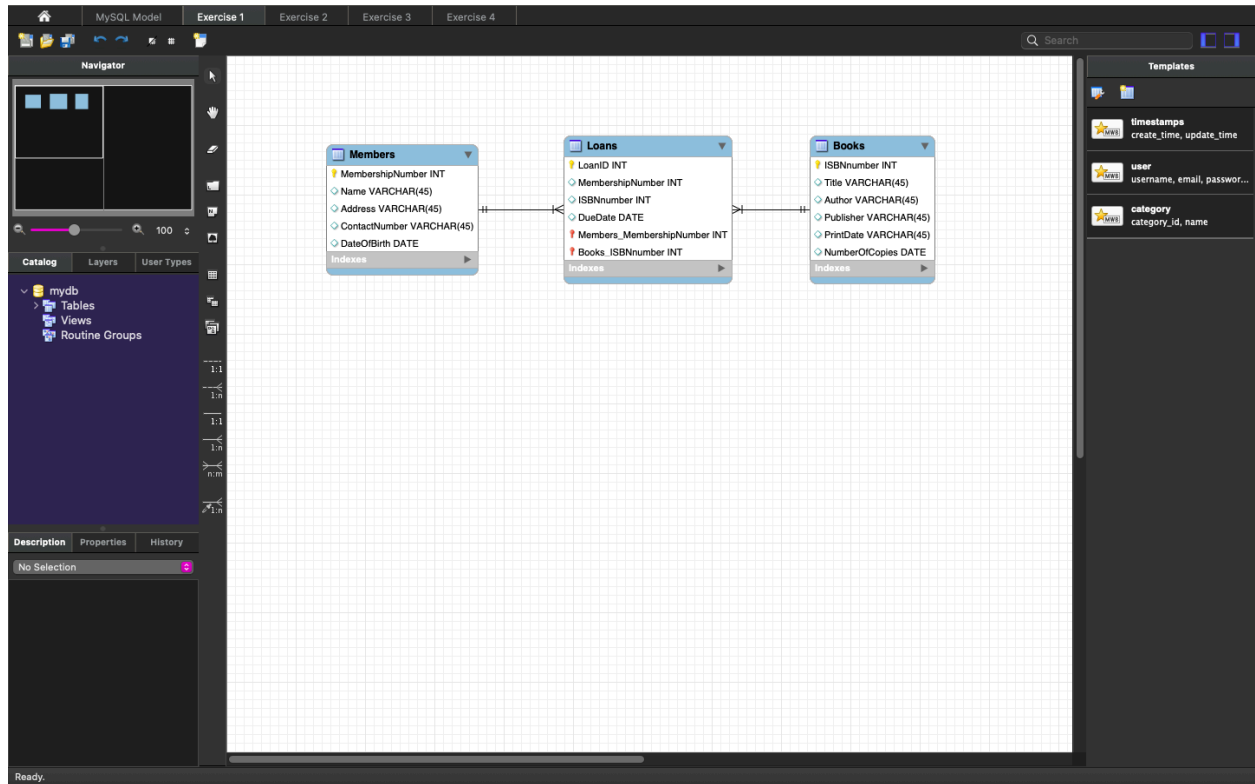


Database Lab 5

Exercise 1



Tables

Members Table

Column Name	Data Type	Constraints
membership_number	INT	PRIMARY KEY
name	VARCHAR(255)	NOT NULL
address	VARCHAR(255)	NOT NULL
contact_number	VARCHAR(20)	NOT NULL
date_of_birth	DATE	NOT NULL

Books Table

Column Name	Data Type	Constraints
ISBN	VARCHAR(13)	PRIMARY KEY
title	VARCHAR(255)	NOT NULL
author	VARCHAR(255)	NOT NULL
publisher	VARCHAR(255)	NOT NULL
print_date	DATE	NOT NULL
number_of_copies	INT	NOT NULL

