**Student Information Management System**

**Objectives:**

The main objective of this project is to create an efficient database for tracking student performance by storing student results, student course enrollments and few other student details. It can be used to generate a term-wise students’ performance reports. It can also be used to manage the details of profiles, courses, logins, exams, fees, assignments. Using our system we add or edit courses, terms, and new student enrollments. This project is useful for administrative purposes for getting and storing the results in a simple manner.

**Scope:**

Our database mainly consists of the following tables: Student, student address, Student enrolment, Department,Faculty, Courses, Terms, sections, and Student Financials details. This complete database was built to make administrative tasks easier and more efficient. Our database provides an easy-to-use interface for the students to retrieve information in a timely and reliable manner The project provides facilities like online registration and profile creation of students thus reducing paperwork and automating the record generation process in an educational institution.

**User Requirements:**

* The database must able to register new students.
* Make changes to the student information.
* Look for a specific student or a group of students based on Student ID, department, courses enrolled.
* It should add and modify the Term details.
* Record the course details and subject information.
* Make changes to the course details.
* Keep track or modify the student internal grades.
* Register a employee or teacher.
* It should maintain details student Payments.

**Business Rules:**

1) A Student can enroll in multiple Sections.

2) Each Section can have multiple Students enroll.

3) Each Faculty is assigned to multiple Sections.

4) Each Section is assigned to exactly one Faculty person.

5) A Term can have multiple Courses.

6) Each Course is belongs to particular Term.

7) A Student must have one Student Payment Details record.

8) Each Student Payment Details record belongs to exactly one student.

9) A Student have one Student Address.

10) Each Student Address belongs to exactly one record.

11) A Department can have one or more Faculties.

12) Each Faculty belongs to only one Department.

13) Every department can have one or more Students.

14) Each student belongs to only one Department.

15) Each Course can have one or more Sections.

16) Every Section have only one Course.

17) A Department can offer multiple Courses.

18) Each Course should belong to exactly one Department

**ERD Diagram:**

Diagram

Description automatically generated

**Data Directory:**

**Table: Student**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attribute name | contents | Data types | Format | Range | Required | Pk/fk | Fk required table |
| Student\_id | Unique id is assigned to each student | Int | 99999 | 0-99999 | Y | pk |  |
| First\_name | First name of the student | VARCHAR(30) | XXXXXXX |  | Y |  |  |
| Last\_name | Last name of the student | VARCHAR(30) | XXXXXXX |  | Y |  |  |
| Email | Email of the student | VARCHAR(30) | XXXXXXX |  | Y |  |  |
| Birth\_date | Date of birth of the student | DATE | yyyy-mm-dd |  | Y |  |  |
| Gender | Gender of the student | VARCHAR(6) | xxxxx |  |  |  |  |
| Total\_grade | Total grade of the student | INT | 9 |  |  |  |  |
| Department\_id | Department ID of the student | INT | 99999 |  | Y | FK | Department |

**Table: Student Payment Details**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Constraint | Data Type | format | range | Required | Pk/fk | Fk required table |
| Student\_id | Unique id is assigned to each student **t** | INT | 99999 | 0-99999 | Y | Pk,fk | student |
| Account\_holder\_name | Name of the Account holder | VARCHAR(30) | Xxxxxxxxx |  | Y |  |  |
| Payment\_type | Payment type (credit or debit) | VARCHAR(10) | Xxxxxxxxx |  | Y |  |  |
| Card\_number | Card number for payment | VARCHAR(30) | Xxxxxxxxx |  | Y |  |  |
| Expiry\_date | Date of expiry of the card | DATE | yyyy-mm-dd |  | Y |  |  |
| Cvv | Cvv number of the card | INT | 999 | 111-999 | Y |  |  |
| Inserted\_date | Inserted date of the payment detail | DATE | yyyy-mm-dd |  | Y |  |  |
| Updated\_date | Updated date of the payment detail | DATE | yyyy-mm-dd |  | Y |  |  |

