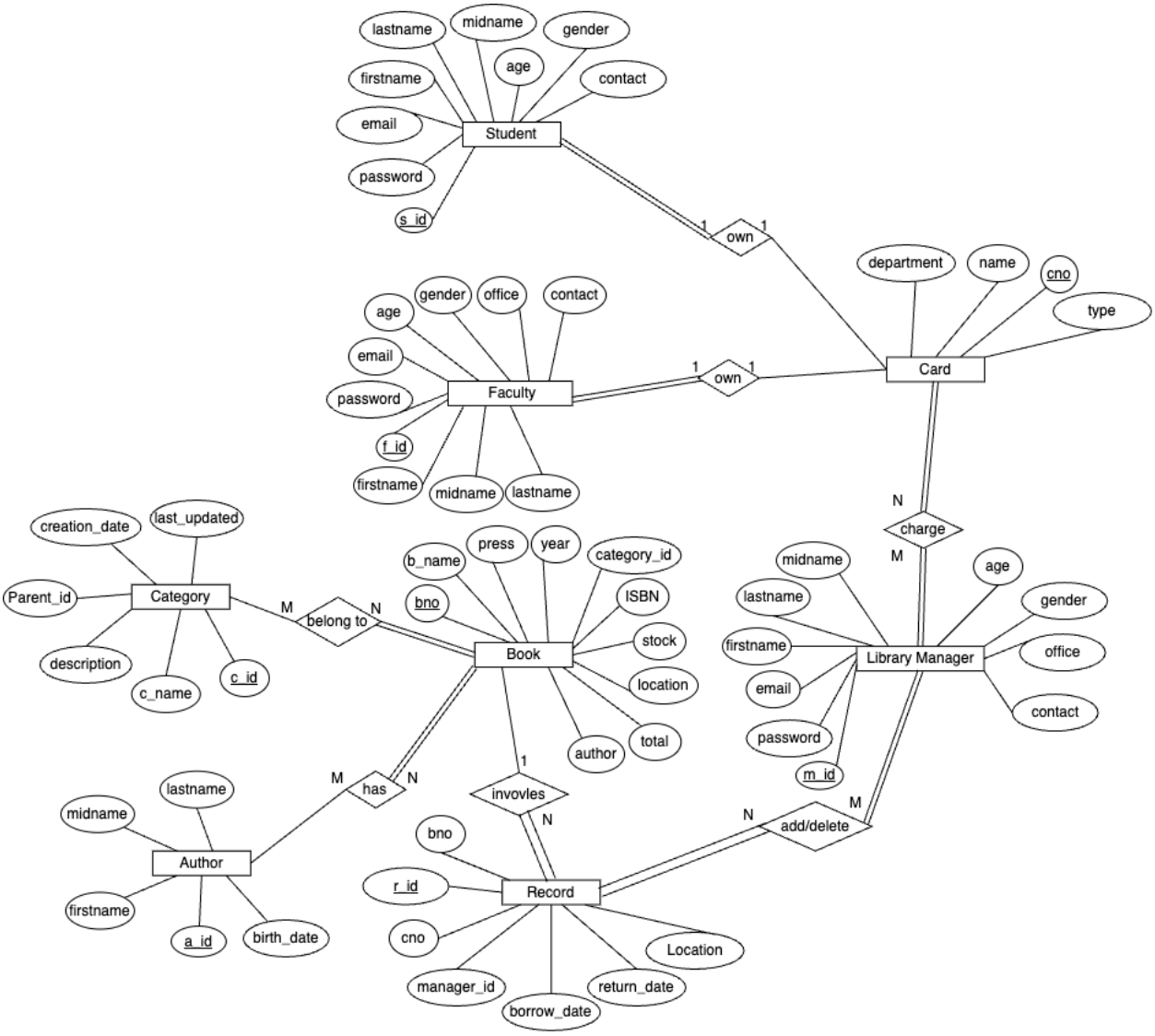
Database Systems - Milestone 2

Group Members & Contribution

Zhenhao Zhang	zzh133@u.rochester.edu	32277234	35%
Mark Xu	kxu28@u.rochester.edu	32025712	35%
Junhao Zhou			30%

Task A

1. Draw an ER diagram



2. Model the entities, attributes, and relationships

1. **Entities**:

   In our ER Diagram, we have eight strong entities. They are as follows

   - Book
   - Card
   - Record
   - Student
   - Faculty
   - Library Manage
   - Author
   - Category

   We do not have any weak entities in our diagram.