Loans Table

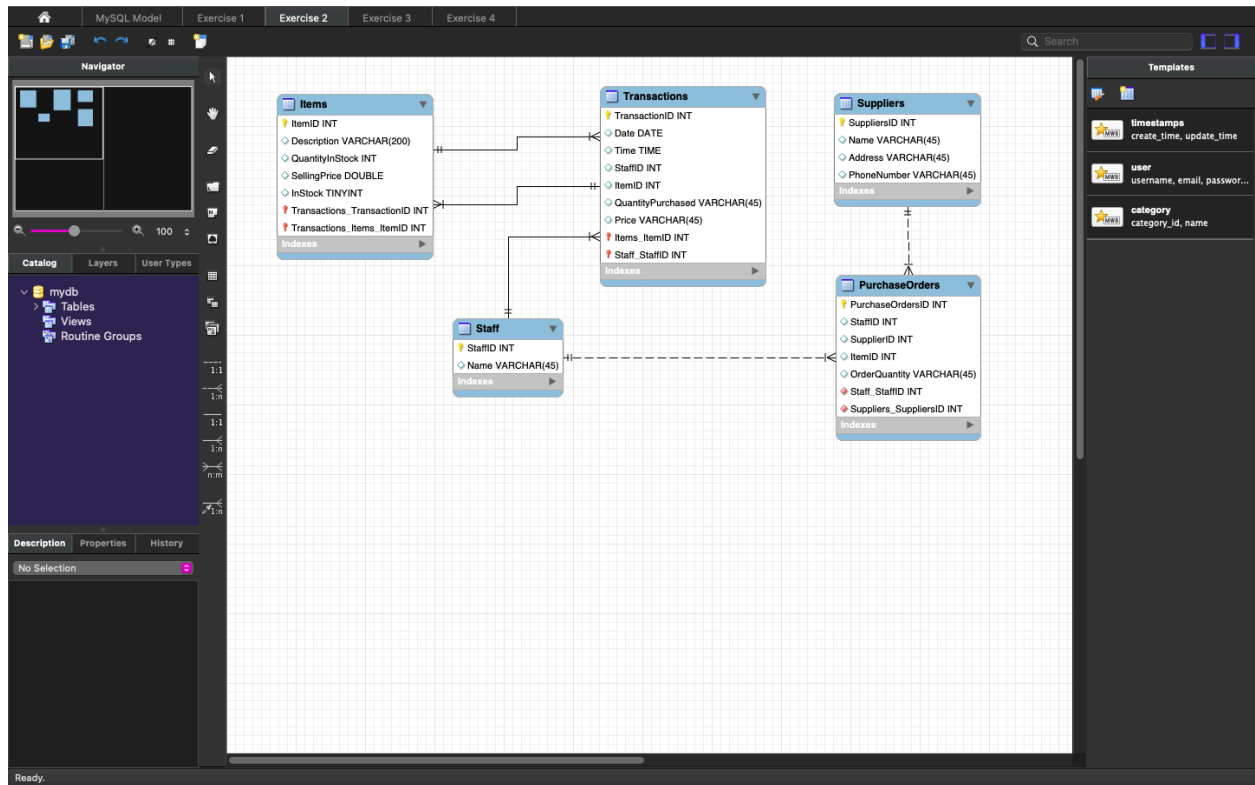
Column Name	Data Type	Constraints
loan_ID	INT	PRIMARY KEY
membership_number	INT	FOREIGN KEY (references Members)
ISBN	VARCHAR(13)	FOREIGN KEY (references Books)
due_date	DATE	NOT NULL

Members Table: There are no transitive dependencies, as all non-primary key attributes (name, address, contact_number, date_of_birth) depend directly on the primary key (membership_number).

Books Table: There are no transitive dependencies, as all non-primary key attributes (title, author, publisher, print_date, number_of_copies) depend directly on the primary key (ISBN).

Loans Table: No transitive dependencies, since the only non-primary attribute is due_date, which depends directly on the composite key.

Exercise 2



Tables

Items Table

Column Name	Data Type	Constraints
item_ID	INT	PRIMARY KEY
description	VARCHAR(255)	NOT NULL
quantity_in_stock	INT	NOT NULL
selling_price	DECIMAL(10,2)	NOT NULL

Transactions Table

Column Name	Data Type	Constraints
transaction_ID	INT	PRIMARY KEY
date	DATE	NOT NULL
time	TIME	NOT NULL
staff_ID	INT	FOREIGN KEY (references Staff)
item_ID	INT	FOREIGN KEY (references Items)
quantity_purchased	INT	NOT NULL
price	DECIMAL(10,2)	NOT NULL

Staff Table

Column Name	Data Type	Constraints
staff_ID	INT	PRIMARY KEY
name	VARCHAR(255)	NOT NULL

Suppliers Table

Column Name	Data Type	Constraints
supplier_ID	INT	PRIMARY KEY
name	VARCHAR(255)	NOT NULL
address	VARCHAR(255)	NOT NULL
phone_number	VARCHAR(20)	NOT NULL

Purchase Orders Table

Column Name	Data Type	Constraints
purchase_order_ID	INT	PRIMARY KEY
staff_ID	INT	FOREIGN KEY (references Staff)
supplier_ID	INT	FOREIGN KEY (references Suppliers)
item_ID	INT	FOREIGN KEY (references Items)
order_quantity	INT	NOT NULL

Items: No transitive dependencies exist. Each attribute is directly dependent on the primary key (item_ID).