**Table: Student Address**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attribute name | contents | Data types | Format | Range | Required | Pk/fk | Fk required table |
| Student\_id | Unique id is assigned to each student **t** | INT | 99999 | 0-99999 | Y | Pk,fk | student |
| Address1 | Address line1 of the student | VARCHAR(30) | xxxxxx |  | Y |  |  |
| Address2 | Address line2 of the student | VARCHAR(30) | Xxxxxx |  |  |  |  |
| city | City of the student | VARCHAR(30) | xxxxx |  | Y |  |  |
| State | State of the student | VARCHAR(30) | Xxxxx |  | Y |  |  |
| zip | Zip code of the student | INTEGER | 99999 | 0-99999 | Y |  |  |
| Phone\_number | Phone number of the student | VARCHAR(10) | xxxxxxxx |  | Y |  |  |
| Inserted\_date | Inserted date of the student detail | DATE | yyyy-mm-dd |  | Y |  |  |
| Updated\_date | Updated date of the student detail | DATE | yyyy-mm-dd |  | Y |  |  |

**Table: Faculty**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attribute name | contents | Data types | Format | Range | Required | Pk/fk | Fk required table |
| Faculty\_id | Unique id is assigned to each faculty | INT | 99999 | 0-99999 | Y | pk |  |
| First\_name | First name of the faculty | VARCHAR(30) | Xxxxxxxx |  | Y |  |  |
| Last\_name | Last name of the faculty | VARCHAR(30) | Xxxxxxxxx |  | Y |  |  |
| Email | Email of the faculty | VARCHAR(30) | xxxxxxxx |  | Y |  |  |
| Birth\_date | Date of birth of the faculty | DATE | yyyy-mm-dd |  |  |  |  |
| Gender | Gender of the faculty | VARCHAR(6) | xxxxxx |  |  |  |  |
| Phone\_number | Phone number of the faculty | VARCHAR(10) | Xxxxxxxxx |  | Y |  |  |
| Department\_id | Department ID of the faculty | INT | 99999 | 0-99999 | Y | Fk | Department |

**Table: Department**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attribute name | contents | Data types | Format | Range | Required | Pk/fk | Fk required table |
| Department\_id | Unique id is assigned to each faculty | INT | 99999 | 0-99999 | Y | pk |  |
| Department\_name | Name of the department | VARCHAR(150) | xxxxxxxx |  | Y |  |  |

**Table: Term**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attribute name | contents | Data types | Format | Range | Required | Pk/fk | Fk required table |
| Term\_id | A unique Id assigned to each Term | INT | 99999 | 0-99999 | Y | PK |  |
| Term\_name | Name of the term | VARCHAR(30) | xxxxxxx |  | Y |  |  |
| Start\_date | Term Start date | DATE | yyyy-mm-dd |  | Y |  |  |
| End\_date | Term End date | DATE | yyyy-mm-dd |  | Y |  |  |

**Table: Course**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attribute name | contents | Data types | Format | Range | Required | Pk/fk | Fk required table |
| Course\_id | A unique Id assigned to each course | INT | 99999 | 0-99999 | Y | pk |  |
| Course\_name | Name of the Course | VARCHAR(150) | Xxxxxxx |  | Y |  |  |
| Course\_code | Course code | INT | 99999 | 0-99999 | Y |  |  |
| Course\_credit | Course credit hours | INT | 9 | 0-4 | Y |  |  |
| Department\_id | Department to which course belongs | INT | 99999 | 0-99999 | Y | FK | Department |
| Term\_id | Term id of the course | INT | 99999 | 0-99999 | Y | FK | Term |
| Start\_date | Course start date | DATE | yyyy-mm-dd |  | Y |  |  |
| End\_date | Course end date | DATE | yyyy-mm-dd |  | Y |  |  |

**Table: Section**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attribute name | contents | Data types | Format | Range | Required | Pk/fk | Fk required table |
| Section\_id | A unique id assigned for each sectiom | INT | 99999 | 0-99999 | Y | pk |  |
| Section\_name | Name of the section | VARCHAR(30) | Xxxxxx |  | Y |  |  |
| Total\_seats | Total number of seats | INT | 999 | 0-999 | Y |  |  |
| Course\_id | Course id of the section | INT | 99999 | 0-99999 | Y | FK | Course |
| Faculty\_id | Faculty id assign to the section | INT | 99999 | 0-99999 | Y | FK | Faculty |

