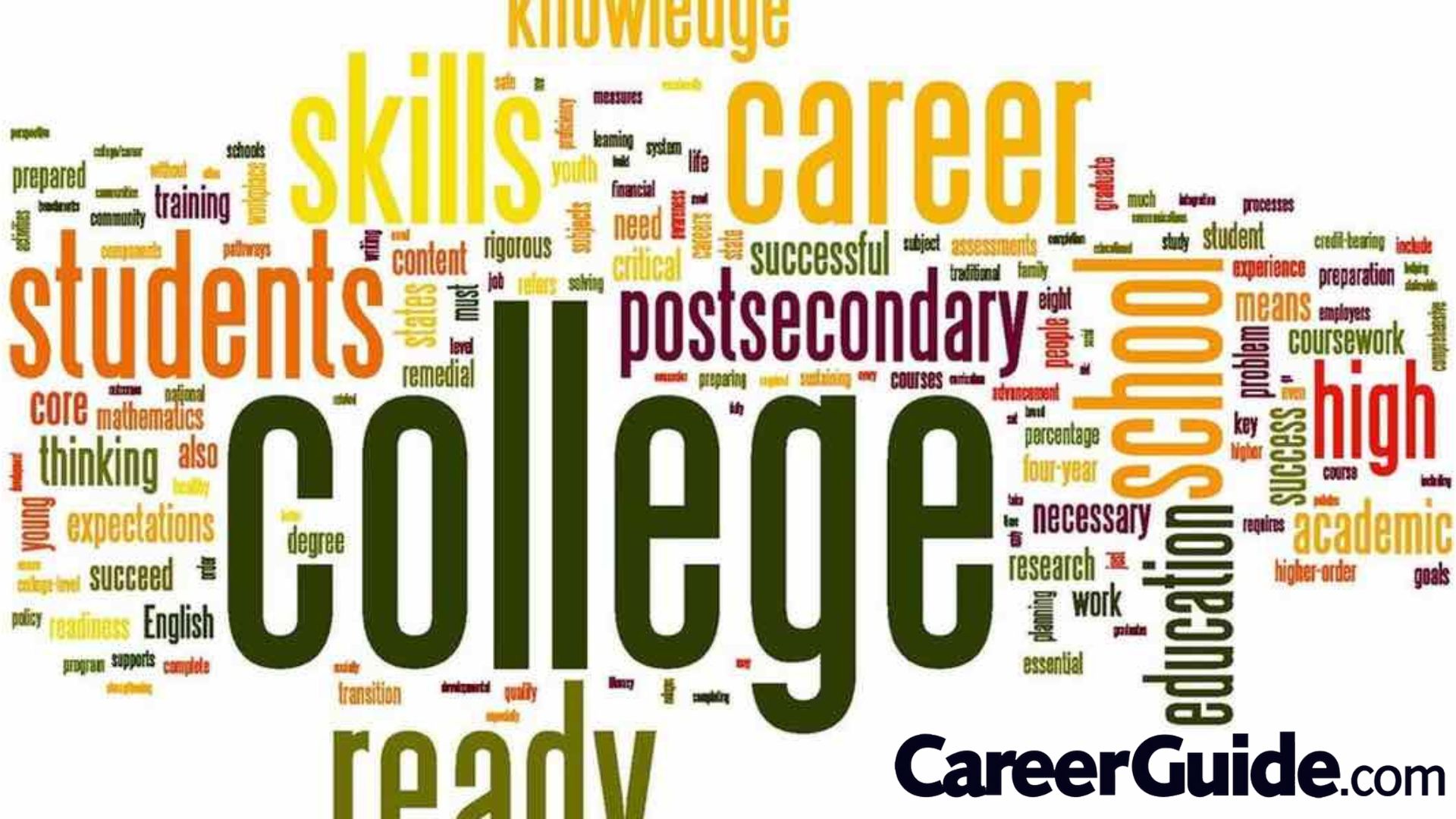
*COLLEGE ENROLLMENT SYSTEM*



SUBMITTED BY :- JASPREET SINGH

TABLE OF CONTENT

INTRODUCTION………………………………………………………………….3

MISSION……………………………………………………………………………..3

OBJECTIVES………………………………………………………………………..4

DATABASE DESIGN……………………………………………………………...5

TABLES……………………………………………………………………………..5

DATA DICTIONARY………………………………………………………….11

ENTITY RELATION DIAGRAM…………………………………………13

DATABASE DEVELOPMENT……………………………………………. 14

INNER JOIN QUERY………………………………………………………15

GROUP BY QUERY………………………………………………………….16

CONCLUSION………………………………………………………………..17

INTRODUCTION

A College Enrollment System is a platform designed to streamline student admissions and registration processes. It enables colleges and universities to efficiently manage large volumes of student data while providing a smooth experience for both administrators and students.