   There are no class hierarchies in our ER diagram.

   The cardinality, total participation, and (min, max) constraints have been enumerated in our diagram.

2. **Attributes**:

   Book: bno (Primary Key)/ category_id (Foreign Key from Category)/ ISBN/ Author (Foreign Key from Author)/ bname (Title)/ press (Publisher)/ year (Year of Publication)/ total (Total Copies)/ stock (Available Copies)/ location (Library Location)

   Card: cno (Primary Key)/ name (Cardholder Name)/ department (Department or Affiliation)/ type (Cardholder Type: Student, Faculty, Manager)

   Record: record_id (Primary Key)/ bno (Foreign Key from Book)/ cno (Foreign Key from Card)/ borrow_date (Date when borrowed)/ return_date (Date returned, nullable)/ manager_id (Foreign Key from Library Manager)/ location (Transaction Location)

   Student: s_id (Primary Key)/ s_password (Password)/ s_email (Unique, Email)/ s_firstname (First Name)/ s_midname (Middle Name)/ s_lastname (Last Name)/ age (Age)/ gender (Gender)/ contact (Phone Number)

   Faculty: f_id (Primary Key)/ f_password (Password)/ f_email (Unique, Email)/ f_firstname (First Name)/ f_midname (Middle Name)/ f_lastname (Last Name)/ age (Age)/ gender (Gender)/ office (Faculty's Office)/ contact (Phone Number)

   Library Manager: m_id (Primary Key)/ m_password (Password/ m_email (Unique, Email)/ m_firstname (First Name)/ m_midname (Middle Name)/ m_lastname (Last Name)/ age (Age)/ gender (Gender)/ office (Manager's Office)/ contact (Phone Number)

Author: a_id (Primary Key)/ birthday/ a_firstname/ a_midname/ a_lastname

Category: c_id (Primary Key)/ c_name/ description/ parent_id/ creation_date/ last_updated

3. **Relationships**:

   Own: All the students and faculties own cards to borrow or return books, part of the cards are owned by students, and part are owned by faculties.

   Charge: One library manager can charge many cards, and library managers can share responsibility for a single card.

   Add/delete: One library manager can manage several records, and records can also be managed y several managers.

   Involves: Each record is associated with exactly one book, and the same book can appear in many different records (or no record).

   Has: This means one author can write multiple books, and a book can be co-authored by several authors.

   Belong to: One category can contain many different books, and the same book can be associated with various categories.

3. *Assumptions and Explanations*

Our assumptions are as follows:

1. Each user (Student, Faculty, Library Manager) must have one unique library card (Card) associated with their ID. The Card serves as the identifier for borrowing and returning books.
2. Each Student, Faculty, and Library Manager must have a unique email and ID. The ID acts as a unique identifier for accounts, ensuring no duplicate users.
3. A Library Manager is responsible for managing books, including adding or deleting book records from the system. They can also manage records of borrowed and returned books. In addition, managers control the students' cards (Card) and can activate and deactivate students' cards.
4. Each book (Book) belongs to a category (Category), and each category can be associated with multiple books.
5. A Student or Faculty can borrow multiple books, but a Record must track each borrowing transaction separately, including the borrow_date and return_date. Students can borrow 5 books at the same time, but Faculty don't have that constraint.
6. An Author can write multiple books, and each book is associated with multiple authors.
7. A Record links a specific card number (cno) and a book number (bno) to track book borrowing and returning.
8. Each user (Student) can be either active or inactive (isActive), indicating their status in the library system.

9. A Library Manager is responsible for controlling records, borrowing and returning books, and controlling students' cards.
10. A Book must have stock information (stock, total, and location), and it can be updated by the Library Manager when the stock level changes.
11. Categories are created by library managers. They do not directly associate with books, so some categories might be empty.
12. For attributes like email address or contact that can be treated as multivalued attributes, we only allow users to enter a single value within the domain. In other words, they are all going to be single-valued attributes.