**Table: Student Enrollment**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attribute name | contents | Data types | Format | Range | Required | Pk/fk | Fk required table |
| Enrollement\_id | A unique id assigned for each enrollement | INT | 99999 | 0-99999 | Y | pk |  |
| Student\_id | Student Id of the enrollment | INT | 99999 | 0-99999 | Y | FK | Student |
| Section\_id | Section Id of the enrollment | INT | 99999 | 0-99999 | Y | FK | Section |
| Enrolled\_date | Enrollment date | DATE | yyyy-mm-dd |  | Y |  |  |
| Grade | Grade of the student for the course enrolled | INT | 9 | 0-4 |  |  |  |

**Data Entry and Update**

1.**Creating department table**

CREATE TABLE Department(

department\_id INTEGER,

departmentName varchar(150) NOT NULL,

CONSTRAINT departmentId\_pk PRIMARY KEY (department\_id)

);

**Data entry into department table**

INSERT INTO Department VALUES

(1, 'Department of Computer Science and Engineering'),

(2, 'Department of Accounting'),

(3, 'Department of Finance'),

(4, 'Department of Information Technology'),

(5, 'Department of Logistics and Operations Management'),

(6, 'Department of Management'),

(7, 'Department of Marketing'),

(8, 'Department of Counseling and Higher Education'),

(9, 'Department of Biomedical Engineering'),

(10, 'Department of Electrical Engineering'),

(11, 'Department of Mechanical Engineering'),

(12, 'Department of Materials Science and Engineering'),

(13, 'Department of Audiology and Speech-Language Pathology'),

(14, 'Department of Behavior Analysis'),

(15, 'Department of Criminal Justice');

Table

Description automatically generated with medium confidence

**2. Creating student table**

CREATE TABLE Student(

student\_id INTEGER,

first\_name varchar(30) NOT NULL,

last\_name varchar(30) NOT NULL,

email varchar(30) NOT NULL,

birth\_date Date NOT NULL,

gender varchar(6),

total\_grade INTEGER,

department\_id INTEGER NOT NULL,

CONSTRAINT student\_id\_pk PRIMARY KEY (student\_id),

CONSTRAINT department\_id\_fk FOREIGN KEY (department\_id) REFERENCES Department(department\_id));

**Data entry into student tables**

INSERT INTO Student VALUES

(1, 'ammie', 'kdad', 'ammie@unt.com', '1980-01-01', 'M', 4, 1),

(2, 'Joni', 'kldd', 'Joni@unt.com', '1980-03-03', 'M', 3, 2),

(3, 'candy', 'risd', 'candy@unt.com', '1980-01-02', 'F', 4, 3),

(4, 'Denny', 'ldsd', 'denny@unt.com', '1980-05-02', 'M', 4, 4),

(5, 'Eddy', 'lgfd', 'eddyl@unt.com', '1998-01-01', 'M', 4, 5),

(6, 'Fanny', 'pdsf', 'fannyp@unt.com', '1998-02-01', 'F', 4, 6),

(7, 'Gerry', 'sgef', 'gerrys@unt.com', '1998-03-01', 'M', 4, 7),

(8, 'Henny', 'sfd', 'henny@unt.com', '1998-04-01', 'M', 4, 8),

(9, 'terry', 'tdsfs', 'terry@unt.com', '1998-05-01', 'M', 4, 9),

(10, 'matthew', 'svsr', 'matthew@unt.com', '1998-06-01', 'M', 4, 10),

(11, 'cindy', 'TW', 'cindy@unt.com', '1998-07-01', 'F', 4, 11),

(12, 'nancy', 'zelif', 'nancy@unt.com', '1998-08-01', 'F', 4, 12),

(13, 'diana', 'immel', 'diana@unt.com', '1998-09-01', 'F', 4, 13),

(14, 'cari', 'fdsf', 'cari@unt.com', '1998-10-01', 'M', 4, 14),

(15, 'james', 'case', 'jamse@unt.com', '1998-11-01', 'M', 4, 15);