By implementing a College Enrollment System, higher education institutions can enhance student recruitment and retention, optimize resource management, and improve data handling. These systems serve as a centralized platform for managing the entire enrollment process—from initial inquiries to final registration—ensuring a seamless and efficient experience for all stakeholders.

MISSION

The goal of the College Enrollment System is to empower students, institutions, and administrators by providing a seamless, efficient, and user-friendly platform.

OBJECTIVE

Simplify enrolling Procedures: Reduce paperwork and delays by streamlining the admissions, enrolling, and application processes for staff and potential students.

**Improve Accessibility**: By using an inclusive digital platform, make sure that all students, regardless of background, have fair access to chances for higher education.

**Encourage openness**: Clearly communicate and update admissions statuses of students show course availability for admission and institutional requirements in real time.

**Encourage the Making of Decisions**: Provide schools

with data-driven insights to enhance resource providing, optimize admissions procedures.

**Assure Security and Reliability**: Use strong security measures to protect private student information and keep system uptime high to allow for continuous availability.

**DATABSE DESIGN**

TABLES:

1) Create table student

create table students (

student\_id int(10) primary key,

first\_name varchar(50) not null,

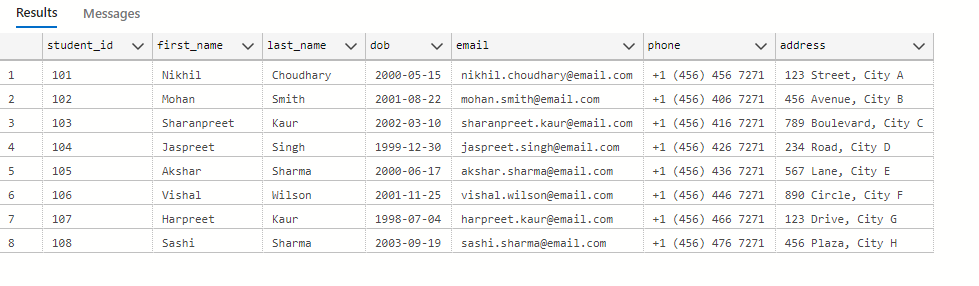
last\_name varchar(50) not null,

dob date not null,

email varchar(120) not null unique,

phone varchar(17) not null,

address varchar(250) not null);



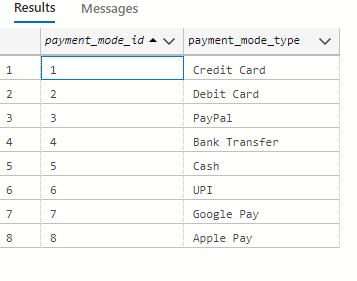
2)Create table payment\_mode

create table payment\_mode (

payment\_mode\_id int(2) primary key auto\_increment,

payment\_mode\_type varchar(30) unique not null

);

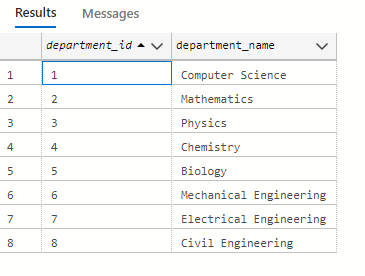


3) Create table department

create table department (

department\_id int(2) primary key auto\_increment,

department\_name varchar(50) not null unique);