Transactions: All attributes are dependent only on the transaction_ID. There are no attributes depending on other non-key attributes.

Staff: No transitive dependencies here.

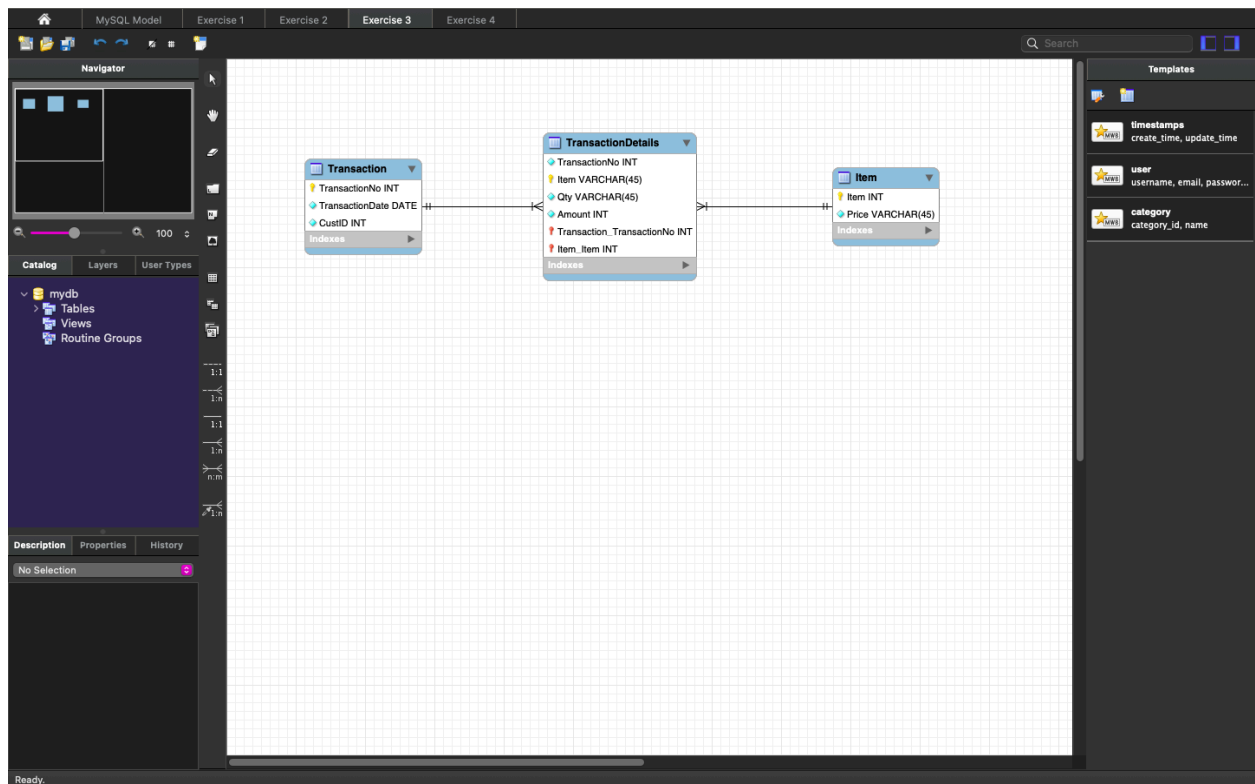
Suppliers: No transitive dependencies.

Purchase Orders: All attributes (staff_ID, supplier_ID, item_ID, order_quantity) depend directly on the purchase_order_ID. No transitive dependencies exist.

Exercise 3

Table 1 Non-normalised relation

TransactionNo	Transaction Date	CustID	Item	Qty	Amount
T10	01/10/2014	4	Paint	2	60
T10	01/10/2014	4	Paint Brushes	4	30
T30	02/10/2014	9	Sealer	1	25



Normalize

Convert to 1NF (First Normal Form)

1NF Definition: A table is in 1NF if it contains only atomic values (i.e., no repeating groups or arrays) and each entry contains a single value.

Tables in 1NF:

In the given table, all the values are atomic (no repeating groups or multi-valued attributes). Therefore, the table is already in 1NF.