Graphical user interface

Description automatically generated with medium confidence

**3. Creating faculty table**

CREATE TABLE Faculty(

faculty\_id INTEGER,

first\_name varchar(30) NOT NULL,

last\_name varchar(30) NOT NULL,

email varchar(30) NOT NULL,

birth\_date Date NOT NULL,

gender varchar(6),

phone\_number varchar(10) NOT NULL,

department\_id INTEGER NOT NULL,

CONSTRAINT faculty\_id\_pk PRIMARY KEY (faculty\_id),

CONSTRAINT department\_id\_fk1 FOREIGN KEY (department\_id) REFERENCES Department(department\_id));

**Data entry into faculty tables**

INSERT INTO Faculty VALUES

(1, 'John', 'Doe', 'john\_doe@unt.com', '1980-01-01', 'M', '1234567890', 1),

(2, 'Joni', 'Adkis', 'Adkis@unt.com', '1980-02-01', 'F', '1234567890', 2),

(3, 'Ali' , 'AlAl', 'AlAl@unt.com', '1980-03-10', 'M', '1234567890', 3),

(4, 'Adem', 'Adsa', 'AjayA@unt.com', '1989-01-09', 'M', '1234567890', 4),

(5, 'Bobby', 'Bkie', 'Bobbyb@unt.com', '1990-01-10', 'M', '1234567890', 5),

(6, 'Candy', 'Cdad', 'candy@unt.com', '1995-01-11', 'F', '1234567890', 6),

(7, 'Denny', 'Dlls', 'denny@unt.com', '1996-01-01', 'M', '1234567890', 7),

(8, 'Eddy', 'Eda', 'eddy@unt.com', '1990-01-01', 'M', '1234567890', 8),

(9, 'Fanny', 'Fcds', 'fanny@unt.com', '1992-03-01', 'F', '1234567890', 9),

(10, 'Gerry', 'Gdsa', 'gerry@unt.com', '1993-04-02', 'M', '1234567890', 10),

(11, 'Henny', 'Hfds', 'henny@unt.com', '1995-01-01', 'M', '1234567890', 11),

(12, 'Izzy', 'Isds', 'izzy@unt.com', '1998-10-02', 'F', '1234567890', 12),

(13, 'Jenny', 'Jlmds', 'jenny@unt.com', '1999-12-12', 'F', '1234567890', 13),

(14, 'Kenny', 'Ksa', 'kenny@unt.com', '1998-09-12', 'M', '1234567890', 14),

(15, 'Lenny', 'Lvad', 'lenny@unt.com', '1982-01-01', 'M', '1234567890', 15);

Text

Description automatically generated with low confidence

**4. Creating student address table**

CREATE TABLE StudentAddress(

student\_id INTEGER,

address1 varchar(30) NOT NULL,

address2 varchar(30),

city varchar(30) NOT NULL,

state varchar(30) NOT NULL,

zip INTEGER NOT NULL,

phone\_number varchar(10) NOT NULL,

inserted\_date DATE NOT NULL,

updated\_date DATE NOT NULL,

CONSTRAINT student\_id\_pk1 PRIMARY KEY (student\_id),

CONSTRAINT student\_id\_fk1 FOREIGN KEY (student\_id) REFERENCES Student(student\_id));

**Data entry into student address table**

INSERT INTO StudentAddress VALUES

(1, '123 Main St', '', 'San Diego', 'CA', '92101', '1234567890', '2020-01-01', '2020-01-01'),

(2, '234 main St', '', 'kansas', 'KS', '92101', '1234567890', '2020-01-01', '2020-01-01'),

(3, '345 Main St', '', 'San Diego', 'CA', '92101', '1234567890', '2020-01-01', '2020-01-01'),

(4, '1115 N college dr', '', 'kansas', 'KS', '92101', '1234567890', '2020-01-01', '2020-01-01'),

(5, '123 Main St', '', 'San Diego', 'CA', '92101', '1234567890', '2020-01-01', '2020-01-01'),

(6, '215 Bonnie dr', '', 'kansas', 'KS', '92101', '1234567890', '2020-01-01', '2020-01-01'),

(7, '123 Main St', '', 'San Diego', 'CA', '92101', '1234567890', '2020-01-01', '2020-01-01'),

(8, '986 Main St', '', 'kansas', 'KS', '92101', '1234567890', '2020-01-01', '2020-01-01'),

(9, '232 bonnie dr', '', 'San Diego', 'CA', '92101', '1234567890', '2020-01-01', '2020-01-01'),

