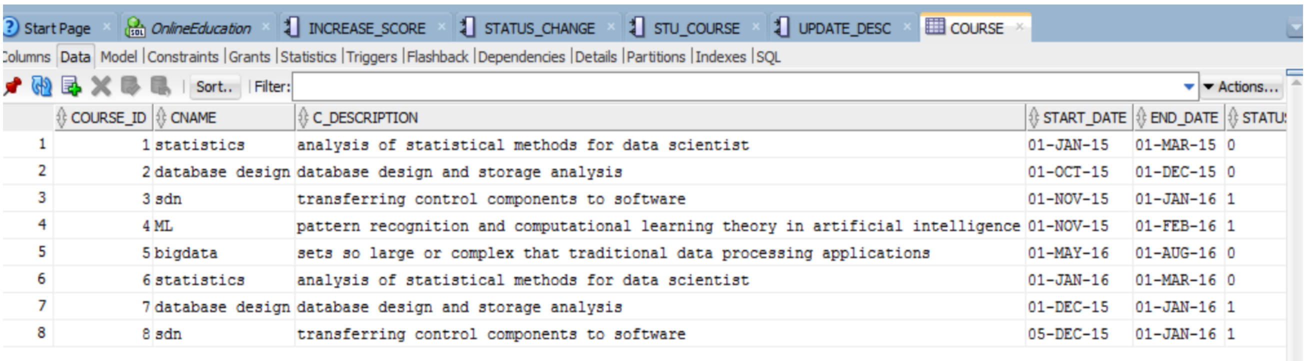
```
SET SERVEROUTPUT ON;
BEGIN
STATUS_CHANGE;
END;
```

Data before :



	COURSE_ID	CNAME	C_DESCRIPTION	START_DATE	END_DATE	STATU
1	1	statistics	analysis of statistical methods for data scientist	01-JAN-15	01-MAR-15	0
2	2	database design	database design and storage analysis	01-OCT-15	01-DEC-15	0
3	3	sdn	transferring control components to software	01-NOV-15	01-JAN-16	0
4	4	ML	pattern recognition and computational learning theory in artificial intelligence	01-NOV-15	01-FEB-16	0
5	5	bigdata	sets so large or complex that traditional data processing applications	01-MAY-16	01-AUG-16	0
6	6	statistics	analysis of statistical methods for data scientist	01-JAN-16	01-MAR-16	0
7	7	database design	database design and storage analysis	01-DEC-15	01-JAN-16	0
8	8	sdn	transferring control components to software	05-DEC-15	01-JAN-16	0

Data After :



	COURSE_ID	CNAME	C_DESCRIPTION	START_DATE	END_DATE	STATU
1	1	statistics	analysis of statistical methods for data scientist	01-JAN-15	01-MAR-15	0
2	2	database design	database design and storage analysis	01-OCT-15	01-DEC-15	0
3	3	sdn	transferring control components to software	01-NOV-15	01-JAN-16	1
4	4	ML	pattern recognition and computational learning theory in artificial intelligence	01-NOV-15	01-FEB-16	1
5	5	bigdata	sets so large or complex that traditional data processing applications	01-MAY-16	01-AUG-16	0
6	6	statistics	analysis of statistical methods for data scientist	01-JAN-16	01-MAR-16	0
7	7	database design	database design and storage analysis	01-DEC-15	01-JAN-16	1
8	8	sdn	transferring control components to software	05-DEC-15	01-JAN-16	1