Task B

1. ER to Relational Mapping Algorithm

Step 1: Mapping of Regular Entity Types

In my database, there are a total of 6 strong entities, each uniquely identified by its own primary key or composite primary key. These entities include structured data that is crucial for maintaining referential integrity and ensuring the uniqueness of records within the system.

Record

bno	cno	borrow_data	return_data	manager_id	record_id	Location

Card

cno	name	department	type

Book

bno	category_id	b_name	press	year	price	total	stock	Location

Student

s_id	s_password	s_email	s_firstname	s_midname	s_lastname	age	gender	contact

Faculty

f_id	f_password	f_email	f_firstname	f_midname	f_lastname	Age	gender	office	contact

Library Manager

m_id	m_ password	m_email	m_firstname	m_midname	m_lastname	age	gender	office	contact

Author

a_id	birth_date	a_firstname	a_midname	a_lastname

Category

c_id	c_name	description	parent_id	creation_date	last_updated

Step 2: Mapping of Weak Entity Types

Since we don't have weak entities in our ER diagram, we don't need to perform the mapping for the weak entity types.

Step 3: Mapping of Binary 1:1 Relationship Types

1. Relationship: own (between Card and Student/Faculty)

Each Card is associated with exactly one Student or Faculty, and each Student or Faculty owns exactly one Card. Adding the cno (Card Number, PK of Card) as a foreign key in both the Student and Faculty tables. Since a Card must be owned by a Student or Faculty, we ensure the relationship by adding cno in both tables, creating a direct reference to the Card entity.

Card

cno	name	department	type

Student

s_id	s_password	s_email	s_firstname	s_midname	s_lastname	age	gender	contact

Faculty

f_id	f_password	f_email	f_firstname	f_midname	f_lastname	Age	gender	office	contact

Step 4: Mapping of Binary 1:N Relationship Types.

For this step, we identified the 1 relationship between the following entity relations:

'Book' and 'Record' ⇒ primary key "bno" is transferred to 'Record' as 'bno'.

Record

bno	cno	borrow_data	return_data	manager_id	record_id	Location

Book

bno	category_id	b_name	press	year	price	total	stock	Location

Note:

In the case of the entities Book and Record, our N entity is Record. The primary key from the Book is already present in the Record as a foreign key with the same name.

Step 5: Mapping of Binary M: N Relationship Types.

For this step, we identified N: M relationships as being between the following entity relations. For each pair of entities listed below, we will create a new relation to effectively manage the complex relationships.

1. Book and Author ⇒ create a new "has" relation.
2. Book and Category ⇒ create a new "belong to" relation.
3. Library Manager and Book ⇒ create a new "add/delete" relation.
4. Library Manager and Card ⇒ create a new "charge" relation.

1. Relationship: has (between Author and Book)

For this new relation, we will take the primary keys from both the Author and the Book.

Book

bno	category_id	b_name	press	year	price	total	stock	Location

Author

a_id	birth_date	a_firstname	a_midname	a_lastname

Has

bno	a_id

This table will establish which authors have contributed to which books, allowing multiple authors for a single book and vice versa.

2. Relationship: belong to (between Book and Category)

For this new relation, we will take the primary keys from both Book and Category.

Book

bno	category_id	b_name	press	year	price	total	stock	Location

Category

c_id	c_name	description	parent_id	creation_date	last_updated

Belong_to

bno	c_id

This relation will map books to their respective categories, where each book can belong to multiple categories and each category can include multiple books.

3. Relationship: Add/delete to (between Library manager and Record)

For this new relation, we will take the primary keys from the Library Manager and Record.

Record

bno	cno	borrow_data	return_data	manager_id	r_id	Location

Library Manager

m_id	m_ password	m_email	m_firstna me	m_midna me	m_lastna me	age	gender	office	contact

Add/delete

r_id	m_id

This table allows us to track which library manager added or deleted specific books from the library's collection, handling multiple books per manager and multiple managers handling the same book.