4)Create table programs

create table programs (

program\_id int(4) primary key,

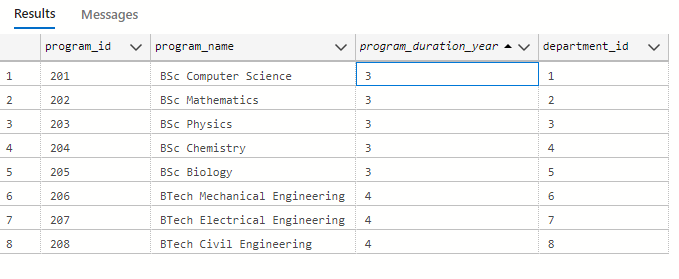
program\_name varchar(150) not null unique,

program\_duration\_year int(1) ,

department\_id int(2) ,

foreign key (department\_id) REFERENCES department(department\_id)

);



5)Create table faculty

create table faculty(

faculty\_id int(10) primary key auto\_increment,

first\_name varchar(50) not null ,

last\_name varchar(50) not null,

email varchar(100) not null unique,

phone varchar(17) not null unique );



6) Create table courses

create table courses (

course\_id int(10) primary key ,

course\_name varchar(150) not null unique,

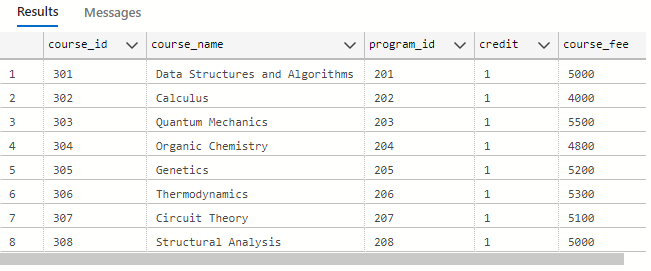
program\_id int(3) ,

credit boolean default(true),

course\_fee int(5) ,

foreign key (program\_id) references programs(program\_id)

);



7) Create table exams

create table exams(

exam\_id int(15) primary key auto\_increment ,

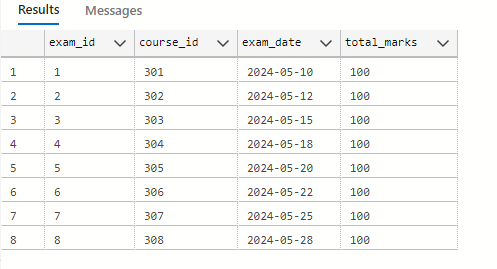
course\_id int(10) not null,

exam\_date date ,

total\_marks int(3),

foreign key (course\_id) references courses (course\_id)

);



8) Create table exam\_result

create table exam\_result(

student\_id int(10) ,

exam\_id int(15),

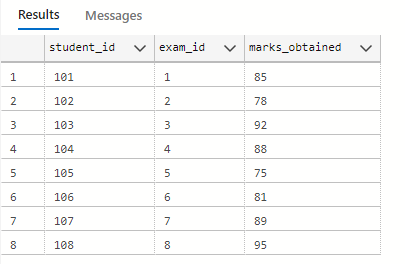
marks\_obtained int(3),

primary key(student\_id , exam\_id), -- composite primary key

foreign key (student\_id) references students(student\_id),

foreign key (exam\_id) references exams(exam\_id)

);



9) Create table enrollments

create table enrollments(

enrollment\_id int(15) primary key auto\_increment,

student\_id int(10) not null,

course\_id int(10) not null,

semester int(2) ,

enrollment\_date date default(CURDATE()),

faculty\_id int(10) ,

payment\_mode\_id int(2),

UNIQUE (student\_id, course\_id),

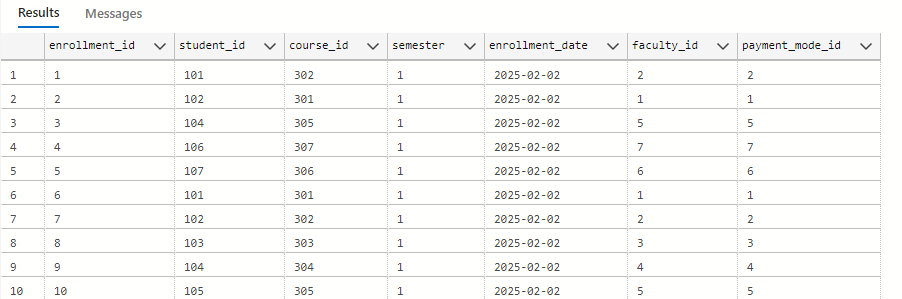
foreign key (student\_id) references students(student\_id),

