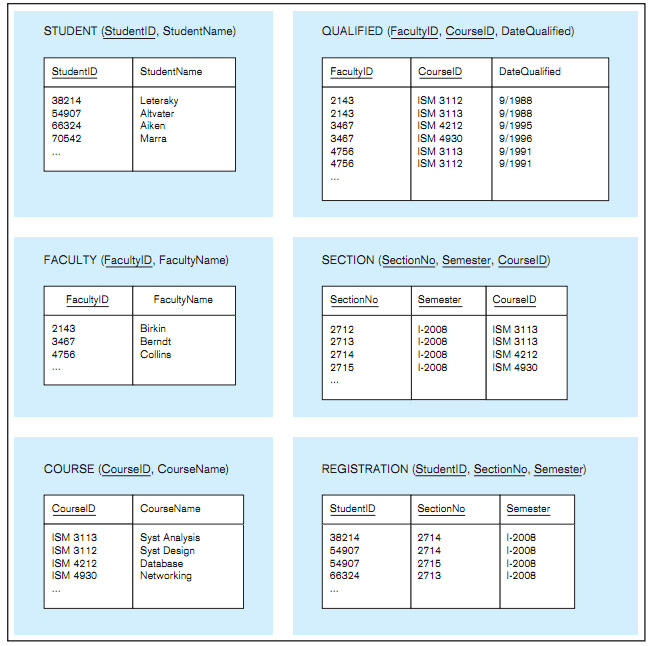
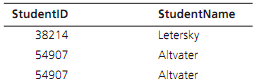
**TUTORIAL 6**

1. Write a database descrip tion for each of the relations shown, using SQL DDL (shorten, abbreviate, or chan ge any data names, as needed for your SQL version). Assume the following attribut e data types:

StudentID (integer, primary key), StudentName (25 characters), Fac ultyID (integer, primary key), FacultyName (25 characters), CourseID (8 characters, primary key), CourseName (15 characters), DateQualified (date), SectionNo (integer, primary key), Semester (7 characters)



**2**. Create an SQL VIEW for foollowing table



**3**. Write SQL data definition c ommands for each of the following queries:

a. How would you add an attribute, Class, to the Student table?

b. How would you remove the Registration table?

c. How would you change the FacultyName field from 25 characters to 40 characters?

1. Write SQL commands for the following:

a. Create two different forms of the INSERT command to add a student with a student ID of 65798 and last name Lopez to the Student table.

b. Now write a command that will remove Lopez from the Student table.

c. Create an SQL command that will modify the name of course ISM 4212 from Database to Introduction to Relational Databases.

**5**. Write SQL queries to answer the following questions:

a. Which students have an ID number that is less than 50000?

b. What is the name of the faculty member whose ID is 4756?

c. What is the smallest section number used in the first semester of 2008?

**6**. Write SQL queries to answer the following questions:

a. How many students are enrolled in Section 2714 in the first semester of 2008?

b. Which faculty members have qualified to teach a course since 1993? List the faculty ID, course, and date of qualification.

**7**. Write SQL queries to answer the following questions:

a. Which students are enrolled in Database and Networking? (Hint: Use SectionNo for each class so you can determine the answer from the Registration table by itself.)

b. Which instructors cannot teach both Syst Analysis and Syst Design?

**8**. Write SQL queries to answer the following questions:

a. What are the courses included in the Section table? List each course only once.

b. List all students in alphabetical order by StudentName.

c. List the students who are enrolled in each course in Semester I, 2008. Group the students by the sections in which they are enrolled.

d. List the courses available. Group them by course prefix. (ISM is the only prefix shown, but there are many others throughout the university.)

**SOLUTION**

**1.**

CREATE TABLE STUDENT (

STUDENT\_ID INTEGER UNSIGNED NOT NULL,

STUDENT\_NAME VARCHAR(25),

CONSTRAINT STUDENT\_PK PRIMARY KEY (STUDENT\_ID)

);

CREATE TABLE FACULTY

(

FACULTY\_ID INTEGER UNSIGNED NOT NULL,

FACULTY\_NAME VARCHAR(25),

CONSTRAINT FACULTY\_PK PRIMARY KEY (FACULTY\_ID)

);

CREATE TABLE COURSE

(COURSE\_ID CHAR(8) NOT NULL,

COURSE\_NAME VARCHAR(15),

CONSTRAINT COURSE\_PK PRIMARY KEY (COURSE\_ID));

CREATE TABLE SECTION

(

SECTION\_NO

INTEGER

UNSIGNED NOT NULL,

SEMESTER

CHAR(7)

