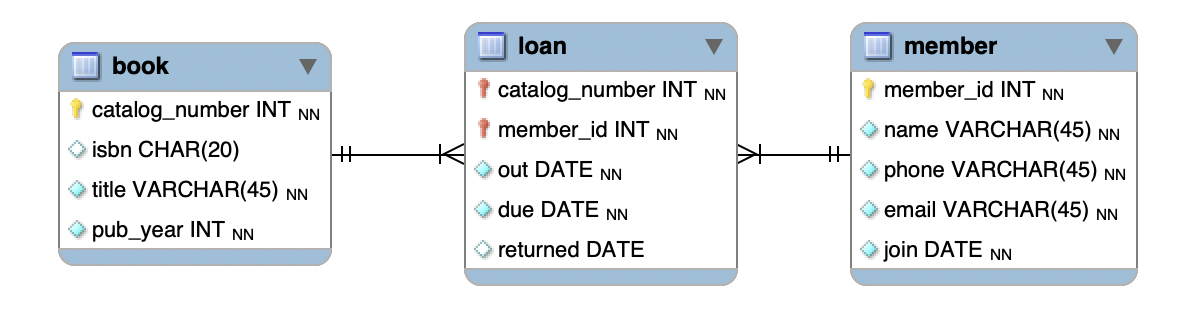
Exercise 5 – Normalization

**Relational Schema in Text**

Sometimes we need to “design” a database without a diagraming tool. While this is certainly not as enjoyable, on some tests, you will not have access to such tools. Consider the library system from Lab 4. Suppose that we came up with the following ERD:



This could also be written in plain-text format as this:

book

\* catalog\_number: integer, primary key

\* isbn; string, unique

\* title; string, not null

\* pub\_year: int, not null

member

\* member\_id; integer, primary key

\* name: string, not null

\* etc

loan

\* catalog\_number: int, primary key, foreign key to book table

\* member\_id: int, primary key, foreign key to member table

\* out: date, not null

\* due: date, not null

\* return: date

# Part A – With Instructor

## Question 1

The following dependency diagram indicates that an airline customer can have many flights for one or more destinations over time. Based on this diagram, **draw an ERD in MySQL Workbench whose tables are in at least second normal form (2NF)**.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| FlightNum | CustID | Date | LoungeAccess | CustName | DestCity | LuggNumAllowed |

## Question 2

Suppose that we have a movie ticket system with the following columns:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Key Information** |
| TicketID | int | Primary Key |
| CustomerID | int | Primary Key |
| MovieID | int |  |
| TheaterID | int |  |
| ShowTime | datetime |  |
| CustomerName | varchar | Dependency on CustomerID |
| CustomerEmail | varchar | Dependency on CustomerID |
| MovieTitle | varchar | Dependency on MovieID |
| MovieGenre | varchar | Dependency on MovieID |
| TheaterName | varchar | Dependency on TheaterID |
| TheaterLocation | varchar | Dependency on TheaterID |

1. Identify the partial dependencies on this table.

CustomerName, CustomerEmail

1. Rewrite the table in second normal form with the partial dependencies removed.
2. Identify the transitive dependencies on the table in 2NF.

MovieTitle, MovieGenre, TheaterName, TheaterLocation

1. Rewrite the table in third normal form.

## Question 3

To keep track of equipment at a certain university, the following table of information is kept:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item ID | Item Name | Room | Building Name | Bldg  Code | Manager |
| 20245 | Xerox Printer | 10255 | College of Computing & IT | CCIT | Will Ali |
| 22245 | HP Monitor | 20229 | College of Health Sciences | CHS | Ashraf Tany |
| 20123 | HP Monitor | 10229 | College of Computing & IT | CHS | Will Ali |
| 20124 | Xerox Scanner | 20112 | College of Health Sciences | CHS | Ashraf Hany |
| 20245 | Xerox Keyboard | 20112 | College of Health Sciences | CCIT | Will Ali |
| 21674 | Xerox Keyboard | 20334 | College of Computing & IT | CCIT | Wael Ali |
| 28539 | HP Laptop | 10256 | College of Computing & IT | CCIT | Will Ali |

1. Which columns have data anomalies in them?

Bldg Code, Manager(?), ItemID

1. Draw a dependency diagram of the information found in the table. You could use draw.io or do this by hand.
2. Normalize the table into 2NF and write the relational schema in text format using Word or Notepad.
3. Normalize the table(s) into 3NF and write the relational schema in text format using Word or Notepad.
4. Draw the ERD of the 3NF using MySQL Workbench, take a screenshot and insert into your document.

Have done the above in class.

## Question 4

Consider the following table definition:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Key Information** |
| RentalID | int | Primary Key |
| BoatID | date | Primary Key |
| CustomerID | int | Primary Key |
| RentalDate | varchar |  |
| BoatName | int |  |
| BoatType | varchar |  |
| CustomerName | varchar |  |
| CustomerEmail | varchar |  |
| DockNumber | int |  |
| RentalRate | date |  |

1. Identify the anomilies in the previous table

BoatID type is date; RentalDate is varchar; BoatName is int; RentalRate is date.

1. Draw the table in 2NF.
2. Draw the table in 3NF.

Since the 2NF version already satisfies the conditions for 3NF (no transitive relations), no changes are needed.

A screen shot of a computer screen

Description automatically generated

# Part B – Additional Exercise

Consider the table of information about patients at a certain medical clinic. Design a relational model that is in 3NF that could store the information shown in this table. Please note that additional attributes may be required.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **pat\_id** | 2015 | 2457 | 2875 | 3984 |
| **pat\_name** | Amjad | Linda | Mina | Jackie |
| **pat\_address** | 24 Al-Sadd Rd | 8th Street, number 12 | 12 Lusail Blvd | 123 Ibn Sina Ave. |
| **doc\_id** | 105 | 214 | 166 | 112 |
| **doc\_spec** | Endocrinology | Orthopedic | Neurology | Ophthalmology |
| **doc\_name** | Emanuel | Ali | Arun | Lakshmi |
| **doc\_loc** | Al Sadd | Lusail | Lusail | Al Khor |
| **pat\_diag** | Diabetes | Herniated Disc | Parkinson’s | Cataract |
| **pat\_proc** | Insulin therapy | Disc Surgery |  | Lens Exchange |

Draw the diagram in MySQL Workbench and write the relational schema in plain text using notepad.