foreign key (course\_id) references courses (course\_id),

foreign key (faculty\_id) references faculty(faculty\_id),

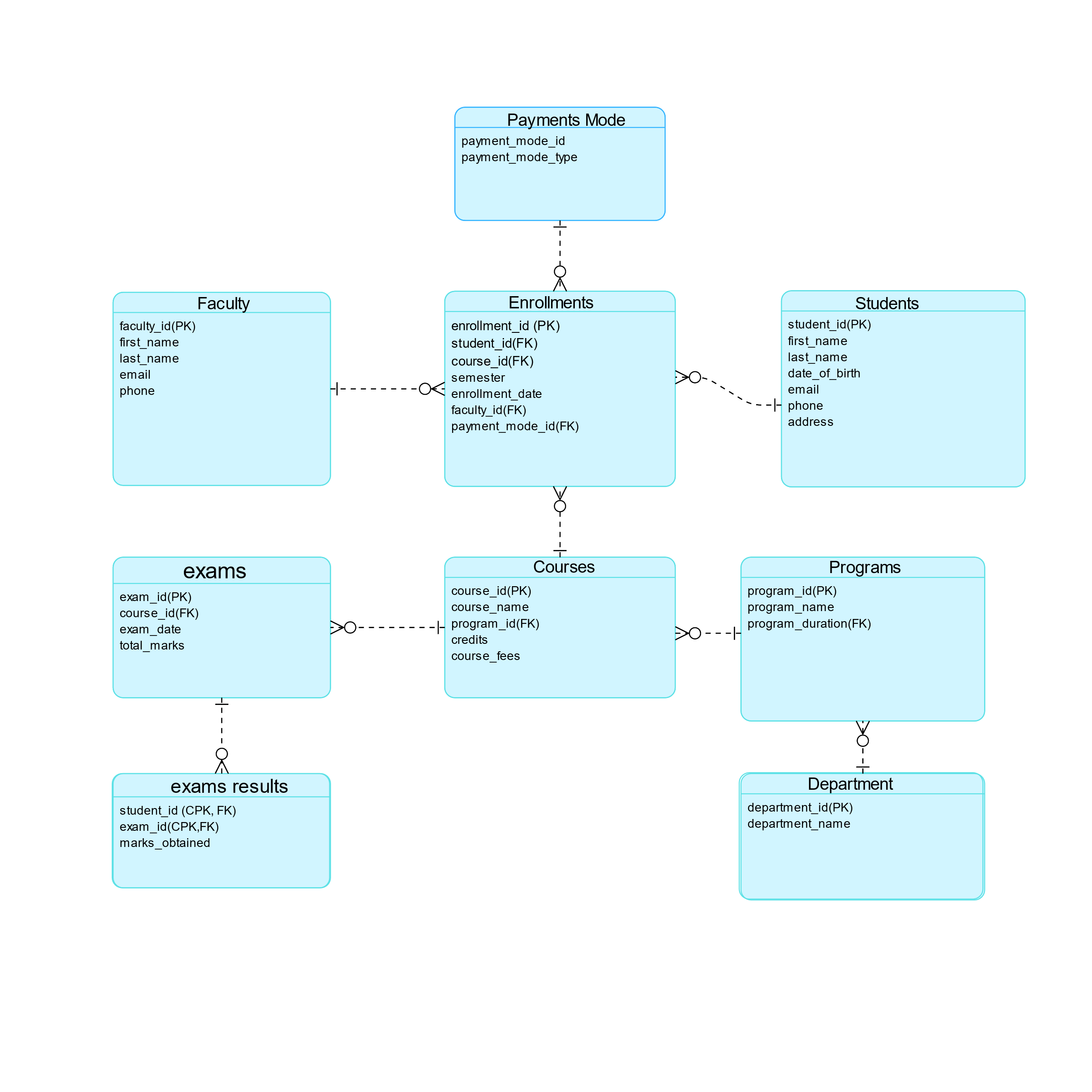
foreign key (payment\_mode\_id) references payment\_mode(payment\_mode\_id)

);



