University Academic Manager

Rahul Gaikwad

**NARRATIVE**

Each Academic University has numerous departments headed by some faculty member. University offers various courses during each semester. Number of Faculty members works under each department.

Faculty members teach various courses during each semester. Some of the faculty member also works as a coordinator for the project. Each project has name and number of student’s works on the project. Students admitted to department can search and register for courses being offered in particular semester. Student can see his transcript for the courses he registered and completed.

Faculty, who teach course/class, conduct classes during the semester and at the end assign grades to the enrolled students, based on some course work completion criteria.

Each department also has many alumni. It is necessary to maintain the details of each alumni and their degrees.

With the help of University Academic Manager being designed student can search various courses being offered and register under them. Student can drop courses and see transcripts. Faculty can see the list of the student enrolled in the class and assign them grades. Project co-coordinator can search for the student working on the project

**Functional Requirements:**

The users of the system will be student, faculty, adviser, and administrator. The functional requirements for the student, faculty, adviser, and administrator are given as follows:

**Student:**

1. The student must be able to view the courses being offered.
2. The student must be able to register for all open courses being offered.
3. The student must be able drop the courses which are not yet graded.
4. The student should be able view his transcript.
5. The student should be able to update his details like address phone no.

**Faculty:**

1. The faculty should be able to view classes being offered.
2. The faculty /instructor should be able view the list of all student enrolled in the class.
3. The faculty must be able to assign grade to the student enrolled in his section.
4. The faculty /instructor must be able add/drop student in the class.
5. The faculty must be able to generate reports for grade for section.
6. The faculty should be able to update his details like address, phone no, office.

**Grade adviser:**

1. The grad adviser must be able add/drop student in the class on students request.

**Data Requirements:**

For each entity, the lists of the fields that must be stored in the database are given here.

Details about each field length field type and null/not null constraint, primary/foreign key constraint is provided in the data model section

**DEPARTMENT:**

|  |  |
| --- | --- |
| **Field Name** | **Description** |
| Dno | A unique identifier of the department in the system |
| Dname | Name of the department, must be unique |
| Office | Location of departments office |
| Mail | Departments mail id |
| Phone | Departments office phone |
| HeadSSN | Deans SSN of the departments |

**FACULTY:**

|  |  |
| --- | --- |
| **Field Name** | **Description** |
| SSN | A unique identifier of the Faculty in the system |
| NetID | Every person has unique netID , which can be username to access system. |
| Fname | First name |
| Minit | Initial of the middle name |
| Lname | Last name |
| Sex | M-Male  F-Female  O-Other |
| Bdate | Birth Date |
| Office | Office location of faculty |
| Mail | Email id |
| Phone | Phone number |
| Street | Street name and Apartment number of the address |
| City | City of the address |
| State | State of the address |
| Zip | Zip code of the address |
| Dno | Identifies the department to which faculty belongs |

**STUDENT:**

|  |  |
| --- | --- |
| **Field Name** | **Description** |
| Sno | A unique identifier of the Student in the system |
| NetID | Every person has unique netID , which can be username to access system. |
| Fname | First name |
| Minit | Initial of the middle name |
| Lname | Last name |
| Sex | M-Male  F-Female  O-Other |
| Class | 1-Freshman  2-Sophomore  3- Junior  4- Senior  5-Graduate  6-Doctarate |
| Bdate | Birth Date |
| mail | Email id |
| Phone | Phone number |
| Street | Street name and Apartment number of the address |
| City | City of the address |
| State | State of the address |
| Zip | Zip code of the address |
| Dno | Identifies the department to which student belongs |
| Grade | Grade achieved by student , null value means student registered but not graded yet. |

**COURSE:**

|  |  |
| --- | --- |
| **Field Name** | **Description** |
| Cno | A unique identifier of the course. |
| Cname | A unique name of the course. |
| Level | Level of the course.  5000 - Undergraduate  6000 - Graduate level  7000 - Doctoral |
| Credits | Number of credits associated with the course |
| Dno | Identifies the department to which this course belongs |
| Description | Course description. |
| PreRecCno | Pre-requisite course number |

**SECTION:**

|  |  |
| --- | --- |
| **Field Name** | **Description** |
| SectionId | A unique identifier of the SectionId. It is system surrogate key |
| Cno | Identifies the course of section |
| SectionNo | Section number e.g 501,301 |
| Semester | Semester of the section  F-Fall  S-Spring  U-Summer |
| Year | Year of the section |
| InstSSN | Identifies faculty who teaches this course |
| Size | Size of the class.  Default value is 60 |
| Status | Status of the class  O - Open  C - Closed  Default : O |
| TASno | Identifies student who works as a Teaching Assistant for this course if any |
| Description | Description if any |
| Location | Classrooms location. |
| Day | Day e.g. Mon,Tues,Thur,etc |
| fromTime | Class start time in 24Hr format e.g 12:30, 18:30 |
| toTime | Class ends time in 24Hr format e.g 12:30, 18:30 |

**ALUMNI:**

|  |  |
| --- | --- |
| **Field Name** | **Description** |
| NetID | Every alumni has unique netID |
| Fname | First name |
| Minit | Initial of the middle name |
| Lname | Last name |
| Sex | M-Male  F-Female  O-Other |
| Bdate | Birth Date |
| mail | Email id |
| Phone | Phone number |
| Street | Street name and Apartment number of the address |
| City | City of the address |
| State | State of the address |
| Zip | Zip code of the address |
| Degree | Degree title |
| Major | Major of the degree |
| Year | Year of the degree |