4. Relationship: charge (between Library Manager and Card)

One library manager can charge many cards, and library managers can share responsibility for a single card. This is a M:N relationship, so we are going to add another relation in the schema to represent the cardinality relationship between card and manager, called charge. And we also want to include an attribute in charge relation besides the two primary keys from card and manager, called is_Active, in order to let the manager activate or deactivate the card for students if they have not used their card for a long period of time.

Card

cno	name	department	type

Library Manager

m_id	m_ password	m_email	m_firstna me	m_midna me	m_lastna me	age	gender	office	contact

Charge

cno	m_id	is_active

The mapping of relationship charge establishes the cards that a library manager is responsible for or the library managers that have charged a card.

Step 6: Mapping of Multivalued Attributes

Since we don’t have multivalued attributes in our ER diagram, we don’t need to perform the mapping for the weak entity types.

Step 7: Mapping of N-ary Relationship Types

Card, Book, and Library Manager ⇒ Record

The "Record" table is a central entity that links three other entities via the fields cno, bno, and manager_id. This table tracks borrowing and return dates for books, along with the library manager responsible for the transaction. It effectively establishes a three-way relationship among the entities involved. Here's a more structured representation of the tables and their fields:

Record

bno	cno	borrow_data	return_data	Manager_id	record_id	Location

Library Manager

m_id	m_ password	m_email	m_firstna me	m_midna me	m_lastna me	age	gender	office	contact

Book

bno	category_id	b_name	press	year	price	total	stock	Location

Card

cno	name	department	type

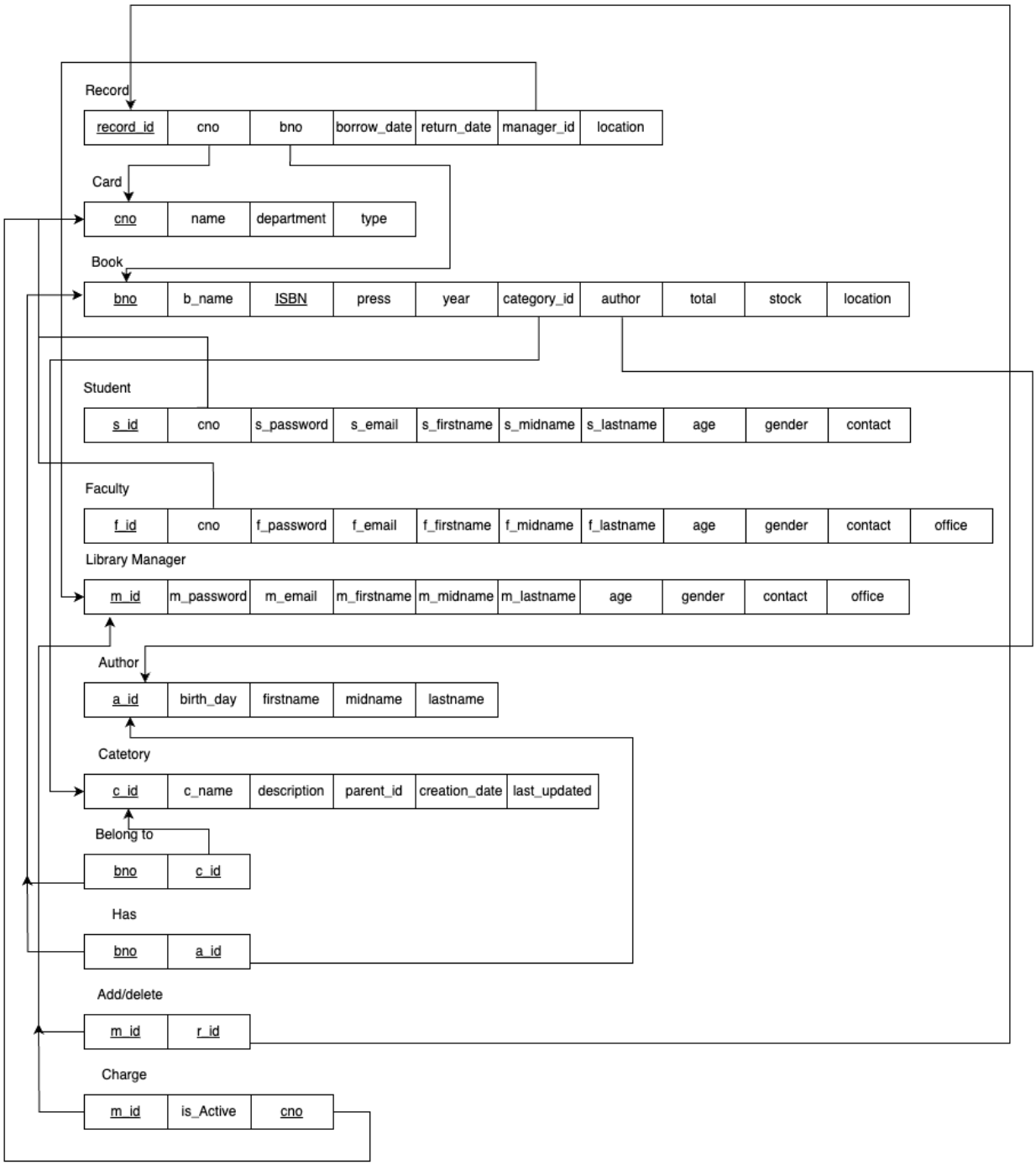
Step 8: Options for Mapping Specialization or Generalization

