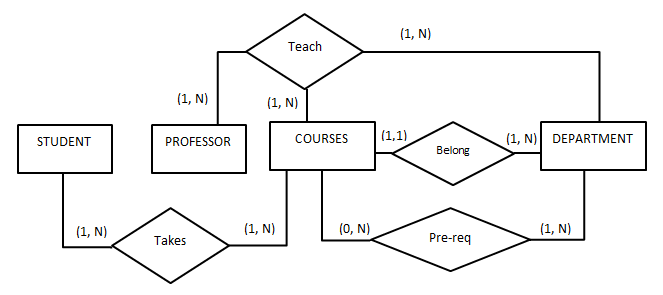
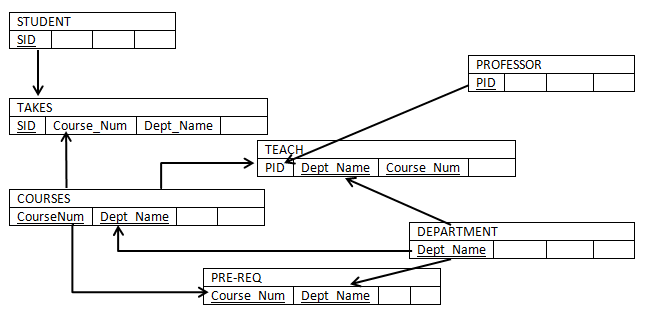
**ER-Diagram: Student Record Management System**

****

**Database Schema**



Note: The above diagram illustrates the relationships between the tables used in the exercise. The lines show the connection between tables.

**Physical schema Description**

The tables are described in order of relationship described in the previous figure.

STUDENT

There is one row in the STUDENT table for each student registering for the course.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Meaning | NULL allowed |
| SID | Varchar (10) | Student ID | No |
| Name | Varchar (45) | Last, First Name | Yes |
| Address | Varchar (45) | Address | Yes |

PROFESSOR

There is one row in the PROFESSOR table for each professor in the department

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Meaning | NULL allowed |
| PID | Varchar (10) | Professor ID | No |
| Name | Varchar (45) | Last, First Name | Yes |
| Office | Varchar (10) | Office number | Yes |
| DateofBirth | Date | Age of Professor | Yes |

COURSE

There is one row in the COURSE table for each course offered that quarter.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Meaning | NULL allowed |
| CourseNum | Integer | Course number | No |
| DeptName | Varchar(45) | Name of Department | No |
| CourseName | Varchar(45) | Course Name | Yes |
| ClassRoom | Varchar(45) | Room Number | Yes |
| Enrollment | Integer | Number of Students Enrolled | Yes |

DEPARTMENTS

There is one row in the DEPARTMENTS table for each department in the University.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Meaning | NULL allowed |
| DeptName | Varchar (45) | Name of Department | No |
| ChairmanID | Varchar (45) | Name of Chairman | Yes |

PREREQ

There is one row in the PREREQ table for each pre-requite for any course.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Meaning | NULL allowed |
| CourseNum | Integer | Course Number | No |
| DeptName | Varchar (45) | Name of Department | No |
| PreReqNumber | Integer | Pre requisite Number | Yes |
| PreReqDeptName | Varchar (45) | Pre requisite Dept Name | Yes |

TEACH

There is a row in the TEACH table for each course taught by a professor.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Meaning | NULL allowed |
| PID | Varchar(10) | Professor who teaches the course | No |
| CourseNum | Integer | The course taught | No |
| DeptName | Varchar (45) | The department the course taught | No |

TAKE

There is a row in the TAKE table for each course enrolled by a student.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Meaning | NULL allowed |
| SID | Varchar (10) | ID of student taking a course | No |
| CourseNum | Integer | The course take by student | No |
| DeptName | Varchar (45) | The department of course | No |
| Grade | Decimal(4,2) | The grade obtained | Yes |
| ProfessorEval | Decimal(4,2) | The professor evaluation | Yes |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Instructions:**

1. Ensure that the database is loaded using the provided database script “StudentdbW21.sql”.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**HANDSON**

use Studentdbw21; /\* Replace <newdb> with the name of the database Studentdbw21\*/

**Example 1: Joining multiple tables:** Using natural join of the Students, Take, Teach and Professor tables

select STUDENT.\*,Take.\*,Teach.\*,PROFESSOR.\*

from STUDENT, Take , Teach, PROFESSOR

where STUDENT.SID = Take.SID AND PROFESSOR.PID = Teach.PID AND Take.coursenum = Teach.courseNum

AND Take.DeptName = Teach.DeptName;

**Example 2: Using Aliasing:** List all students who took courses from the Department of Engineering and Science

select S.\*,t.\*

from STUDENT S, Take T

where S.SID = T.SID AND T.DeptName = 'Engineering and Science';

**Example 3:** **Using arithmetic operations in SQL**

/\*Using the Arithmetic operator DATEDIFF that takes two input values e.g DATEDIFF('2021-01-22', DateofBirth) and returns difference of the dates in **DAYS**,

**QUERY:** Write a SQL query that lists the names and ages (in days) of Professors ordered in the descending order

select p1.name, p1.dateofbirth,

DATEDIFF('2021-01-22',p1.DateOfBirth) as age

from PROFESSOR p1

**order by** dateofbirth desc;

**Example 4: Using Set Operator - UNION**

The following is an illustration of combining two relationships using the UNION operator.

select coursenum

from PreReq

**UNION**

select coursenum

from Take;

**Example 5: Using Set Operator – UNION ALL**

/\* UNION ALL includes duplicates\*/

select courseNum

from PreReq

**UNION ALL**

select coursenum

from Take;

**Sub-Queries:**

**Example 6: Using Subqueries in WHERE clause**

**Query:** List the course numbers of those pre-requisites that have been taken by students.

select PreReq.coursenum

from PreReq

where PreReq.coursenum **IN** (select Take.coursenum

from Take);

**Example 7: Using NOT EXISTS**

**Query:** List of courses that do not have a pre-requisite

select \*

from COURSE c

where **NOT EXISTS** (select \*

from PreReq p

where p.coursenum = c.coursenum);

**Example 8: Using Operator ALL**

**Query:** Find the course that has the highest enrollment

select max(p.enrollment) from COURSE p;

***OR***

select \*

from COURSE c

where c.enrollment **>= ALL** (select p.enrollment

from COURSE p);

**Example 9: Using Operator ANY**

**Query:** List all courses EXCEPT those with the highest enrollment

select \*

from COURSE c

where c.enrollment **< ANY** (select p.enrollment

from COURSE p);

**Example 10: Using Subqueries in the FROM clause**

**Query:** Obtain the names of students who took course number 101 from the Department of Engineering and Science, and obtained a grade >= 3.40

select s.Name

from STUDENT s, (select \* from Take where Grade >=3.40) t

where t.sid = s.sid AND t.courseNum = '101'

AND t.DeptName = 'Engineering and Science';

**Example 11: Using Subqueries in the SELECT clause**

**Query:** Obtain a list of courses and the names of the professors that teach them.

select th.\*,(select p.Name

from PROFESSOR p

where p.pid = th.pid) as 'ProfName'

from Teach th;