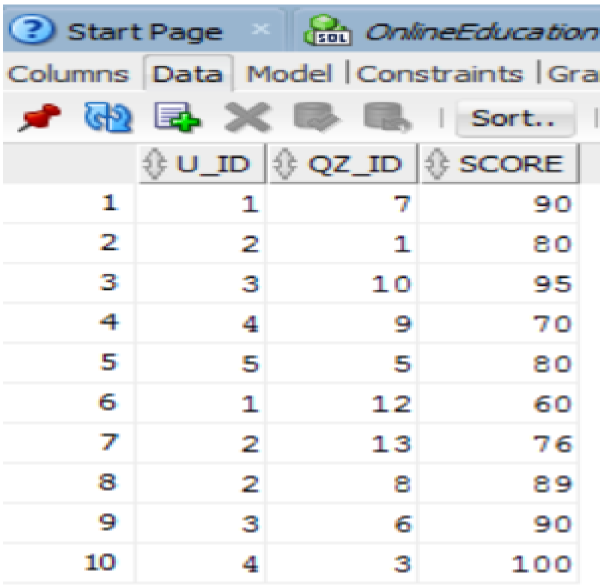
2. **PL/SQL stored procedure to update the quiz score to 100 of the students with score between 91& 99.**

```
CREATE OR REPLACE PROCEDURE INCREASE_SCORE AS
BEGIN
DECLARE
ScoreDetails USER_QUIZ%ROWTYPE;
     CURSOR SCORE_CHECK IS SELECT * FROM USER_QUIZ WHERE SCORE
BETWEEN 90 AND 99;
BEGIN
     OPEN SCORE_CHECK;
    DBMS_OUTPUT.PUT_LINE('CURSOR OPENED');
     LOOP
          FETCH SCORE_CHECK INTO ScoreDetails;
          EXIT WHEN (SCORE_CHECK%NOTFOUND);
        UPDATE USER_QUIZ SET SCORE = 100 WHERE U_ID =
ScoreDetails.U_ID AND QZ_ID = ScoreDetails.QZ_ID;
     END LOOP;
     CLOSE SCORE_CHECK;
END;
END INCREASE_SCORE;
```

Procedure Call:

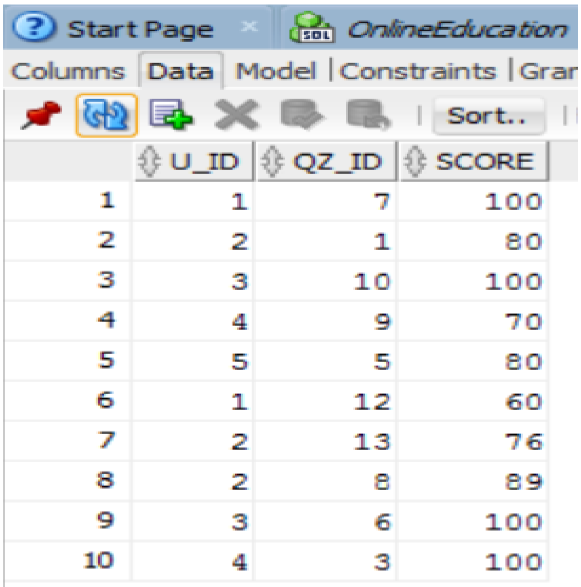
```
SET SERVEROUTPUT ON;
BEGIN
INCREASE_SCORE;
END;
```

Data before :



	U_ID	QZ_ID	SCORE
1	1	7	90
2	2	1	80
3	3	10	95
4	4	9	70
5	5	5	80
6	1	12	60
7	2	13	76
8	2	8	89
9	3	6	90
10	4	3	100

Data After:



	U_ID	QZ_ID	SCORE
1	1	7	100
2	2	1	80
3	3	10	100
4	4	9	70
5	5	5	80
6	1	12	60
7	2	13	76
8	2	8	89
9	3	6	100
10	4	3	100

3. **Stored PL/SQL procedure that updates the T_description of the TOPIC table to "Topic description not provided" for all the topics with T_description value = NULL.**