The ER diagram does not explicitly show any generalization or specialization relationships, so this step is not applicable.

Step 9: Mapping of Union Types (Categories)

There are no union types or categories indicated in the diagram.

Final Relational Schema



2. Database Schema

Book

Attribute name	Type	Key	details	Justification	Description	Action on FK
bno	VARCHAR(20)	PK	NOT NULL, Unique	Uniquely identifies each book for tracking and management.	Unique book number; primary key identifying each book.	Cascade on Delete
bname	VARCHAR(100)		NOT NULL	Essential for identification and cataloging.	Title of the book.	
ISBN	VARCHAR(100)	Unique	NOT NULL	Required for book identification standards.	International Standard Book Number; a unique identifier for books.	
press	VARCHAR(100)		NOT NULL	Publisher information is crucial for bibliographic records.	Publisher of the book.	
Category_ID	INT	FK	NOT NULL	Links to the category table for classification	Category ID representing the book's genre	Cascade on Update
year	YEAR		NOT NULL	Important for bibliographic records and copyright determination.	Year of publication.	
author	VARCHAR(100)		NOT NULL	Authorship is critical for cataloging and legal compliance.	Author(s) of the book.	
total	INTEGER		NOT NULL	Tracks total inventory of titles.	Total number of copies owned by the library.	
stock	INTEGER		NOT NULL	Directly affects borrowing availability.	Number of copies currently available for borrowing.	
Location	VARCHAR(50)		NOT NULL	Ensures the book can be located physically in the library.	Physical location of the book within the library.	

Record

Attribute name	Type	Key	details	Justification	Description	Action on FK
bno	VARCHAR(20)	KF	NOT NULL	Foreign key referencing the Books table to ensure referential integrity.	Represents the book number; foreign key referencing the Books table to ensure referential integrity.	Cascade on Delete
cno	VARCHAR(20)	KF	NOT NULL	Foreign key referencing the Card table to ensure referential integrity.	Represents the students, faculty number; foreign key referencing the card table to ensure referential integrity.	Cascade on Delete
borrow_data	DATE		NOT NULL	Allows tracking of unreturned books.	Date when the book was borrowed; defaults to the current date for new records.	
return_data	DATE		May be NULL	Allows tracking of unreturned books.	Date when the book was returned; can be NULL if the book hasn't been returned yet.	
M_id	VARCHAR(255)	FK	NOT NULL	Foreign key referencing the Library Manager table to record who processed the transaction.	Represents the Manager_ ID; foreign key referencing the Library Manager table to record who processed the transaction.	
record_id	VARCHAR(255)	PK	NOT NULL, Unique	Uniqueness ensures that each transaction is distinct and can be referenced independently.	Unique identifier for each record; primary key with auto-increment to ensure uniqueness.	Cascade on Delete
Location	VARCHAR(20)		NOT NULL	Foreign key (if applicable) or NOT NULL to ensure a location is always recorded.	Represents the location where the transaction took place or the book's location during the transaction.	SET NULL