**PROJECT:**

|  |  |
| --- | --- |
| **Field Name** | **Description** |
| ProjectId | A unique identifier of the project. |
| ProjectName | Name of the project |
| CordinatorSSN | Identifies the project coordinator. |
| StartDate | Projects Start date |
| EndDate | Projects End date |
| Description | Projects Description |
| Dno | Identifies the department to which this faculty belongs |
| Sno | Identifies the student who works on the project . refers to Sno of student table |

**ER to RELATIONAL MAPPING**

1. **DEPARTMENT:**

(Dno, Dname, Office, Phone, Mail, HeadSSN)

FDs

* Dno --> Dname, Office, Phone, Mail, HeadSSN
* Dname --> Dno

1. **FACULTY:**

(SSN, NetID, Fname, Lname, Minit, Office, Sex, Bdate, mail, Phone, Street, City, State, Zip, Dno)

FDs

* SSN --> Fname, Lname, Minit, Office, Sex, Bdate, mail, Phone, Street, City, State, Zip, Dno
* NetID --> SSN

1. **STUDENT:**

(Sno, NetID, Fname, Lname, Minit, Sex, Bdate, Class, mail, Phone, Street, City, State, Zip, Dno)

FDs

* Sno --> NetID, Fname, Lname, Minit, Sex, Bdate, Class, mail, Phone, Street, City, State, Zip, Dno
* NetID --> Sno

1. **COURSE:**

(Cno, Cname, Level, Credits, Dno, Description)

FDs

* Cno --> Cno, Cname, Level, Credits, Dno, Description
* Cno --> Cname

1. **PRE\_REQ:**

(MainCno,PreReqCno)

1. **SECTION:**

(SectionId, Cno, SectionNo, Semester, Year, InstSSN, Size, Status, TASno, Description)

FDs

* SectionId --> SectionId, Cno, SectionNo, Semester, Year, InstSSN, Size, Status, TASno, Description

1. **SECTION\_TIMES:**

(SectionId,Day, FromTime, ToTime, Location)

FDs

* SectionId, Day, FromTime --> ToTime, Location

1. **GRADE\_REPORT:**

(Sno, SectionId, Grade)

FDs

* Sno --> SectionId, Grade

1. **ALUMNI:**

(NetID, Fname, Lname, Minit, Sex, DOB, mail, Phone, Street, City, State, Zip)

FDs

* NetID --> Fname, Lname, Minit, Sex, DOB, mail, Phone, Street, City, State, Zip

1. **ALUMNI\_DEGREE:**

(NetId, Degree, Major, Year)

FDs

* NetId, Degree --> Major, Year

1. **PROJECT:**

(ProjectId,ProjectName, CordinatorSSN, StartDate, EndDate, Dno, Description)

FDs

* ProjectId --> ProjectName, CordinatorSSN, StartDate, EndDate, Dno, Description

1. **PROJECT\_ STUDENT:**

(Sno, ProjectId, WorkDesc)

FDs

* Sno, ProjectId --> WorkDesc

**Tables in 3NF:**

**DEPARTMENT:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dno** | **Dname** | **Office** | **Phone** | **Mail** | **HeadSSN** |

**FACULTY:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **SSN** | **NetID** | **Fname** | **Lname** | **Minit** | **Office** | **Sex** | **Bdate** |
| **mail** | **Phone** | **Street** | **City** | **State** | **Zip** | **Dno** |  |

**STUDENT:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sno** | NetID | Fname | Lname | Minit | Sex | Bdate | Class |
| mail | Phone | Street | City | State | Zip | Dno |  |

**COURSE:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cno** | Cname | Level | Credits | Dno | Description |

**PRE\_REQ:**

|  |  |
| --- | --- |
| **MainCno** | **PreReqCno** |

**SECTION:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SectionId** | Cno | SectionNo | Semester | Year | InstSSN | Size | Status | TASno | Description |

**SectionId is surrogate key.**

**SECTION\_TIMES:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SectionId** | **Day** | **FromTime** | ToTime | Location |

**GRADE\_REPORT:**

|  |  |  |
| --- | --- | --- |
| **Sno** | **SectionId** | Grade |

**ALUMNI:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NetID** | Fname | Lname | Minit | Sex | DOB | mail | Phone | Street | City | State | Zip |

**ALUMNI\_DEGREE:**

|  |  |  |  |
| --- | --- | --- | --- |
| **NetId** | **Degree** | Major | Year |

**PROJECT:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Id** | **Project Name** | Cordinator SSN | StartDate | EndDate | Dno | Description |

**PROJECT\_ STUDENT:**

|  |  |  |
| --- | --- | --- |
| **Sno** | **ProjectId** | WorkDesc |

**DATA MODEL:**

**DEPARTMENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Null** | **Key** | **Description** |
| Dno | int(3) | NO | PRI | A unique identifier of the department in the system |
| Dname | varchar(15) | NO | UNI | Name of the department, must be unique |
| Office | varchar(5) | YES |  | Location of departments office |
| Mail | varchar(15) | NO |  | Departments mail id, can be different from heads mail id |
| Phone | int(10) | YES |  | Departments office phone |
| HeadSSN | int(9) | YES | UNI | Deans SSN of the departments |