(10, '123 Main St', '', 'San Diego', 'CA', '92101', '1234567890', '2020-01-01', '2020-01-01'),

(11, '21 Main St', '', 'kansas', 'KS', '92101', '1234567890', '2020-01-01', '2020-01-01'),

(12, '77 Bonnie dr', '', 'San Diego', 'CA', '92101', '1234567890', '2020-01-01', '2020-01-01'),

(13, '123 St', '', 'San Diego', 'MO', '92101', '1234567890', '2020-01-01', '2020-01-01'),

(14, '123 college St', '', 'San Diego', 'CA', '92101', '1234567890', '2020-01-01', '2020-01-01'),

(15, '123 Main St', '', 'San Diego', 'CA', '92101', '1234567890', '2020-01-01', '2020-01-01');

Table

Description automatically generated

**5. Creating Student Payment Details Table**

CREATE TABLE StudentPaymentDetails(

student\_id INTEGER,

account\_holder\_name varchar(30) Not NULL,

payment\_type varchar(10) NOT NULL,

card\_number varchar(30) NOT NULL,

expiry\_date Date Not NULL,

cvv INTEGER NOT NULL,

inserted\_date DATE NOT NULL,

updated\_date DATE NOT NULL,

CONSTRAINT student\_id\_pk2 PRIMARY KEY (student\_id),

CONSTRAINT student\_id\_fk2 FOREIGN KEY (student\_id) REFERENCES Student(student\_id));

**Data entry into student payment details table**

INSERT INTO StudentPaymentDetails VALUES

(1, 'ammie', 'credit', '123457902201', '2020-01-01', '232', '2020-01-01', '2020-01-01'),

(2, 'Joni', 'credit', '9891289182121', '2020-01-01', '632', '2020-01-01', '2020-01-01'),

(3, 'candy', 'credit', '123457902201', '2020-01-01', '221', '2020-01-01', '2020-01-01'),

(4, 'Denny', 'credit', '9891289182121', '2020-01-01', '230', '2020-01-01', '2020-01-01'),

(5, 'Eddy', 'credit', '123457902201', '2020-01-01', '221', '2020-01-01', '2020-01-01'),

(6, 'Fanny', 'credit', '9891289182121', '2020-01-01', '230', '2020-01-01', '2020-01-01'),

(7, 'Gerry', 'credit', '123457902201', '2020-01-01', '221', '2020-01-01', '2020-01-01'),

(8, 'Henny', 'credit', '9891289182121', '2020-01-01', '230', '2020-01-01', '2020-01-01'),

(9, 'terry', 'credit', '123457902201', '2020-01-01', '221', '2020-01-01', '2020-01-01'),

(10, 'matthew', 'credit', '9891289182121', '2020-01-01', '230', '2020-01-01', '2020-01-01'),

(11, 'cindy', 'credit', '123457902201', '2020-01-01', '221', '2020-01-01', '2020-01-01'),

(12, 'nancy', 'credit', '9891289182121', '2020-01-01', '230', '2020-01-01', '2020-01-01'),

(13, 'diana', 'credit', '123457902201', '2020-01-01', '221', '2020-01-01', '2020-01-01'),

(14, 'cari', 'credit', '9891289182121', '2020-01-01', '230', '2020-01-01', '2020-01-01'),

(15, 'james', 'credit', '123457902201', '2020-01-01', '221', '2020-01-01', '2020-01-01');

Table

Description automatically generated

**6. Creating term table**

CREATE TABLE Term(

term\_id INTEGER,

term\_name varchar(30) NOT NULL,

start\_date Date NOT NULL,

end\_date Date NOT NULL,

CONSTRAINT term\_id\_pk PRIMARY KEY (term\_id));