NOT NULL,

COURSE\_ID

CHAR(8),

CONSTRAINT SECTION\_PK PRIMARY KEY (SECTION\_NO, SEMESTER),

CONSTRAINT SECTION\_FK FOREIGN KEY (COURSE\_ID) REFERENCES COURSE (COURSE\_ID)

);

CREATE TABLE QUALIFIED

(FACULTY\_ID

INTEGER

UNSIGNED NOT NULL ,

COURSE\_ID

CHAR(8)

NOT NULL,

DATE\_QUALIFIED DATE,

CONSTRAINT QUALIFIED\_PK PRIMARY KEY (FACULTY\_ID, COURSE\_

CONSTRAINT QUALIFIED\_FACULTY\_FK FOREIGN KEY (FACULTY\_ID) REFERENCES FACULTY (FACULTY\_ID),

CONSTRAINT QUALIFIED\_COURSE\_FK FOREIGN KEY (COURSE\_ID) REFERENCES COURSE (COURSE\_ID));

|  |  |  |
| --- | --- | --- |
| CREATE TABLE REGISTRATION |  |  |
| (STUDENT\_ID | INTEGER | UNSIGNED NOT NULL, |
| SECTION\_NO | INTEGER | UNSIGNED NOT NULL, |
| SEMESTER | CHAR(7) | NOT NULL, |

CONSTRAINT REGISTERED\_PK PRIMARY KEY (STUDENT\_ID, SECTION\_NO, SEMESTER),

CONSTRAINT STUDENT\_REGISTERED\_FK FOREIGN KEY(STUDENT\_ID) REFERENCES STUDENT(STUDENT\_ID),

CONSTRAINT COURSE\_REGISTERED\_FK FOREIGN KEY (SECTION\_NO, SEMESTER) REFERENCES SECTION(SECTION\_NO, SEMESTER));

1. CREATE VIEW STUDENT\_VIEW AS SELECT STUDENT\_ID, STUDENT\_NAME FROM STUDENT;

3.

a. ALTER TABLE STUDENT ADD CLASS VARCHAR(5)

b. DROP TABLE REGISTRATION;

1. ALTER TABLE FACULTY MODIFY FACULTY\_NAME VARCHAR(50);

4.

a. INSERT INTO STUDENT VALUES (65798, “LOPEZ”);

INSERT INTO STUDENT (STUDENT\_ID, STUDENT\_NAME) VALUES (65798, “LOPEZ”);

b. DELETE FROM STUDENT WHERE STUDENT\_NAME = ‘LOPEZ’;

1. UPDATE COURSE SET COURSE\_NAME = ‘INTRODUCTION TO RELATIONAL DATABASE’ WHERE COURSE\_ID= ‘ISM 4212’;

5.

1. SELECT STUDENT\_ID, STUDENT\_NAME FROM STUDENT WHERE STUDENT\_ID <50000;
2. SELECT STUDENT\_NAME FROM STUDENT WHERE STUDENT\_ID = 4756;
3. SELECT MIN(SECTION\_NO) FROM REGISTRATION WHERE SEMESTER= ‘I-2008’;

6.

1. SELECT COUNT(STUDENT\_ID) FROM REGISTRATION WHERE SECTION\_NO =2714 AND SEMESTER= ‘I-2008’
   1. SELECT Q.FACULTY\_ID, C.COURSE\_NAME, Q.DATE\_QUALIFIED

FROM FACULTY AS F, COURSE AS C, QUALIFIED AS Q

WHERE YEAR(Q.DATE\_QUALIFIED) >= 1993

AND C.COURSE\_ID =Q.COURSE\_ID;

SELECT FACULTY\_ID, COURSE\_ID, DATE\_QUALIFIED

FROM QUALIFIED

WHERE DATE\_QUALIFIED >= ‘1993-01-01';

7. a,

SELECT STUDENT\_ID, COUNT(\*)

FROM REGISTRATION

WHERE SECTION\_NO IN (2714,2715)

GROUP BY STUDENT\_ID

HAVING COUNT(\*) > 1;

1. SELECT FACULTY\_ID,COUNT(\*)FROM QUALIFIED WHERE COURSE\_ID IN ('ISM 3113','ISM 3112')

GROUP BY INSTRUCTOR\_ID HAVING COUNT(\*) = 1;

8

1. SELECT DISTINCT COURSE\_ID FROM SECTION; B. SELECT STUDENT\_NAME

FROM STUDENT

ORDER BY STUDENT\_NAME;

1. SELECT SECTION\_NO,SEMESTER,STUDENT\_ID FROM REGISTRATION

WHERE SEMESTER = 'I-2008'

ORDER BY SECTION\_NO, SEMESTER, STUDENT\_ID;

1. SELECT COURSE\_ID,COURSE\_NAME

FROM COURSE

ORDER BY COURSE\_ID;