**FACULTY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Null** | **Key** | **Description** |
| SSN | int(9) | NO | PRI | A unique identifier of the Faculty in the system |
| NetID | varchar(9) | NO | UNI | Every person has unique netID , which can be username to access system. |
| Fname | varchar(15) | NO |  | First name |
| Minit | char(1) | YES |  | Initial of the middle name |
| Lname | varchar(15) | NO |  | Last name |
| Sex | char(1) | YES |  | M-Male  F-Female  O-Other |
| Bdate | Date | YES |  | Birth Date |
| Office | varchar(5) | YES |  | Office location of faculty |
| Mail | varchar(15) | NO |  | Email id |
| Phone | int(10) | YES |  | Phone number |
| Street | varchar(10) | YES |  | Street name and Apartment number of the address |
| City | varchar(10) | YES |  | City of the address |
| State | varchar(10) | YES |  | State of the address |
| Zip | int(5) | YES |  | Zip code of the address |
| Dno | int(3) | NO | FOR | Identifies the department to which this faculty belongs |

**STUDENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Null** | **Key** | **Description** |
| Sno | int(10) | NO | PRI | A unique identifier of the Student in the system |
| NetID | varchar(9) | NO | UNI | Every person has unique netID , which can be username to access system. |
| Fname | varchar(15) | NO |  | First name |
| Minit | char(1) | YES |  | Initial of the middle name |
| Lname | varchar(15) | NO |  | Last name |
| Sex | char(1) | YES |  | M-Male  F-Female  O-Other |
| Class | int(1) | NO |  | 1-Freshman  2-Sophomore  3- Junior  4- Senior  5-Graduate  6-Doctarate |
| Bdate | date | YES |  | Birth Date |
| mail | varchar(25) | NO |  | Email id |
| Phone | int(10) | YES |  | Phone number |
| Street | varchar(10) | YES |  | Street name and Apartment number of the address |
| City | varchar(10) | YES |  | City of the address |
| State | varchar(10) | YES |  | State of the address |
| Zip | int(5) | YES |  | Zip code of the address |
| Dno | int(3) | NO | FOR | Identifies the department to which this student belongs |

**COURSE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Null** | **Key** | **Description** |
| Cno | int(4) | NO | PRI | A unique identifier of the course. |
| Cname | varchar(25) | NO | UNI | A unique name of the course. |
| Level | int(4) | NO |  | Level of the course.  5000 - Undergraduate  6000 - Graduate level  7000 - Doctoral |
| Credits | int(1) | NO |  | Number of credits associated with the course |
| Dno | int(3) | NO |  | Identifies the department to which this course belongs |
| Description | varchar(50) | YES |  | Course description. |

**PRE\_REQ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Null** | **Key** | **Description** |
| MainCno | int(4) | NO | PRI,FOR | Main Course Number, refers Cno from course table |
| PreRecCno | int(4) | NO | PRI, FOR | Pre-requisite course number, refers Cno from course table |

**SECTION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Null** | **Key** | **Description** |
| SectionId | int(5) | NO | PRI | A unique identifier of the SectionId. It is system surrogate key |
| Cno | int(4) | NO | FOR | Identifies the course of section |
| SectionNo | int(3) | NO |  | Section number e.g 501,301 |
| Semester | char(1) | NO |  | Semester of the section  F-Fall  S-Spring  U-Summer |
| Year | int(4) | NO |  | Year of the section |
| InstSSN | int(9) | NO | FOR | Identifies faculty who teaches this course |
| Size | int(3) | NO |  | Size of the class.  Default value is 60 |
| Status | char(1) | NO |  | Status of the class  O - Open  C - Closed  Default : O |
| TASno | int(10) | YES |  | Identifies student who works as a Teaching Assistant for this course if any |
| Description | varchar(50) | YES |  | Description if any |

**SECTION\_TIMES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Null** | **Key** | **Description** |
| SectionId | int(5) | NO | PRI,FOR | Identifies Section . |
| Location | varchar(5) | YES |  | Classrooms location. |
| Day | varchar(5) | NO | PRI | Day e.g. Mon,Tues,Thur,etc |
| fromTime | Time | NO | PRI | Class start time in 24Hr format e.g 12:30, 18:30 |
| toTime | Time | NO |  | Class ends time in 24Hr format e.g 12:30, 18:30 |

**GRADE\_REPORT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Null** | **Key** | **Description** |
| Sno | int(10) | NO | PRI,FOR | Identifies Student from student table |
| SectionId | int(5) | NO | PRI,FOR | Identifies Section from section table |
| Grade | varchar(2) | YES |  | Grade achieved by student , null value means student registered but not graded yet. |

**ALUMNI**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Null** | **Key** | **Description** |
| NetID | varchar(9) | NO | PRI | Every alumni has unique netID |
| Fname | varchar(15) | NO |  | First name |
| Minit | char(1) | YES |  | Initial of the middle name |
| Lname | varchar(15) | NO |  | Last name |
| Sex | char(1) | YES |  | M-Male  F-Female  O-Other |
| Bdate | Date | YES |  | Birth Date |
| mail | varchar(15) | NO |  | Email id |
| Phone | int(10) | YES |  | Phone number |
| Street | varchar(10) | YES |  | Street name and Apartment number of the address |
| City | varchar(10) | YES |  | City of the address |
| State | varchar(10) | YES |  | State of the address |
| Zip | int(5) | YES |  | Zip code of the address |

**ALUMNI\_DEGREE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Null** | **Key** | **Description** |
| NetId | varchar(9) | NO | PRI,FOR | |  |  | | --- | --- | | Identifies the alumni to which this degree belongs, (i.e., the owner alumni). |  | |
| Degree | varchar(15) | NO | PRI | Degree title |
| Major | varchar(15) | YES |  | Major of the degree |
| Year | int(4) | NO |  | Year of the degree |

