**Lab - 10**

**Project**

* Consider the following set of requirements for a UNIVERSITY database that is used to keep track of students' transcripts.

**Description**:

a) The university keeps track of each student's name, student number, Social Security number, current address and phone number, permanent address and phone number, birth date, sex, class (freshman, sophomore, ..., graduate), major department, minor department (if any), and degree program (B.A., B.S., ..., Ph.D.). Some user applications need to refer to the city, state, and ZIP Code of the student's permanent address and to the student's last name. Both Social Security number and student number have unique values for each student.

b) Each department is described by a name, department code, office number, office phone number, and college. Both name and code have unique values for each department.

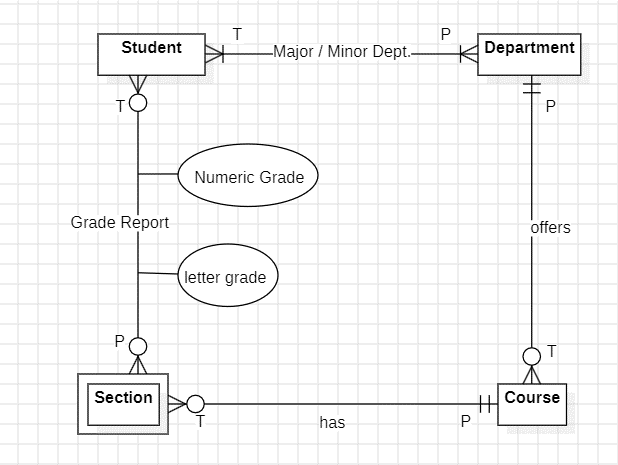
c) Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of the course number is unique for each course.

d) Each section has an instructor, semester, year, course, and section number. The section number distinguishes sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each semester.

e) A grade report has a student, section, letter grade, and numeric grade (0, 1, 2, 3,or 4).

1. Draw an ER diagram for the schema(Make sure to use correct notation for specifying cardinality ratios, total/partial participations, key constraints.)

|  |  |
| --- | --- |
|  | |
|  | A diagram of a course  Description automatically generated |
|  | |



1. Design the relational schema for this application.

|  |
| --- |
| STUDENT(StudentID, SSN,  FirstName, LastName,  CurrentAddress, CurrentPhone,  PermanentAddres, PermanentPhone,  BirthDate, Sex, Class,  MajorDept, MinorDept,  DegreeProgram) |
| DEPARTMENT(DepartmentCode, DepartmentName,  OfficeNumber, OfficePhone, College) |
| COURSE( CourseNumber, CourseName,  Description, SemesterHours,  Level, OfferingDept) |
| SECTION(SectionNumber , CourseNumber,  Instructor, Semester, Year) |
| GRADEREPORT(StudentID, SectionID, CourseNumber  LetterGrade, NumericGrade) |

1. Create tables in SQL for all the relations along with constraints.

|  |  |
| --- | --- |
| QUERY | CREATE TABLE U\_Department (  DepartmentCode NUMBER(10) PRIMARY KEY,  DepartmentName VARCHAR2(50) UNIQUE NOT NULL,  OfficeNumber VARCHAR2(10) NOT NULL,  OfficePhone CHAR(10) NOT NULL,  College VARCHAR(50) NOT NULL  ); |
| OUTPUT |  |
|  |

|  |  |
| --- | --- |
| QUERY | CREATE TABLE u\_student (      StudentID NUMBER(12) PRIMARY KEY,      SSN CHAR(9) UNIQUE NOT NULL,      FirstName VARCHAR2(50) NOT NULL,      LastName VARCHAR2(50),      CurrentAddress VARCHAR2(100) NOT NULL,      CurrentPhone CHAR(10) NOT NULL,      PermanentAddress VARCHAR2(100),      PermanentPhone CHAR(10),      BirthDate DATE NOT NULL,      Sex CHAR(1) NOT NULL,      Class VARCHAR(20) NOT NULL,      MajorDepartment NUMBER(10),      MinorDepartment NUMBER(10),      CONSTRAINT u\_student\_maj\_dept      DegreeProgram VARCHAR(10) NOT NULL,          FOREIGN KEY(MajorDepartment)          REFERENCES U\_Department(DepartmentCode),      CONSTRAINT u\_student\_min\_dept          FOREIGN KEY(MinorDepartment)          REFERENCES U\_Department(DepartmentCode)  ); |
| OUTPUT |  |
|  |

|  |  |
| --- | --- |
| QUERY | CREATE TABLE u\_course (  CourseNumber NUMBER(15) PRIMARY KEY,  CourseName VARCHAR2(100) NOT NULL,  Description VARCHAR2(255) NOT NULL,  SemesterHours NUMBER(6) NOT NULL CHECK(SemesterHours > 20),  "Level" VARCHAR2(20) NOT NULL,  OfferingDepartment NUMBER(10),  CONSTRAINT offering\_dept\_fk  FOREIGN KEY(OfferingDepartment)  REFERENCES u\_department(DepartmentCode)  ); |

|  |  |
| --- | --- |
| OUTPUT |  |
|  |

|  |  |
| --- | --- |
| QUERY | CREATE TABLE u\_section (  SectionNumber NUMBER(10),  CourseNumber NUMBER(15),  Instructor VARCHAR(50) NOT NULL,  Semester VARCHAR(10) NOT NULL,  Year NUMBER(2) NOT NULL,  CONSTRAINT section\_course\_fk  FOREIGN KEY(CourseNumber)  REFERENCES u\_course(CourseNumber),  CONSTRAINT section\_course\_pk  PRIMARY KEY (CourseNumber, SectionNumber)  ); |
| OUTPUT |  |
|  |

|  |  |
| --- | --- |
| QUERY | CREATE TABLE u\_grade (  StudentID NUMBER(12) NOT NULL,  SectionNumber NUMBER(10) NOT NULL,  CourseNumber NUMBER(15) NOT NULL,  LetterGrade CHAR(2),  NumericGrade NUMBER(3),  CONSTRAINT grade\_student\_fk  FOREIGN KEY (StudentID)  REFERENCES u\_student(StudentID),  CONSTRAINT grade\_section\_fk  FOREIGN KEY (SectionNumber, CourseNumber)  REFERENCES u\_section(SectionNumber, CourseNumber),  CONSTRAINT grade\_course\_st\_pk  PRIMARY KEY (StudentID, CourseNumber, SectionNumber)  ); |

|  |  |
| --- | --- |
| OUTPUT |  |
|  |