Convert to 2NF (Second Normal Form)

Tables:

Transaction Table (2NF)

TransactionNo	Transaction Date	CustID
T10	01/10/2014	4
T30	02/10/2014	9

TransactionDetails Table (2NF)

TransactionNo	Item	Qty	Amount
T10	Paint	2	60
T10	Paint Brushes	4	30
T30	Sealer	1	25

2NF Definition:

1. It is already in 1NF.
2. All non-key attributes depend on the whole primary key, not just a part of it.

Identifying the Primary Key:

- Here, the **TransactionNo** alone is not unique, as it repeats for different items in the same transaction.
- A composite primary key is required: **(TransactionNo, Item)** can uniquely identify each row.

Partial Dependency Issue:

- **CustID** and **TransactionDate** depend only on **TransactionNo**, not on the entire composite key.

- These attributes violate 2NF because they depend only on part of the primary key.

Solution:

- Split the table into two:
 1. **Transaction Table**: This will store transaction-level information.
 2. **Transaction Details Table**: This will store item-specific information for each transaction.

Tables in 2NF:

1. **Transaction Table**:
 - **TransactionNo** (Primary Key)
 - **TransactionDate**
 - **CustID**
2. **Transaction Details Table**:
 - **TransactionNo** (Foreign Key)
 - **Item** (Composite Primary Key)
 - **Qty**
 - **Amount**

Convert to 3NF (Third Normal Form)

Tables:

Transaction Table

TransactionNo	Transaction Date	CustID
T10	01/10/2014	4
T30	02/10/2014	9

TransactionDetails Table

TransactionNo	Item	Qty
T10	Paint	2
T10	Paint Brushes	4
T30	Sealer	1

Item Table

Item	Price
Paint	30
Paint Brushes	7.5
Sealer	25

3NF Definition:

1. It is already in 2NF.
2. There are no transitive dependencies (i.e., non-key attributes do not depend on other non-key attributes).

Transitive Dependency Issue:

- **Amount** depends on **Qty** and the price of the item, meaning that the **Item** has an implicit price that is not stored in the table.
- This creates a transitive dependency.

Solution:

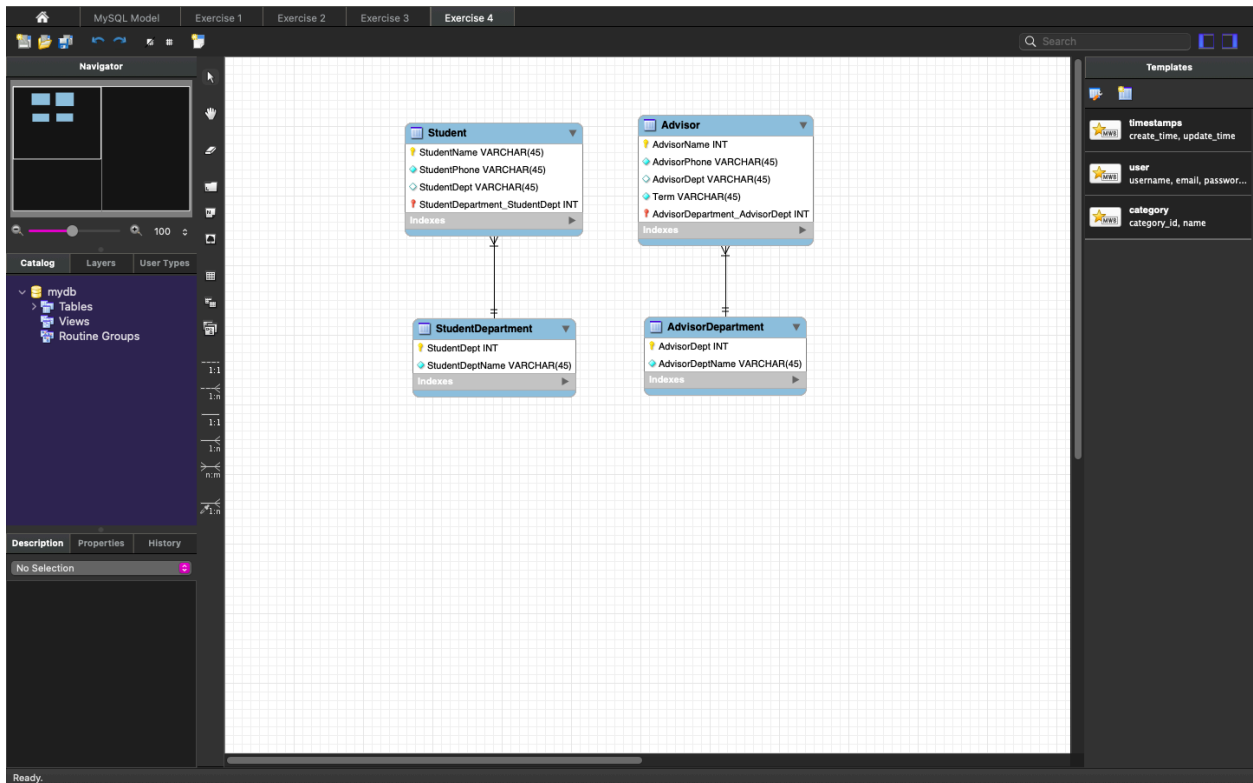
- Create an **Item Table** to store the price of each item separately.