**PROJECT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Null** | **Key** | **Description** |
| ProjectId | int(10) | NO | PRI | A unique identifier of the project. |
| ProjectName | varchar(50) | NO |  | Name of the project |
| CordinatorSSN | int(9) | YES | FOR | Identifies the project coordinator. refers to SSN of faculty table |
| StartDate | date | YES |  | Projects Start date |
| EndDate | date | YES |  | Projects End date |
| Description | varchar(50) | YES |  | Projects Description |
| Dno | int(3) | NO | FOR | Identifies the department to which this faculty belongs |

**PROJECT\_ STUDENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Null** | **Key** | **Description** |
| Sno | int(10) | NO | PRI,FOR | Identifies the student who works on the project . refers to Sno of student table |
| ProjectId | int(10) | NO | PRI,FOR | Identifies the project. refers to ProjectId of project table |
| WorkDesc | varchar(50) | YES |  | Work description if any |

**SQL Commands to create TABLE and CONSTRAINT**

**DEPARTMENT**

CREATE TABLE DEPARTMENT(

Dno **INT**(**3**) PRIMARY KEY,

Dname **VARCHAR**(**15**) NOT NULL UNIQUE,

Office **VARCHAR**(**5**),

Mail **VARCHAR**(**25**) NOT NULL,

Phone **INTEGER**(**10**),

HeadSSN **INTEGER**(**9**) NOT NULL UNIQUE);

ALTER TABLE DEPARTMENT ADD CONSTRAINT FK\_DEPARTMENT FOREIGN KEY(HEADSSN) REFERENCES FACULTY(SSN);

**STUDENT**

CREATE TABLE STUDENT(

Sno **INTEGER**(**10**) PRIMARY KEY AUTO\_INCREMENT ,

NetID **VARCHAR**(**9**) NOT NULL UNIQUE,

Fname **VARCHAR**(**15**) NOT NULL,

Minit CHAR,

Lname **VARCHAR**(**15**) NOT NULL,

Sex CHAR,

Class **INT**(**1**) NOT NULL,

Bdate **DATE**,

mail **VARCHAR**(**25**) NOT NULL,

Phone **INTEGER**(**10**),

Street **VARCHAR**(**10**),

City **VARCHAR**(**10**),

State **VARCHAR**(**10**),

Zip **INT**(**5**),

Dno **INT**(**3**) NOT NULL);

ALTER TABLE STUDENT AUTO\_INCREMENT=**1000**;

ALTER TABLE STUDENT ADD CONSTRAINT FK\_DEPT FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dno);

**FACULTY**

CREATE TABLE FACULTY(

SSN **INTEGER**(**9**) PRIMARY KEY,

NetID **VARCHAR**(**9**) NOT NULL UNIQUE,

Fname **VARCHAR**(**15**) NOT NULL,

Minit CHAR,

Lname **VARCHAR**(**15**) NOT NULL,

Sex CHAR,

Bdate **DATE**,

Office **VARCHAR**(**5**),

Mail **VARCHAR**(**25**) NOT NULL,

Phone **INTEGER**(**10**),

Street **VARCHAR**(**10**),

City **VARCHAR**(**10**),

State **VARCHAR**(**10**),

Zip **INT**(**5**),

Dno **INT**(**3**) NOT NULL);

ALTER TABLE FACULTY ADD CONSTRAINT FK\_FACULTY FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dno);

**COURSE**

CREATE TABLE COURSE(

Cno **INTEGER**(**4**) PRIMARY KEY,

Cname **VARCHAR**(**25**)NOT NULL UNIQUE,

Level **INTEGER**(**4**)NOT NULL,

Credits **INTEGER**(**1**)NOT NULL,

Dno **INT**(**3**) NOT NULL,

Description **VARCHAR**(**50**));

ALTER TABLE COURSE ADD CONSTRAINT FK\_COURSE FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dno);

**PRE\_REQ**

CREATE TABLE PRE\_REQ(

MainCno **INTEGER**(**4**) NOT NULL,

PreRecCno **INTEGER**(**4**) NOT NULL,

PRIMARY KEY (MainCno,PreRecCno));

ALTER TABLE PRE\_REQ ADD CONSTRAINT FK\_PRE\_REQ FOREIGN KEY (MainCno) REFERENCES COURSE(Cno);

ALTER TABLE PRE\_REQ ADD CONSTRAINT FK\_PRE\_REQ1 FOREIGN KEY (PreRecCno) REFERENCES COURSE(Cno);

**SECTION**

CREATE TABLE SECTION(

SectionId **INTEGER**(**5**) PRIMARY KEY AUTO\_INCREMENT,

Cno **INTEGER**(**4**) NOT NULL,

SectionNo **INTEGER**(**3**) NOT NULL,

Semester CHAR NOT NULL,

Year **INTEGER**(**4**) NOT NULL,

InstSSN **INTEGER**(**9**) NOT NULL,

Size **INTEGER**(**3**) NOT NULL DEFAULT **60**,

Status CHAR NOT NULL DEFAULT 'O',

TASno **INTEGER** (**10**),

Description **VARCHAR**(**50**),

UNIQUE (Cno,SectionNo,Semester,**Year**));

ALTER TABLE SECTION AUTO\_INCREMENT=**80000**;

ALTER TABLE SECTION ADD CONSTRAINT FK\_SECTION FOREIGN KEY (Cno) REFERENCES COURSE(cno);

ALTER TABLE SECTION ADD CONSTRAINT FK\_SECTION1 FOREIGN KEY (InstSSN) REFERENCES FACULTY(SSN);

ALTER TABLE SECTION ADD CONSTRAINT FK\_SECTION2 FOREIGN KEY (TASno) REFERENCES STUDENT(Sno);

