Final Project: Phase 3 & 4

PROG2111

Group Member(s): Josiah Williams and Jobair Ahmed Jisan

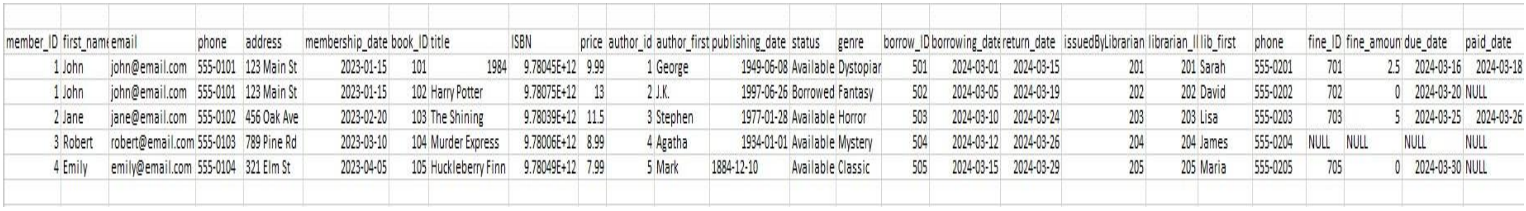
Student Numer(s): 8706882 and 8944480

Submission Date: 2025-12-03

Table of Contents

1. Phase 3: Normalization (1NF-3NF)

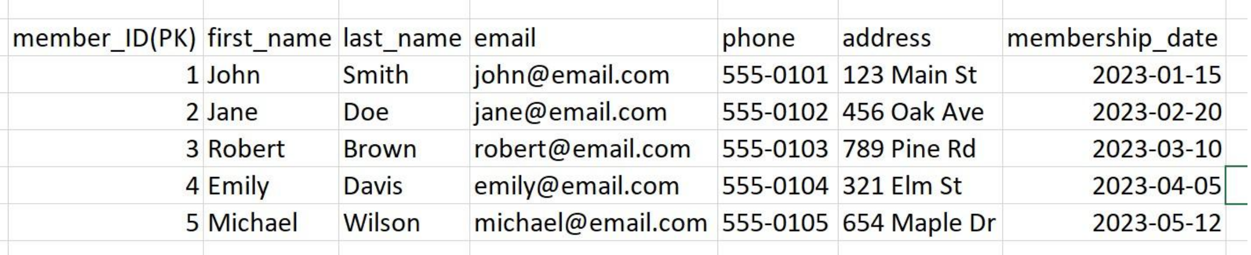
Part 1: Database Normalization 1NF



member_ID	first_nam	email	phone	address	membership_date	book_ID	title	ISBN	price	author_id	author_first	publishing_date	status	genre	borrow_ID	borrowing_dat	return_date	issuedByLibrarian	librarian_I	lib_first	phone	fine_ID	fine_amoun	due_date	paid_date
1	John	john@email.com	555-0101	123 Main St	2023-01-15	101	1984	9.78045E+12	9.99	1	George	1949-06-08	Available	Dystopiar	501	2024-03-01	2024-03-15	201	201	Sarah	555-0201	701	2.5	2024-03-16	2024-03-18
1	John	john@email.com	555-0101	123 Main St	2023-01-15	102	Harry Potter	9.78075E+12	13	2	J.K.	1997-06-26	Borrowed	Fantasy	502	2024-03-05	2024-03-19	202	202	David	555-0202	702	0	2024-03-20	NULL
2	Jane	jane@email.com	555-0102	456 Oak Ave	2023-02-20	103	The Shining	9.78039E+12	11.5	3	Stephen	1977-01-28	Available	Horror	503	2024-03-10	2024-03-24	203	203	Lisa	555-0203	703	5	2024-03-25	2024-03-26
3	Robert	robert@email.com	555-0103	789 Pine Rd	2023-03-10	104	Murder Express	9.78006E+12	8.99	4	Agatha	1934-01-01	Available	Mystery	504	2024-03-12	2024-03-26	204	204	James	555-0204	NULL	NULL	NULL	NULL
4	Emily	emily@email.com	555-0104	321 Elm St	2023-04-05	105	Huckleberry Finn	9.78049E+12	7.99	5	Mark	1884-12-10	Available	Classic	505	2024-03-15	2024-03-29	205	205	Maria	555-0205	705	0	2024-03-30	NULL