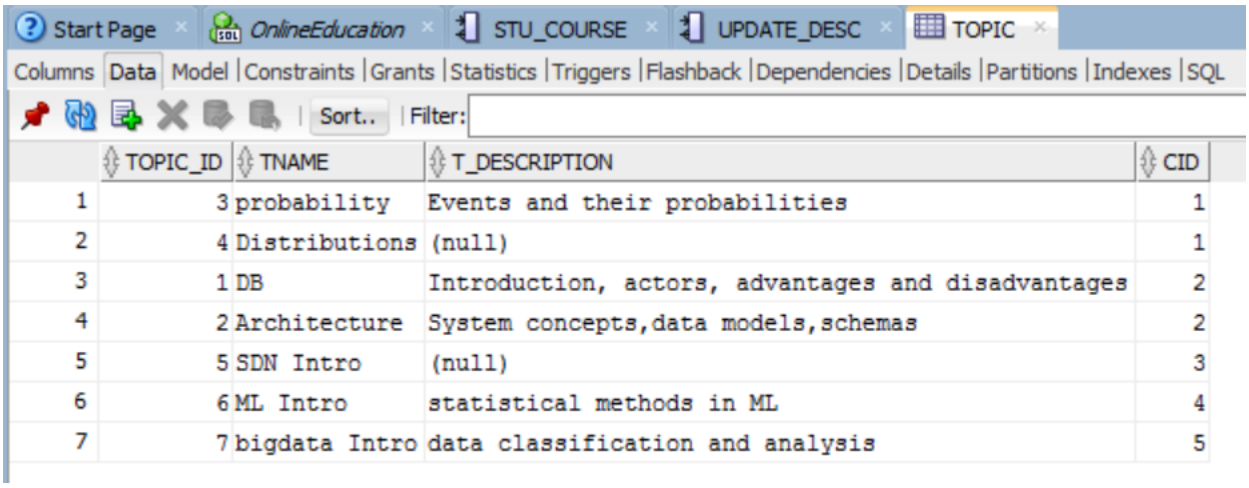
```
CREATE OR REPLACE PROCEDURE UPDATE_DESC AS
BEGIN
DECLARE
TopicDetails TOPIC%ROWTYPE;
     CURSOR DESC_CHECK IS SELECT * FROM TOPIC;
BEGIN
     OPEN DESC_CHECK;
    DBMS_OUTPUT.PUT_LINE('CURSOR OPENED');
     LOOP
          FETCH DESC_CHECK INTO TopicDetails;
          EXIT WHEN (DESC_CHECK%NOTFOUND);
```

```
IF(TopicDetails.T_DESCRIPTION IS NULL) THEN
        UPDATE TOPIC SET T_DESCRIPTION = 'Topic description not provided' WHERE TOPIC_ID = TopicDetails.TOPIC_ID;
        END IF;
     END LOOP;
     CLOSE DESC_CHECK;
END;
END UPDATE_DESC;
```

Procedure Call:

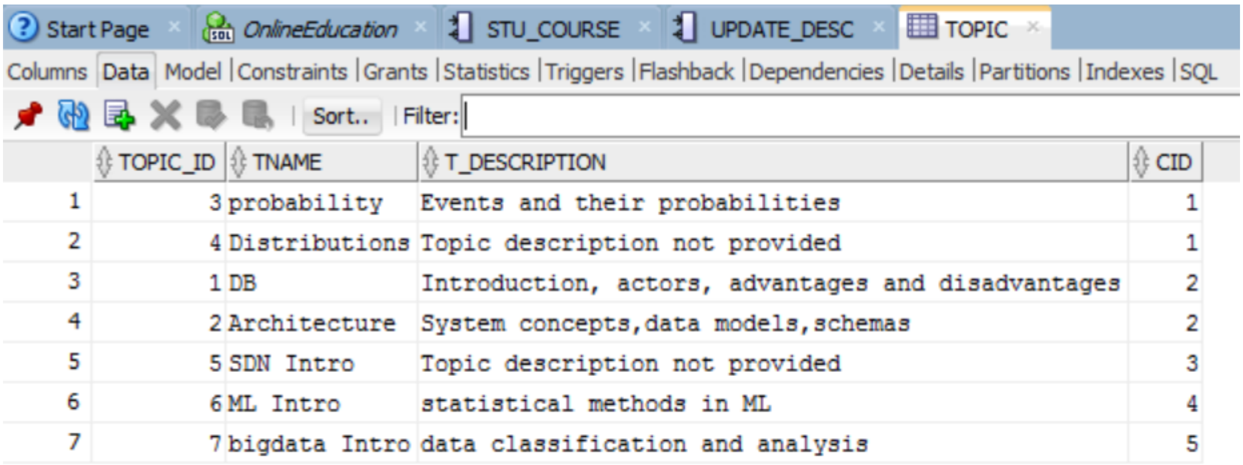
```
SET SERVEROUTPUT ON;
BEGIN
UPDATE_DESC;
END;
```

Data Before:



	TOPIC_ID	TNAME	T_DESCRIPTION	CID
1	3	probability	Events and their probabilities	1
2	4	Distributions	(null)	1
3	1	DB	Introduction, actors, advantages and disadvantages	2
4	2	Architecture	System concepts,data models,schemas	2
5	5	SDN Intro	(null)	3
6	6	ML Intro	statistical methods in ML	4
7	7	bigdata Intro	data classification and analysis	5

Data After:



	TOPIC_ID	TNAME	T_DESCRIPTION	CID
1	3	probability	Events and their probabilities	1
2	4	Distributions	Topic description not provided	1
3	1	DB	Introduction, actors, advantages and disadvantages	2
4	2	Architecture	System concepts,data models,schemas	2
5	5	SDN Intro	Topic description not provided	3
6	6	ML Intro	statistical methods in ML	4
7	7	bigdata Intro	data classification and analysis	5

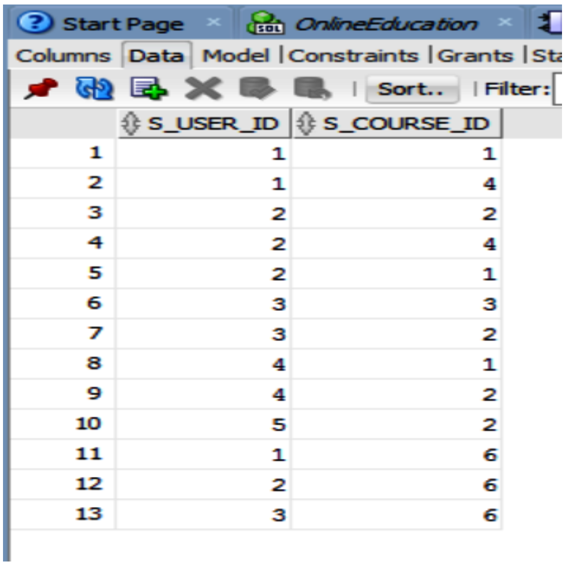
4. Stored PL/SQL procedure to find the details of courses beginning in Jan 2016 and has at least 2 students enrolled in it;

```
create or replace PROCEDURE STU_COURSE AS
BEGIN
  DECLARE
  CourseDesc COURSE%ROWTYPE;
  COUNT_VAR  INTEGER;
  CURSOR COURSE_CHECK IS SELECT * FROM COURSE WHERE START_DATE
BETWEEN '01-JAN-16' AND '31-JAN-16';
  BEGIN
    OPEN COURSE_CHECK;
      LOOP
          FETCH COURSE_CHECK INTO CourseDesc;
          EXIT WHEN (COURSE_CHECK%NOTFOUND);
          SELECT COUNT(S_USER_ID) INTO COUNT_VAR FROM
STUDENT_COURSE WHERE S_COURSE_ID = CourseDesc.COURSE_ID;
          IF(COUNT_VAR > 1) THEN
              DBMS_OUTPUT.PUT_LINE('ID: '||CourseDesc.COURSE_ID ||'
COURSE NAME: '||CourseDesc.CNAME);
          END IF;
      END LOOP;
      CLOSE COURSE_CHECK;
  END;
END STU_COURSE;
```

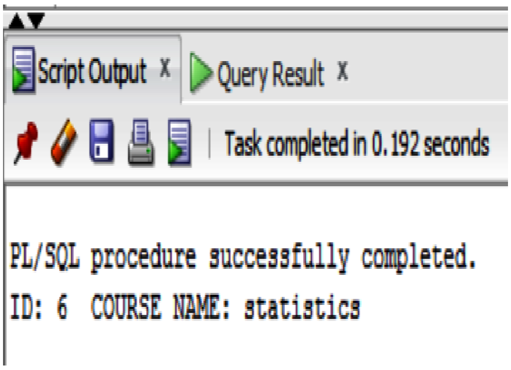
Procedure Call:

```
SET SERVEROUTPUT ON;
BEGIN
STU_COURSE;
END;
```

Data Before:



	S_USER_ID	S_COURSE_ID
1	1	1
2	1	4
3	2	2
4	2	4
5	2	1
6	3	3
7	3	2
8	4	1
9	4	2
10	5	2
11	1	6
12	2	6
13	3	6



```
Task completed in 0.192 seconds

PL/SQL procedure successfully completed.
ID: 6  COURSE NAME: statistics
```