**SECTION\_TIMES**

CREATE TABLE SECTION\_TIMES(

SectionId **INTEGER**(**5**) NOT NULL,

Location **VARCHAR**(**5**),

Day **VARCHAR**(**5**),

fromTime time,

toTime time,

PRIMARY KEY(SectionId,Day,fromTime));

ALTER TABLE SECTION\_TIMES ADD CONSTRAINT FK\_SECTION\_TIMES FOREIGN KEY (SectionId) REFERENCES SECTION(SectionId) ON DELETE CASCADE;

**GRADE\_REPORT**

CREATE TABLE GRADE\_REPORT(

Sno **INTEGER**(**10**) NOT NULL,

SectionId **INTEGER**(**5**) NOT NULL,

Grade **VARCHAR**(**2**),

PRIMARY KEY (Sno, SectionId));

ALTER TABLE GRADE\_REPORT ADD CONSTRAINT FK\_GRADE\_REPORT FOREIGN KEY (Sno) REFERENCES STUDENT(Sno);

ALTER TABLE GRADE\_REPORT ADD CONSTRAINT FK\_GRADE\_REPORT1 FOREIGN KEY (SectionId) REFERENCES SECTION (SectionId);

**ALUMNI**

CREATE TABLE ALUMNI(

NetID **VARCHAR**(**9**) PRIMARY KEY,

Fname **VARCHAR**(**15**) NOT NULL,

Minit CHAR,

Lname **VARCHAR**(**15**) NOT NULL,

Sex CHAR NOT NULL,

Bdate **DATE**,

Mail **VARCHAR**(**25**) NOT NULL,

Phone **INTEGER**(**10**),

Street **VARCHAR**(**10**),

City **VARCHAR**(**10**),

State **VARCHAR**(**10**),

Zip **INT**(**5**));

ALTER TABLE ALUMNI ADD CONSTRAINT FK\_ALUMNI FOREIGN KEY (NetID) REFERENCES STUDENT(NetID);

**ALUMNI\_DEGREE**

CREATE TABLE ALUMNI\_DEGREE(

NetId **VARCHAR**(**9**) NOT NULL ,

Degree **VARCHAR**(**15**) NOT NULL,

Major **VARCHAR**(**15**),

**Year** **INTEGER**(**4**) NOT NULL,

PRIMARY KEY(NetId, Degree));

ALTER TABLE ALUMNI\_DEGREE ADD CONSTRAINT FK\_ALUMNI\_DEGREE FOREIGN KEY (NetID) REFERENCES ALUMNI(NetID) ON DELETE CASCADE;

**PROJECT**

CREATE TABLE PROJECT(

ProjectId **INTEGER**(**10**) PRIMARY KEY,

ProjectName **VARCHAR**(**50**) NOT NULL,

CordinatorSSN **INTEGER**(**9**),

StartDate **DATE**,

EndDate **DATE**,

Dno **INT**(**3**) NOT NULL

Description **VARCHAR**(**50**) );

ALTER TABLE PROJECT ADD CONSTRAINT FK\_PROJECT FOREIGN KEY (CordinatorSSN) REFERENCES FACULTY(SSN);

ALTER TABLE PROJECT ADD CONSTRAINT FK\_ PROJECT1 FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dno);

**PROJECT\_ STUDENT**

CREATE TABLE PROJECT\_STUDENT(

Sno **INTEGER**(**10**),

ProjectId **INTEGER**(**10**),

WorkDesc **VARCHAR**(**50**),

PRIMARY KEY (Sno,ProjectId));

ALTER TABLE PROJECT\_STUDENT ADD CONSTRAINT FK\_PROJECT\_STUDENT FOREIGN KEY (Sno) REFERENCES STUDENT(Sno) ON DELETE CASCADE;

ALTER TABLE PROJECT\_STUDENT ADD CONSTRAINT FK\_PROJECT\_STUDENT1 FOREIGN KEY (ProjectId) REFERENCES PROJECT(ProjectId) ON DELETE CASCADE;

ALTER TABLE PROJECT\_STUDENT DROP FOREIGN KEY FK\_PROJECT\_STUDENT;

ALTER TABLE PROJECT\_STUDENT DROP FOREIGN KEY FK\_PROJECT\_STUDENT1;

**VIEWS**

**SECTION\_DETAILS**

create VIEW section\_details as select s.SectionId,s.Semester,s.**Year**,s.SectionNo,s.Size,s.Status,s.Description,

c.Cname,c.Cno,f.SSN,f.Fname,f.Lname,t.Day,t.fromTime,t.toTime,t.Location

from section s,course c,faculty f,section\_times t

where s.cno = c.cno and s.InstSSN = f.SSN and s.SectionId = t.SectionId

**TRANSCRIPT**

create VIEW Transcript as

select G.Sno, G.SectionId, C.Cname, C.Credits, G.Grade,S.Semester, S.**Year**

from grade\_report G,section S,course C

where G.SectionId = S.SectionId AND S.Cno = C.Cno AND G.Grade IS NOT NULL

order by G.Sno

**SQL commands to drop all Tables and their Constraints:**

ALTER TABLE PROJECT\_STUDENT DROP FOREIGN KEY FK\_PROJECT\_STUDENT;

ALTER TABLE PROJECT\_STUDENT DROP FOREIGN KEY FK\_PROJECT\_STUDENT1;

ALTER TABLE PROJECT DROP FOREIGN KEY FK\_ PROJECT;

ALTER TABLE PROJECT DROP FOREIGN KEY FK\_ PROJECT1;

ALTER TABLE GRADE\_REPORT DROP FOREIGN KEY FK\_GRADE\_REPORT;

ALTER TABLE GRADE\_REPORT DROP FOREIGN KEY FK\_GRADE\_REPORT1;