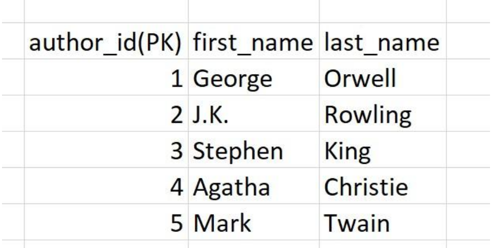
2NF

Member



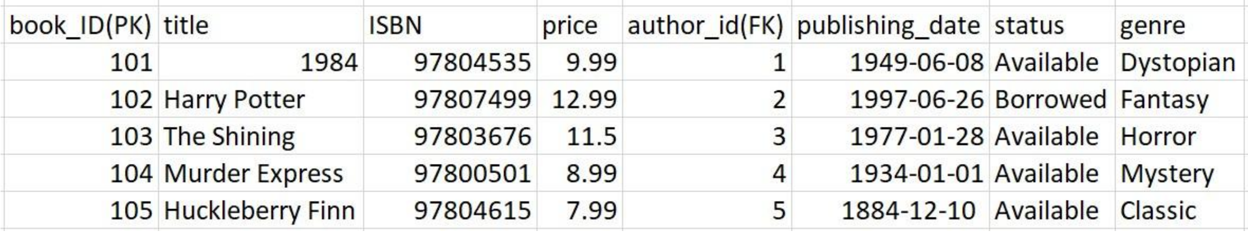
member_ID(PK)	first_name	last_name	email	phone	address	membership_date
1	John	Smith	john@email.com	555-0101	123 Main St	2023-01-15
2	Jane	Doe	jane@email.com	555-0102	456 Oak Ave	2023-02-20
3	Robert	Brown	robert@email.com	555-0103	789 Pine Rd	2023-03-10
4	Emily	Davis	emily@email.com	555-0104	321 Elm St	2023-04-05
5	Michael	Wilson	michael@email.com	555-0105	654 Maple Dr	2023-05-12

Author



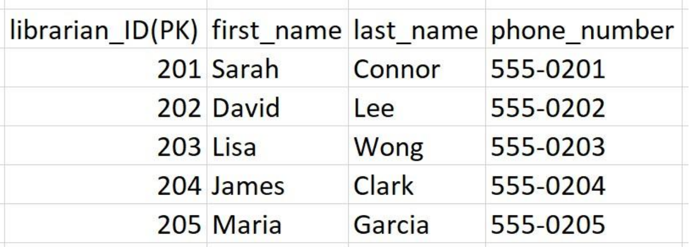
author_id(PK)	first_name	last_name
1	George	Orwell
2	J.K.	Rowling
3	Stephen	King
4	Agatha	Christie
5	Mark	Twain

Book



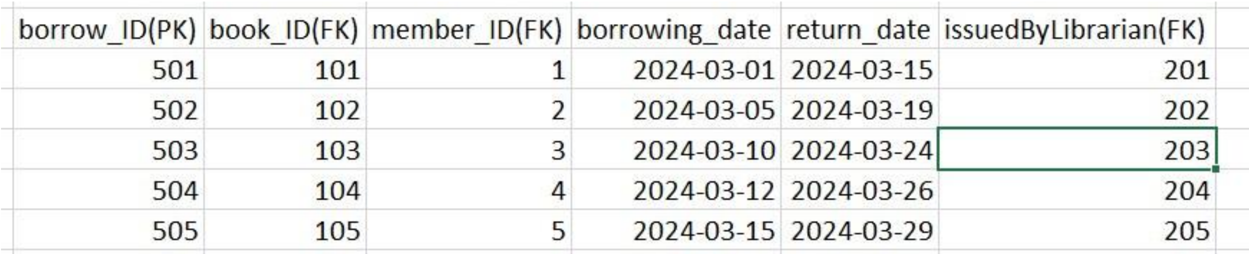
book_ID(PK)	title	ISBN	price	author_id(FK)	publishing_date	status	genre
101	1984	97804535	9.99	1	1949-06-08	Available	Dystopian
102	Harry Potter	97807499	12.99	2	1997-06-26	Borrowed	Fantasy
103	The Shining	97803676	11.5	3	1977-01-28	Available	Horror
104	Murder Express	97800501	8.99	4	1934-01-01	Available	Mystery
105	Huckleberry Finn	97804615	7.99	5	1884-12-10	Available	Classic