**Data entry into term table**

INSERT INTO Term VALUES

(1, '2018 Fall', '2018-09-01', '2018-12-31'),

(2, '2018 Spring', '2018-01-01', '2018-04-30'),

(3, '2018 Summer', '2018-05-01', '2018-08-31'),

(4, '2019 Fall', '2019-09-01', '2019-12-31'),

(5, '2019 Spring', '2019-01-01', '2019-04-30'),

(6, '2019 Summer', '2019-05-01', '2019-08-31'),

(7, '2020 Fall', '2020-09-01', '2020-12-31'),

(8, '2020 Spring', '2020-01-01', '2020-04-30'),

(9, '2020 Summer', '2020-05-01', '2020-08-31'),

(10, '2021 Fall', '2021-09-01', '2021-12-31'),

(11, '2021 Spring', '2021-01-01', '2021-04-30'),

(12, '2021 Summer', '2021-05-01', '2021-08-31'),

(13, '2022 Fall', '2022-09-01', '2022-12-31'),

(14, '2022 Spring', '2022-01-01', '2022-04-30'),

(15, '2022 Summer', '2022-05-01', '2022-08-31');

Text

Description automatically generated with medium confidence

**7. Creating course table**

CREATE TABLE Course(

course\_id INTEGER,

course\_name varchar(150) NOT NULL,

course\_code varchar(30) NOT NULL,

course\_credit INTEGER NOT NULL,

department\_id INTEGER NOT NULL,

start\_date Date NOT NULL,

end\_date Date NOT NULL,

CONSTRAINT course\_id\_pk PRIMARY KEY (course\_id),

CONSTRAINT department\_id\_fk2 FOREIGN KEY (department\_id) REFERENCES Department(department\_id));

**Data entry into course table**

INSERT INTO Course VALUES

(1, 'Discovering Computer Science', 'CSCE 1010', 4, 1, '2018-09-01', '2018-12-31'),

(2, 'Computer Science I', 'CSCE 1030', 4, 1, '2018-01-01', '2018-04-30'),

(3, 'Computer Science II', 'CSCE 1040', 4, 1, '2018-05-01', '2018-08-31'),

(4, 'Foundations of Data Structures', 'CSCE 1060', 4, 1, '2018-09-01', '2018-12-31'),

(5, 'Applied Artificial Intelligence', 'CSCE 1070', 4, 1, '2018-05-01', '2018-08-31'),

(6, 'Accounting Principles I (Financial Accounting)', 'ACCT 1010', 4, 2, '2022-01-01', '2022-04-30'),

(7, 'Accounting Principles II (Managerial Accounting)', 'ACCT 1020', 4, 2, '2022-05-01', '2022-08-31'),

(8, 'Intermediate Accounting I', 'ACCT 1030', 4, 2, '2022-01-01', '2022-04-30'),

(9, 'Intermediate Accounting II', 'ACCT 1040', 4, 2, '2022-05-01', '2022-08-31'),

(10, 'Financial Statement Analysis', 'ACCT 1050', 4, 2, '2022-01-01', '2022-04-30'),

(11, 'Managerial Accounting', 'ACCT 1060', 4, 2, '2022-05-01', '2022-08-31'),

(12, 'Introduction to Computers in Business', 'BCIS 2610', 4, 3, '2022-01-01', '2022-04-30'),

(13, 'Basic Information Systems', 'BCIS 2620', 4, 3, '2022-05-01', '2022-08-31'),

(14, 'Object-Oriented Programming for Business', 'BCIS 2630', 4, 3, '2022-01-01', '2022-04-30'),

(15, 'Analysis of Business Information Systems', 'BCIS 2640', 4, 3, '2022-05-01', '2022-08-31'),

(16, 'Introduction to Aviation Industry', 'LGAV 3100', 4, 4, '2019-09-01', '2019-12-31');

Table

Description automatically generated with medium confidence

**8. Creating section table**

CREATE TABLE Section(

section\_id INTEGER,

section\_name varchar(30) NOT NULL,

total\_seats INTEGER NOT NULL,

course\_id INTEGER NOT NULL,

term\_id INTEGER NOT NULL,

faculty\_id INTEGER NOT NULL,

CONSTRAINT section\_id\_pk PRIMARY KEY (section\_id),

CONSTRAINT term\_id\_fk1 FOREIGN KEY (term\_id) REFERENCES Term(term\_id),

CONSTRAINT course\_id\_fk1 FOREIGN KEY (course\_id) REFERENCES Course(course\_id),

CONSTRAINT faculty\_id\_fk1 FOREIGN KEY (faculty\_id) REFERENCES Faculty(faculty\_id));