ALTER TABLE FACULTY DROP FOREIGN KEY FK\_FACULTY;

ALTER TABLE DEPARTMENT DROP FOREIGN KEY FK\_DEPARTMENT;

ALTER TABLE STUDENT DROP FOREIGN KEY FK\_DEPT;

ALTER TABLE PRE\_REQ DROP FOREIGN KEY FK\_PRE\_REQ;

ALTER TABLE PRE\_REQ DROP FOREIGN KEY FK\_PRE\_REQ1;

ALTER TABLE ALUMNI\_DEGREE DROP FOREIGN KEY FK\_ALUMNI\_DEGREE;

ALTER TABLE ALUMNI DROP FOREIGN KEY FK\_ALUMNI;

ALTER TABLE SECTION\_TIMES DROP FOREIGN KEY FK\_SECTION\_TIMES;

ALTER TABLE SECTION DROP FOREIGN KEY FK\_SECTION;

ALTER TABLE SECTION DROP FOREIGN KEY FK\_SECTION1;

ALTER TABLE SECTION DROP FOREIGN KEY FK\_SECTION2;

ALTER TABLE COURSE DROP FOREIGN KEY FK\_COURSE;

DROP TABLE ALUMNI\_DEGREE;

DROP TABLE ALUMNI;

DROP TABLE PROJECT\_STUDENT;

DROP TABLE PROJECT;

DROP TABLE GRADE\_REPORT;

DROP TABLE SECTION\_TIMES;

DROP TABLE SECTION;

DROP TABLE PRE\_REQ;

DROP TABLE COURSE;

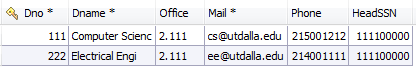
DROP TABLE STUDENT;

DROP TABLE FACULTY;

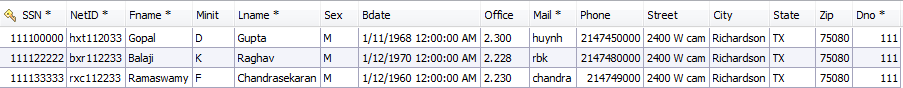
DROP TABLE DEPARTMENT;

**DATA STORED IN ALL TABLES**

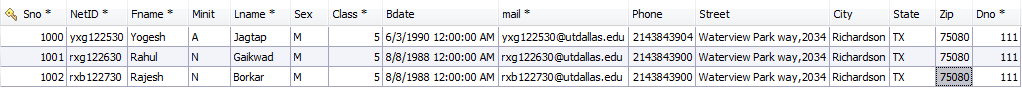
**DEPARTMENT**

****

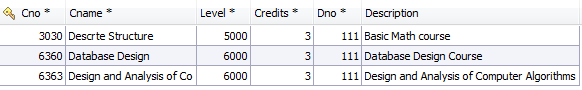
**FACULTY**

****

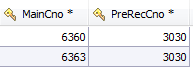
**STUDENT**

****

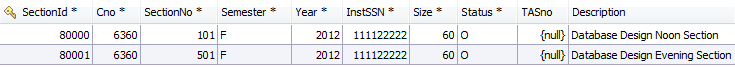
**COURSE**

****

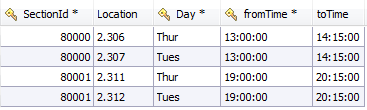
**PRE\_REQ**

****

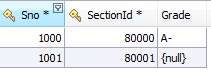
**SECTION**

****

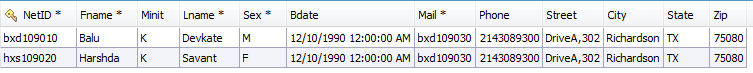
**SECTION\_TIMES**

****

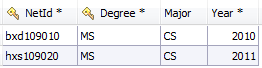
**GRADE\_REPORT**

****

**ALUMNI**

****

**ALUMNI\_DEGREE**

****

**PROJECT**

****

**PROJECT\_ STUDENT**



**Most frequently used SQL queries:**

**Query to retireve instructor ssn from faculty table**

select SSN from faculty where Fname like '%Balaji%' OR Lname like '%Balaji%'

**Query to retireve sectionid from section times**

select sectionId from section\_times where ('18:30') <= TIME\_TO\_SEC(fromTime) and TIME\_TO\_SEC('20:30') >= TIME\_TO\_SEC(toTime)

**Query to retrieve student fname and lname from sno**

select Fname,Minit,Lname from student where Sno = 8001

**Query to retrieve sections information**

select distinct s.Semester,s.**Year**,s.SectionId,c.Cname,s.Cno,s.SectionNo,

f.Fname,f.Lname,s.Size,s.Status,s.Description

from section s,course c,faculty f,section\_times t

where s.cno = c.cno and s.InstSSN = f.SSN and s.SectionId = t.SectionId

AND s.Cno = **6360** AND **Year** = **2012** AND Semester = 'F'

order by s.**Year** desc, s.Semester,s.Cno,s.SectionNo

**Query to insert into grade report(new registration)**

Insert into grade\_report(SectionId,Sno) VALUES (**80000**,**8001**)

