Core Tables:

* Student:
  + student\_id (primary key)
  + name
  + email
  + address
  + phone\_number
  + date\_of\_birth
  + gender
  + program\_id (foreign key to Program table)
  + year\_of\_enrollment
* Faculty:
  + faculty\_id (primary key)
  + name
  + designation
  + email
  + office\_address
  + phone\_number
  + department\_id (foreign key to Department table)
* Course:
  + course\_id (primary key)
  + course\_name
  + credit\_hours
  + department\_id (foreign key to Department table)
  + faculty\_id (foreign key to Faculty table)
* Department:
  + department\_id (primary key)
  + department\_name
  + faculty\_id (foreign key to Faculty table, for department head)
* Registration:
  + registration\_id (primary key)
  + student\_id (foreign key to Student table)
  + course\_id (foreign key to Course table)
  + semester
  + year
* Grades:
  + grade\_id (primary key)
  + registration\_id (foreign key to Registration table)
  + grade

Additional Tables (depending on requirements):

* Program:
  + program\_id (primary key)
  + program\_name
  + department\_id (foreign key to Department table)
* Fees:
  + fee\_id (primary key)
  + fee\_type
  + amount
* Payments:
  + payment\_id (primary key)
  + student\_id (foreign key to Student table)
  + fee\_id (foreign key to Fees table)
  + payment\_date
  + amount\_paid
* Attendance:
  + attendance\_id (primary key)
  + registration\_id (foreign key to Registration table)
  + date
  + attendance\_status
* Exams:
  + exam\_id (primary key)
  + course\_id (foreign key to Course table)
  + exam\_date
  + exam\_type
* User Accounts:
  + user\_id (primary key)
  + username
  + password
  + role (student, faculty, admin)

Relationships:

* One-to-many:
  + Department to Faculty
  + Program to Student
  + Course to Registration
  + Registration to Grades
* Many-to-many (if applicable):
  + Student to Course (through Registration)
  + Faculty to Course

Considerations:

* Data types: Choose appropriate data types for each field (e.g., INT for IDs, VARCHAR for names, DATE for dates).
* Constraints: Enforce data integrity using constraints (e.g., primary keys, foreign keys, unique constraints, not null constraints).
* Normalization: Normalize the schema to reduce redundancy and improve data consistency.
* Security: Implement security measures to protect sensitive data (e.g., user authentication, access controls).