**Data entry into section table**

INSERT INTO Section VALUES

(1, 'Section 1 spring 2022', 30, 1, 1, 1),

(2, 'Section 2 fall 2022', 30, 2, 2, 2),

(3, 'Section 3 summer 2021', 30, 3, 3, 3),

(4, 'Section 1 spring 2021', 30, 4, 4, 4),

(5, 'Section 1 fall 2021', 30, 5, 6, 6),

(6, 'Section 1 summer 2020', 30, 6, 5, 5),

(7, 'Section 1 spring 2020', 30, 7, 7, 7),

(8, 'Section 1 fall 2020', 30, 8, 8, 8),

(9, 'Section 1 summer 2019', 30, 9, 9, 9),

(10, 'Section 1 spring 2019', 30, 10, 10, 10),

(11, 'Section 1 fall 2019', 30, 11, 11, 11),

(12, 'Section 1 summer 2018', 30, 12, 12, 12),

(13, 'Section 1 spring 2018', 30, 13, 13, 13),

(14, 'Section 1 fall 2018', 30, 14, 14, 14),

(15, 'Section 1 summer 2017', 30, 15, 15, 15);

A picture containing text

Description automatically generated

**9. Creating student enrollment table**

CREATE TABLE StudentEnrollment(

enrollment\_id INTEGER,

student\_id INTEGER NOT NULL,

section\_id INTEGER NOT NULL,

enrolled\_date Date NOT NULL,

grade INTEGER,

CONSTRAINT enrollment\_id\_pk PRIMARY KEY (enrollment\_id),

CONSTRAINT student\_id\_fk FOREIGN KEY (student\_id) REFERENCES Student(student\_id),

CONSTRAINT section\_id\_fk FOREIGN KEY (section\_id) REFERENCES Section(section\_id));

**Data entry into student enrollment table**

INSERT INTO StudentEnrollment VALUES

(1, 1, 1, '2020-01-01', 4),

(2, 2, 2, '2020-01-01', 3),

(3, 3, 3, '2020-01-01', 4),

(4, 4, 4, '2020-01-01', 4),

(5, 5, 5, '2020-01-01', 3),

(6, 6, 6, '2020-01-01', 4),

(7, 7, 7, '2020-01-01', 3),

(8, 8, 8, '2020-01-01', 3),

(9, 9, 9, '2020-01-01', 4),

(10, 10, 10, '2020-01-01', 3),

(11, 11, 11, '2020-01-01', 4),

(12, 12, 12, '2020-01-01', 3),

(13, 13, 13, '2020-01-01', 4),

(14, 14, 14, '2020-01-01', 3),

(15, 15, 15, '2020-01-01', 3);

Text

Description automatically generated with medium confidence

**Database in DBMS**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Data Retrieval and Sample Reports**

1. **Display all the course details sorted by their startdate.**

SELECT \* FROM sms.course order by start\_date desc;

**Graphical user interface, table

Description automatically generated with medium confidence**

**2. Display all the student enrollment details with the student first name , enrolled course and the section name.**

select student.first\_name, studentenrollment.enrollment\_id, sms.section.section\_name, sms.course.course\_name

from studentdatabase.studentenrollment join sms.student on sms.student.student\_id = sms.studentenrollment.student\_id

Join sms.section on sms.section.section\_id = sms.studentenrollment.section\_id

Join sms.course on sms.course.course\_id = sms.section.course\_id;

Graphical user interface, application

Description automatically generated

**3. Show all the section details including faculty name and course name sort by course name.**

select course. Course\_name, section.section\_name, faculty.first\_name from section

join faculty on faculty. faculty\_id = section. section\_id

JOIN course on course. course\_id = section.course\_id order by course.course\_name;

Graphical user interface, application

Description automatically generated

**4. Display all the student details including student name, address, payment details, and section details.**

select \* from student join student address on studentaddress.student\_id = student.student\_id join studentpaymentdetails on studentpaymentdetails.student\_id = student.student\_id;

**Graphical user interface, table

Description automatically generated**

**5. Show all courses details along with the department name and term sort by course name**

select course. Course\_name, department. department Name, term.term\_name from course join department on department.department\_id = course.department\_id join section on section.course\_id = course.course\_id join term on term.term\_id = section.term\_id

ORDER by course. course\_name;

Graphical user interface

Description automatically generated