Librarian



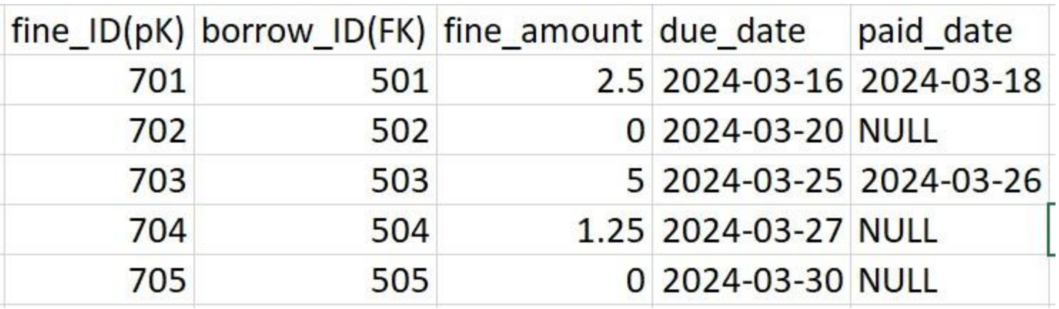
librarian_ID(PK)	first_name	last_name	phone_number
201	Sarah	Connor	555-0201
202	David	Lee	555-0202
203	Lisa	Wong	555-0203
204	James	Clark	555-0204
205	Maria	Garcia	555-0205

Borrow



borrow_ID(PK)	book_ID(FK)	member_ID(FK)	borrowing_date	return_date	issuedByLibrarian(FK)
501	101	1	2024-03-01	2024-03-15	201
502	102	2	2024-03-05	2024-03-19	202
503	103	3	2024-03-10	2024-03-24	203
504	104	4	2024-03-12	2024-03-26	204
505	105	5	2024-03-15	2024-03-29	205

Fine



fine_ID(pK)	borrow_ID(FK)	fine_amount	due_date	paid_date
701	501	2.5	2024-03-16	2024-03-18
702	502	0	2024-03-20	NULL
703	503	5	2024-03-25	2024-03-26
704	504	1.25	2024-03-27	NULL
705	505	0	2024-03-30	NULL

3NF

After converting the tables to 2NF, we checked for transitive dependencies.

A transitive dependency occurs when a non-key attribute depends on another non-key attribute.
In our design, all attributes depend directly on their table’s primary key and none depend on other non-key attributes. Therefore, there are no transitive dependencies, and the tables are already in Third Normal Form (3NF).

Part 2: Database Normalization Report

Normalization Process Summary Step
1: 1NF

- All library data was in one table, maintained atomicity
- Member info repeated for each book borrowed
- Author details copied for every book
- Hard to update or maintain

Step 2: 2NF
What we did:

- Split data into 6 separate tables
- Gave each table a unique ID (primary key)
- Made sure each column had single values only

Tables Created:

- Member - Library users
- Author - Book writers
- Book - Library inventory
- Librarian - Library staff
- Borrow - Book lending records
- Fine - Late fees

Step 3: 3NF What
we checked:

- Made sure everything in each table relates directly to that table's main ID
- Since all tables use single ID columns, they were already in 2NF
- Checked that no data depends on other non-key data
- Added connections between tables (foreign keys)
- Example: Book table now points to Author table instead of storing author names

Part 3: How We Fixed Redundancy and Organized Data

1. Redundancy Problems We Solved:

Before (Messy):

- John Smith's phone number repeated 10+ times
- Stephen King's name copied for every horror book
- Same librarian info on every borrowing slip

After (Clean):

- John's details stored ONCE in Member table

- Stephen King stored ONCE in Author table
- Books just reference author by ID number
- Borrowings just reference member by ID

2. Data Organization Clear Relationships:

- Authors → write → Books
- Members → borrow → Books (through Borrow table)
- Librarians → issue → Borrowings
- Borrowings → may have → Fines

Benefits:

- Save Space - No repeating information
- Easy Updates - Change phone number in one place
- No Mistakes - Can't have Stephen King with two different spellings
- Fast Searches - Find all books by an author quickly

The database is now efficient, organized, and ready for the library to use!