**DATA DICTIONARY:**

|  |  |  |  |
| --- | --- | --- | --- |
| SR NO | FIELD NAME | TYPE | MEANING |
| 1 | Student\_id | Integer | Id number of student |
| 2 | First\_name | Varchar | First name of student |
| 3 | Last\_name | Varchar | Family name of student |
| 4 | Email | Varchar | Email of student |
| 5 | Phone | Varchar | Contact number of student |
| 6 | DateofBirth | Date | Birth date |
| 7 | Address | Varchar | Address of student |
| 8 | Course\_id | Integer | Id number of course |
| 9 | Course\_name | Varchar | Name of course |
| 10 | Credits | Boolean | Number of credits |
| 11 | Program\_name | Varchar | Name of program |
| 12 | Program\_duration\_year | Integer | Duration of program |
| 13 | Department\_id | Varchar | Department name |
| 14 | Program\_id | Integer | Id number of PROGRAM |
| 15 | Payment\_mode\_id | Integer | Id number of the payment |
| 16 | Payment\_mode\_type | Varchar | Mode of payment |
| 17 | Status | Varchar | Status of payment |
| 18 | Faculty\_id | Integer | Id number of faculty |
| 19 | Exam\_date | Date | Date of exam |
| 20 | Exam\_id | Integer | Id number of exam |
| 21 | Total\_marks | Integer | Total Marks of exam |
| 22 | Marks\_obtained | Integer | Total Marks student scored |
| 23 | Enrollment\_id | Integer | Id number of enrolment |
| 24 | Enrollment\_date | Date | Date of enrollment |
| 25 | Semester | Integer | Semester offered |
| 26 | Course\_fee | Integer | Fee of course |

ER DIAGRAM

Students - Enrollments: One-to-Many (A student can enroll in multiple courses).

Courses - Enrollments: One-to-Many (A course can have multiple enrollments).

Courses - Schedules: One-to-One (Each course has a specific schedule).

Faculty - Schedules: One-to-Many (An Faculty can teach multiple courses).

Classrooms -Schedules: One-to-Many (A classroom can host multiple schedules).

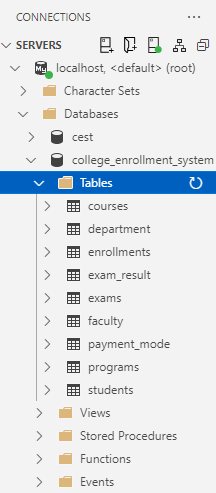
Departments - Courses/Instructors: One-to-Many (A department offers multiple courses and has multiple faculties).

**DATABSE DEVELOPMENT**

1. Query to create database:-

Create database student\_enrollment\_system;

Use student\_enrollment\_system;



INNER JOIN QUERY

1. ) Query to Get Faculty Information for Courses in a Specific Program

SELECT DISTINCT f.first\_name AS faculty\_first\_name, f.last\_name AS

faculty\_last\_name, c.course\_name

FROM faculty f

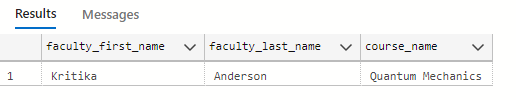
JOIN enrollments en ON f.faculty\_id = en.faculty\_id

JOIN courses c ON en.course\_id = c.course\_id

JOIN programs p ON c.program\_id = p.program\_id

WHERE p.program\_name = 'BSc physics';

OUTPUT:-



GROUPBY QUERY

1. Query to Find the Total Fees Collected for Each Program

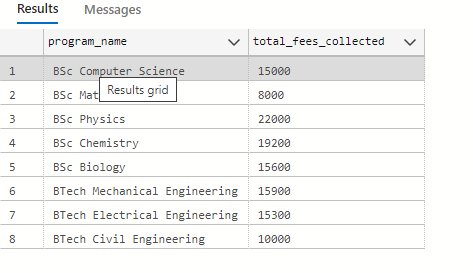
SELECT p.program\_name, SUM(c.course\_fee) AS total\_fees\_collected

FROM enrollments e

JOIN courses c ON e.course\_id = c.course\_id

JOIN programs p ON c.program\_id = p.program\_id

GROUP BY p.program\_name;



CONCLUSION

In conclusion, the College Enrollment System is a vital asset for modern higher education institutions. By streamlining student admissions and registration processes, it enhances efficiency and accuracy in managing large volumes of student data. This system supports the entire enrollment journey, from initial inquiries to final registration, ensuring a seamless and satisfying experience for both administrators and students.