Card

Attribute name	Type	Key	details	Justification	Description	Action on FK
cno	VARCHAR(20)	PK	NOT NULL, Unique	Primary key to uniquely identify each cardholder.	Unique card number; primary key identifying each cardholder.	CASCADE
name	VARCHAR(50)		NOT NULL	Ensures every card has an associated cardholder name.	Name of the cardholder.	
department	VARCHAR(50)		NOT NULL	Connects card to a department or affiliation.	Department or affiliation of the cardholder.	
type	VARCHAR(20)		NOT NULL	Specifies the type of cardholder to manage permissions.	Type of cardholder (e.g., student, faculty, Library Manager);	

Library Manager

Attribute name	Type	Key	Default	Justification	Description	Action on FK
M_id	INTEGER	PK	NOT NULL, Unique	Primary key to uniquely identify each manager in the database.	Unique identifier for each manager; primary key.	Cascade on Delete
M_password	VARCHAR(255)		NOT NULL	Security requirement to protect manager access.	Password for manager authentication.	
M_Email	VARCHAR(255)	UNIQUE	NOT NULL, Unique	Ensures each manager's email is unique.	Manager's Email	
M_First_name	VARCHAR(50)		NOT NULL	Essential for personal identification and records.	First Name of the manager.	
M_Middle_name	VARCHAR(50)		Nullable	Optional based on cultural naming practices or availability.	Middle Name of the manager.	
M_Last_name	VARCHAR(50)		NOT NULL	Essential for personal identification and records.	Last Name of the manager.	
M_Age	NUMBER		Nullable	Useful for demographic analysis or policy enforcement.	Manager's Age.	
Gender	char(1)		Nullable	Used for demographic analysis and potentially for personalized communication.	Manager's Gender.	
Office	VARCHAR(50)		Nullable	Optional; useful if physical location tracking within premises is needed.	Manager's Office.	
M_contact	VARCHAR(50)		Nullable	Optional; provides a method for direct contact.	Contact information for the manager.	

Faculty

Attribute name	Type	Key	Default	Justification	Description	Action on FK
cno	INTEGER	Unique		Uniquely identifies a condition or context associated with the student.	Faculty Card ID.	Cascade on Delete
F_Id	INTEGER	PK	NOT NULL, Unique	Primary key to uniquely identify each Faculty in the database.	Unique identifier for the Faculty; primary key.	Cascade on Delete
F_password	VARCHAR(255)		NOT NULL	Security requirement to protect Faculty access.	Password for Faculty authentication.	
F_Email	VARCHAR(255)	Unique	NOT NULL, Unique	Each email address is unique to each individual manager.	Faculty's Email	
F_First_name	VARCHAR(50)		NOT NULL	Essential for personal identification and records.	First Name of the Faculty.	
F_Middle_name	VARCHAR(50)		Nullable	Optional based on cultural naming practices.	Middle Name of the Faculty.	
F_Last_name	VARCHAR(50)		NOT NULL	Essential for personal identification and records.	Last Name of the Faculty.	
F_Age	NUMBER		Nullable	Useful for demographic analysis or ensuring age-related policies.	Faculty's Age.	
F_Gender	char(1)		Nullable	Used for demographic analysis and personal identification.	Faculty's Gender.	
F_Office	VARCHAR(50)		Nullable	Optional and can be left blank if not applicable.	Faculty's Office.	
F_contact	VARCHAR(50)		Nullable	Optional for flexibility in providing contact information.	Contact information for the faculty.	