**Query to delete from grade report(cancel registration)**

Delete from grade\_report where SectionId = **80000** and Sno = **8001**

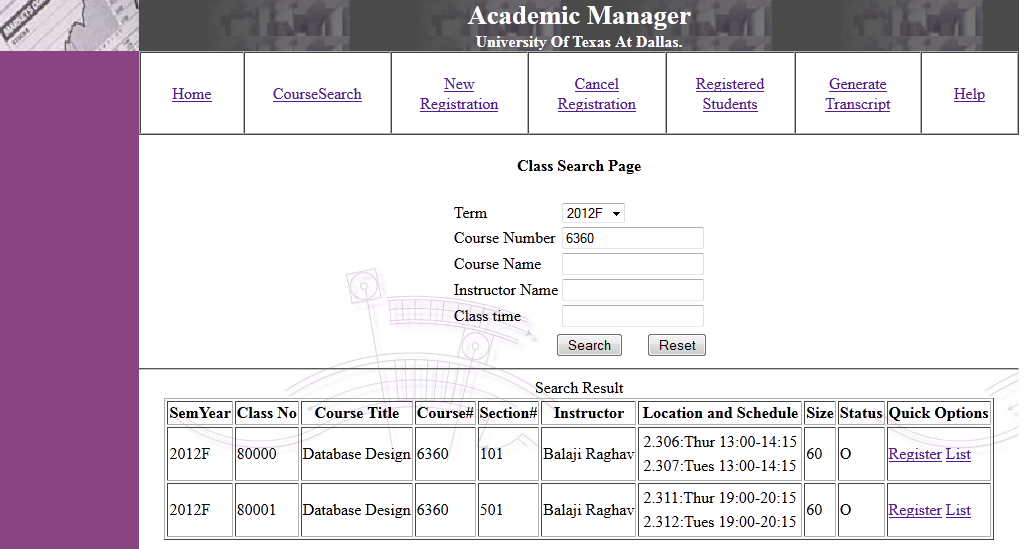
**Query to select from grade report(generate transcript)**

select G.SectionId, G.Grade, C.Cname, C.Credits,S.Semester,S.**Year**

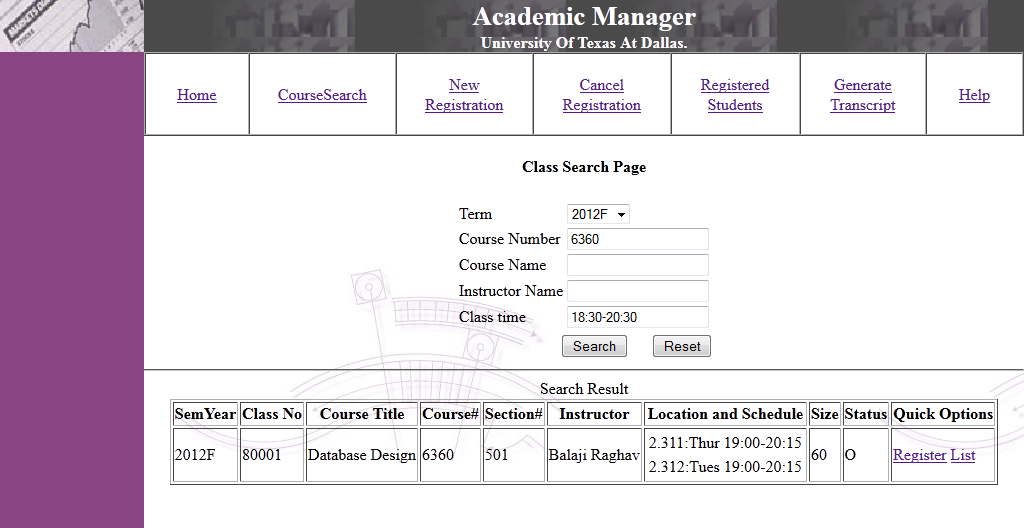
from grade\_report G,section S,course C

where G.SectionId = S.SectionId AND S.Cno = C.Cno And Sno = **1000**

* **Graphical screen to Search section by Course Number**



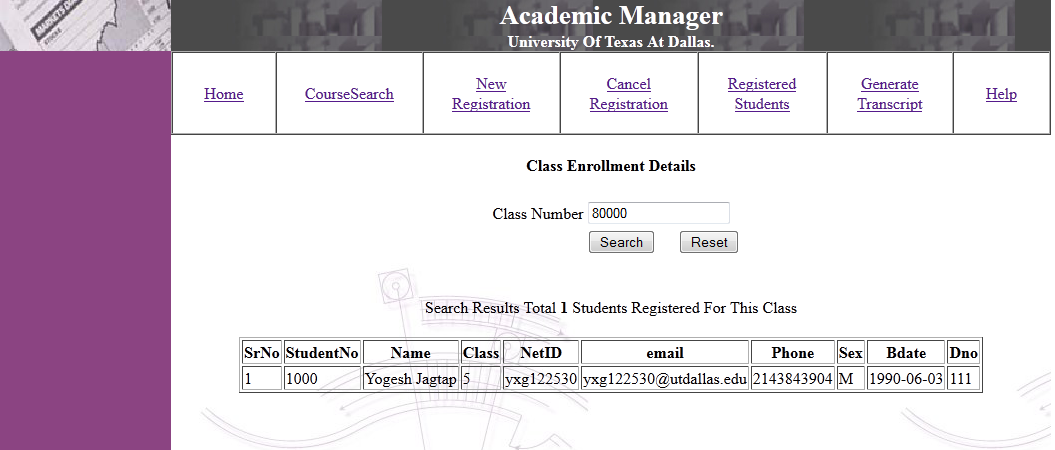
* **Graphical screen to Search section by Course Number and Class time**