2. Phase 4: DDL Statements

```
#Create Database
CREATE DATABASE Library_Management_System;
USE Library_Management_System;

#Tables
CREATE TABLE Member (
   member_ID INT PRIMARY KEY
AUTO_INCREMENT, first_name VARCHAR(50) NOT
NULL, last_name VARCHAR(50) NOT NULL, email
VARCHAR(50) UNIQUE NOT NULL, phone
VARCHAR(50), membership_date DATE
);
CREATE TABLE Librarian (
   librarian_ID INT PRIMARY KEY
AUTO_INCREMENT, first_name VARCHAR(50) NOT
NULL, last_name VARCHAR(50) NOT NULL,
phone_number VARCHAR(50)
);
CREATE TABLE Author( author_id INT PRIMARY
KEY AUTO_INCREMENT, first_name
VARCHAR(50) NOT NULL, last_name
VARCHAR(50) NOT NULL
);
CREATE TABLE BOOK ( book_ID INT PRIMARY
KEY AUTO_INCREMENT,
title VARCHAR(50) NOT NULL,
ISBN INT UNIQUE, price
DECIMAL(5,2), author_ID INT
NOT NULL, publishing_date
DATE, book_status BOOL,
genre VARCHAR(50),
FOREIGN KEY (author_ID) REFERENCES Author(author_ID)
);
CREATE TABLE Borrow ( borrow_ID INT
PRIMARY KEY AUTO_INCREMENT, book_ID INT
NOT NULL, member_ID INT NOT NULL,
issuedByLibrarian INT NOT NULL, borrowing_date
DATE, return_date DATE,
FOREIGN KEY (book_ID) REFERENCES Book(book_ID),
FOREIGN KEY (member_ID) REFERENCES Member(member_ID),
FOREIGN KEY (issuedByLibrarian) REFERENCES Librarian(librarian_ID)
);
CREATE TABLE Fines(
fine_ID INT PRIMARY KEY AUTO_INCREMENT,
borrow_ID INT NOT NULL,
fine_amount
DECIMAL(5,2),     due_date
DATE, paid_date DATE,
FOREIGN KEY (borrow_ID) REFERENCES Borrow(borrow_ID)
);

#Add data to tables from normilazad data
INSERT INTO Member (first_name, last_name, email, phone, membership_date) VALUES
('John', 'Smith', 'john@email.com', '555-0101', '2023-01-15'),
('Jane', 'Doe', 'jane@email.com', '555-0102', '2023-02-20'),
('Robert', 'Brown', 'robert@email.com', '555-0103', '2023-03-10'),
('Emily', 'Davis', 'emily@email.com', '555-0104', '2023-04-05'), ('Michael',
'Wilson', 'michael@email.com', '555-0105', '2023-05-12');

INSERT INTO Author (author_ID, first_name, last_name) VALUES
(1, 'George', 'Orwell'),
(2, 'J.K.', 'Rowling'),
(3, 'Stephen', 'King'),
(4, 'Agatha', 'Christie'),
(5, 'Mark', 'Twain');

INSERT INTO Book (book_ID, title, ISBN, price, author_ID, publishing_date, book_status, genre) VALUES
(101, '1984', '97804535', 9.99, 1, '1949-06-08', TRUE, 'Dystopian'),
(102, 'Harry Potter', '97807499', 12.99, 2, '1997-06-26', FALSE, 'Fantasy'),
(103, 'The Shining', '97803676', 11.50, 3, '1977-01-28', TRUE, 'Horror'),
(104, 'Murder Express', '97800501', 8.99, 4, '1934-01-01', TRUE, 'Mystery'), (105,
'Huckleberry Finn', '97804615', 7.99, 5, '1884-12-10', TRUE, 'Classic');

INSERT INTO Librarian (librarian_ID, first_name, last_name, phone_number) VALUES
(201, 'Sarah', 'Connor', '555-0201'),
(202, 'David', 'Lee', '555-0202'),
(203, 'Lisa', 'Wong', '555-0203'),
(204, 'James', 'Clark', '555-0204'),
(205, 'Maria', 'Garcia', '555-0205');

INSERT INTO Borrow (borrow_ID, book_ID, member_ID, issuedByLibrarian, borrowing_date, return_date)
VALUES
(501, 101, 1, 201, '2024-03-01', '2024-03-15'),
(502, 102, 2, 202, '2024-03-05', '2024-03-19'),
(503, 103, 3, 203, '2024-03-10', '2024-03-24'),
(504, 104, 4, 204, '2024-03-12', '2024-03-26'),
(505, 105, 5, 205, '2024-03-15', '2024-03-29');

INSERT INTO Fines (fine_ID, borrow_ID, fine_amount, due_date, paid_date) VALUES
(701, 501, 2.50, '2024-03-16', '2024-03-18'),
(702, 502, 0.00,  '2024-03-20', NULL),
(703, 503, 5.00,  '2024-03-25', '2024-03-26'),
(704, 504, 1.25,  '2024-03-27', NULL),
(705, 505, 0.00,  '2024-03-30', NULL);

# Check All data was added
SELECT * FROM author;
SELECT * FROM book;
SELECT * FROM borrow;
SELECT * FROM fines;
SELECT * FROM librarian;
SELECT * FROM member;

#Create
INSERT INTO Member (first_name, last_name, email, phone, membership_date) VALUES ('Will', 'Dillon',
'will@email.com', '555-235', '2022-01-15');

#Read
SELECT member_ID, first_name, last_name, email FROM Member WHERE last_name = 'Dillon';

#Update
UPDATE Member
SET first_name = 'Bob'
WHERE member_ID = 6;

#Delete
```

```
DELETE FROM  Member Where member_ID = 6;
```