Student

Attribute name	Type	Key details	Default	Justification	Description	Action on FK
cno	INTEGER	Unique		Uniquely identifies a condition or context associated with the student.	Students Card ID.	Cascade on Delete
Student_Id	INTEGER	PK	NOT NULL, Unique	Unique identifier for each student to be able to identify every student in the database.	Unique identifier for each Student; primary key.	Cascade on Delete
S_password	VARCHAR(255)		NOT NULL	Security requirement to protect student access.	Password for Student authentication.	
S_Email	VARCHAR(255)	Unique	NOT NULL, Unique	Every email address is unique to each individual student.	Student's email.	
S_First_name	VARCHAR(50)		NOT NULL	Essential for personal identification and records.	First name of the Student.	
S_Middle_name	VARCHAR(50)		Nullable	Optional based on cultural naming practices.	Middle name of the Student.	
S_Last_name	VARCHAR(50)		NOT NULL	Essential for personal identification and records.	Last name of the Student	
S_Age	NUMBER		Nullable	Age can be useful for demographic analysis and service eligibility.	Student's age	
S_Gender	char(1)		Nullable	Can be used for demographic analysis and personal identification.	Char to determine if Student is male or female.	
S_contact	VARCHAR(50)		Nullable	Contact information is crucial for communication but may not always be required.	Contact information (Phone number).	

Category

Category						
Attribute name	Type	Key	details	Justification	Description	Action on FK
C_id	INT	Primary	Not Null, Unique	Unique identifier for each entry	Customer ID	Cascade on Delete
C_name	VARCHAR(255)		Not Null	Used for identifying customer by name	Customer Name	
description	TEXT		Nullable	Additional information about the customer	Description of the customer	
Parent_ID	INT	FK	Nullable	Links to the parent entity if applicable	ID of the parent customer	Set Null on Delete
Creation_data	DATE		Not Null	To record when the entry was created	Date when the customer was registered	
Last_updated	TIMESTAMP		Not Null	To know when the entry was last updated	Timestamp of the last update	

Author

Attribute name	Type	Key	details	Justification	Description	Action on FK
A_First_name	VARCHAR(255)		NOT NULL	Essential for personal identification	Author's first name	
A_Middle_name	VARCHAR(255)		NOT NULL	Not all authors have a middle name	Author's middle name	
A_Last_name	VARCHAR(255)		NOT NULL	Essential for personal identification	Author's last name	
A_ID	INT	PK	NOT NULL, Unique	Unique identifier for each author	Author ID	Cascade on Delete
birth_date	DATA		NOT NULL	Necessary to provide age-related information	Author's date of birth	

Belong_to

Belong to						
Attribute name	Type	Key	details	Justification	Description	Action on FK
bno	VARCHAR(20)	PK	NOT NULL, Unique	Uniquely identifies each book for tracking and management.	Unique book number; primary key identifying each book.	Cascade on Delete
C_id	INT		Not Null, Unique	Unique identifier for each entry	Customer ID	Cascade on Delete

Has

Has						
Attribute name	Type	Key	details	Justification	Description	Action on FK
bno	VARCHAR(20)	PK	NOT NULL, Unique	Uniquely identifies each book for tracking and management.	Unique book number; primary key identifying each book.	Cascade on Delete
A_ID	INT		NOT NULL, Unique	Unique identifier for each author	Author ID	Cascade on Delete

Add/delete

Add/ Delete						
Attribute name	Type	Key	details	Justification	Description	Action on FK
M_id	INTEGER	PK	NOT NULL, Unique	Primary key to uniquely identify each manager in the database.	Unique identifier for each manager; primary key.	Cascade on Delete
record_id	VARCHAR(255)		NOT NULL, Unique	Uniqueness ensures that each transaction is distinct and can be referenced independently.	Unique identifier for each record; primary key with auto-increment to ensure uniqueness.	Cascade on Delete

Charge

Charge						
Attribute name	Type	Key	details	Justification	Description	Action on FK
M_id	INTEGER	PK	NOT NULL, Unique	Primary key to uniquely identify each manager in the database.	Unique identifier for each manager; primary key.	Cascade on Delete
cno	INTEGER			Uniquely identifies a condition or context associated with the student.	Faculty Card ID.	Cascade on Delete
is_Active	Boolean		NOT NULL	Status flag (active/inactive).	Helps in identifying whether a charge is currently applicable or archived.	Represents whether the charge is currently active in the system.