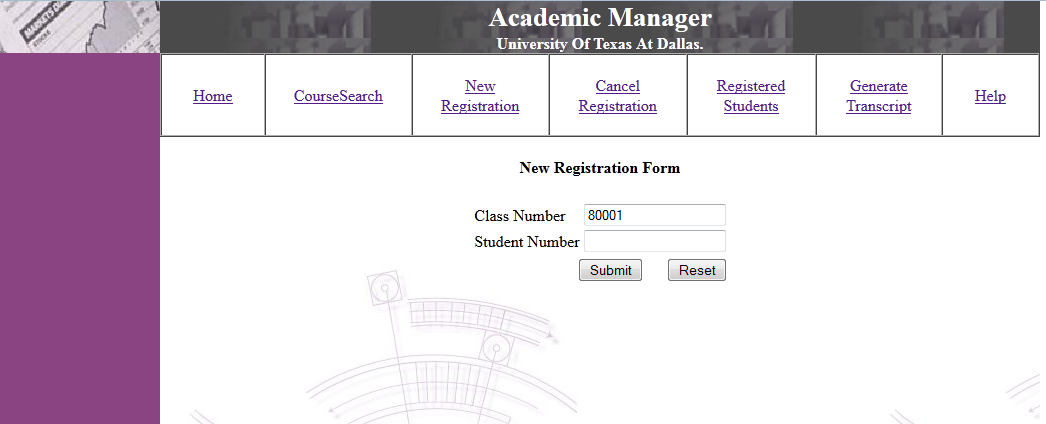
* **Graphical screen to Search section by Instructor Name**



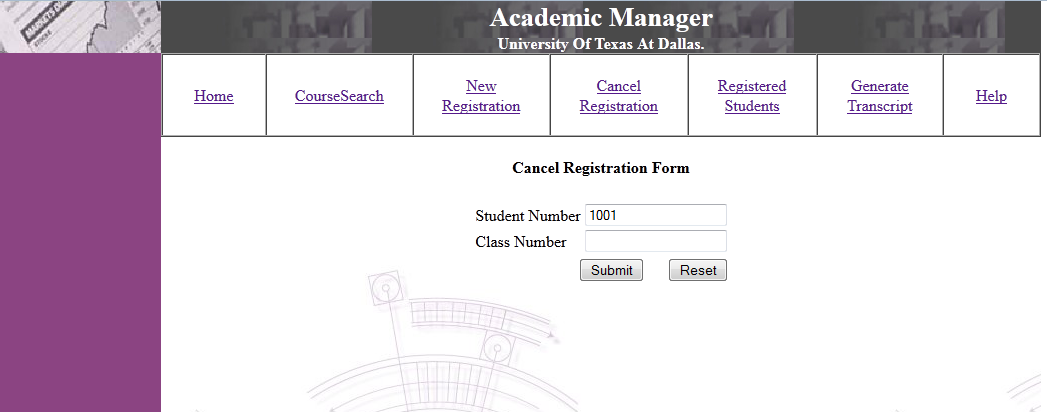
* **Graphical screen to List Down Students of the class**

****

* **Graphical screen For New Class registration**



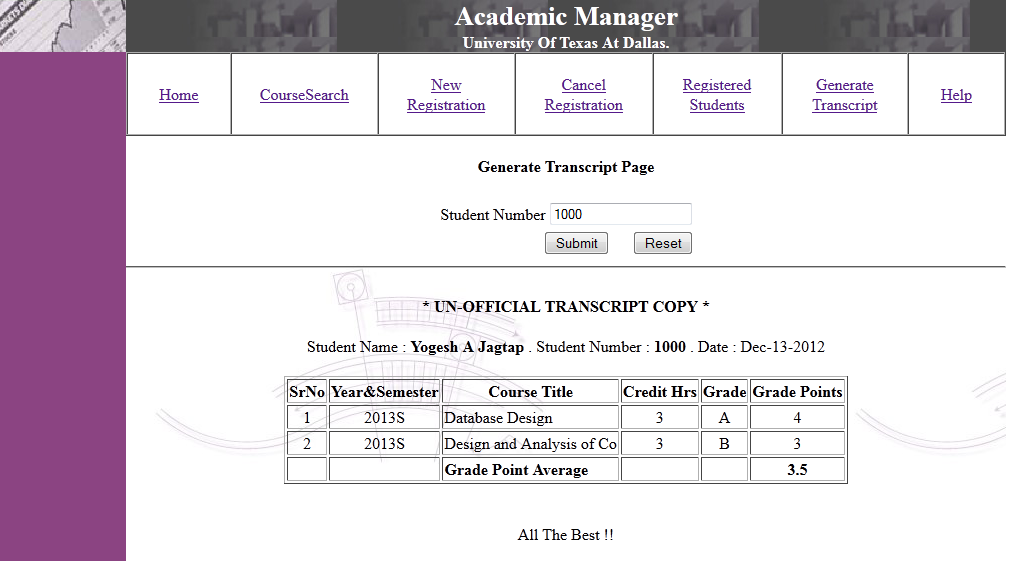
* **Graphical screen For Cancel Class registration**



**Extra Credit-Work:**

The student can generate transcript. Transcript will contain all the courses student attended, grade received and Grade Point Average(GPA) .

* **Graphical screen To Generate Transcript**



**Good point of the design:**

All tables in the design are well decomposed into 3NF. Insertion/updating/deletion anomalies are well prevented.

Access path i.e. primary indexes are created for fast data access.

Views are used wherever data frequently accessed from multiple tables using joins.

**Minus point of the design:**

Security issues like injection attack are not prevented by the system.

Queries are not optimized.

**Conclusion:**

The implemented academic manager system allows web based access to search courses. Student can search classes depends on search criteria. Student can register or drop courses. Student view and print his transcript. Instructor can view list of the entire student registered in the class.

**Future Scope:**

The system should implement user authentication module in the future release. Security issues like SQL injection attack and encryption must be handled by the system. System log must be created in order to monitor system.