Tables in 3NF:

1. **Transaction Table:**
 - **TransactionNo** (Primary Key)
 - **TransactionDate**
 - **CustID**
2. **Transaction Details Table:**
 - **TransactionNo** (Foreign Key)
 - **Item** (Composite Primary Key)
 - **Qty**
3. **Item Table:**
 - **Item** (Primary Key)
 - **Price**

Exercise 4

StudentName	StudentPhone	StudentDept	StudentDeptName	AdvisorName	AdvisorPhone	AdvisorDept	Term
Tolstoy	593-3824	21	English	Caplice	253-3233	ESD	Fall
Thoreau	644-2343	21	English	Caplice	253-3233	ESD	Fall
James	534-2534	21	English	Lapide	253-1111	ESD	Fall
Woolf	643-5436	18	Mathematics	Toomre	253-6322	Mathematics	Spring
Shakespeare	634-6344	8	Physics	Smith	253-8453	Physics	Spring
Pushkin	534-9832	7	Biology	Griffith	253-9833	Bio Engr	Spring



Normalize

Step 1: 1NF (First Normal Form)

In 1NF, we need to ensure that:

1. All attributes contain atomic values (i.e., no multi-valued or composite attributes).
2. There are no repeating groups.

The given table appears to already be in 1NF, as all attributes have atomic values, and there are no repeating groups.

Table:

StudentName	StudentPhone	StudentDept	StudentDeptName	AdvisorName	AdvisorPhone	AdvisorDept	Term
Tolstoy	593-3824	21	English	Caplice	253-3233	ESD	Fall
Thoreau	644-2343	21	English	Caplice	253-3233	ESD	Fall
James	534-2534	21	English	Lapide	253-1111	ESD	Fall
Woolf	643-5436	18	Mathematics	Toomre	253-6322	Mathematics	Spring
Shakespeare	634-6344	8	Physics	Smith	253-8453	Physics	Spring
Pushkin	534-9832	7	Biology	Griffith	253-9833	Bio Engr	Spring

Step 2: 2NF (Second Normal Form)

In 2NF, we remove partial dependencies, meaning every non-key attribute must depend on the entire primary key. If the primary key is composite, no part of it should determine non-key attributes.

We can identify potential partial dependencies:

- Attributes such as StudentPhone, StudentDept, and StudentDeptName depend only on StudentName.
- Similarly, attributes like AdvisorName, AdvisorPhone, AdvisorDept, and Term depend on AdvisorName.

To bring this into 2NF, we can split the table into two tables: Student and Advisor.

2NF Tables:

Student Table

StudentName	StudentPhone	StudentDept	StudentDeptName
Tolstoy	593-3824	21	English
Thoreau	644-2343	21	English
James	534-2534	21	English
Woolf	643-5436	18	Mathematics
Shakespeare	634-6344	8	Physics

Pushkin	534-9832	7	Biology	
---------	----------	---	---------	--

Advisor Table

AdvisorName	AdvisorPhone	AdvisorDept	Term	
-----	-----	-----	-----	
Caplice	253-3233	ESD	Fall	
Lapide	253-1111	ESD	Fall	
Toomre	253-6322	Mathematics	Spring	
Smith	253-8453	Physics	Spring	
Griffith	253-9833	Bio Engr	Spring	

Step 3: 3NF (Third Normal Form)

In 3NF, we remove transitive dependencies. A transitive dependency occurs when a non-key attribute depends on another non-key attribute.

Here, in the Student Table, StudentDeptName depends on StudentDept rather than directly on StudentName. Similarly, in the Advisor Table, AdvisorDept might depend on AdvisorName. To resolve this, we can further normalize the table by creating separate tables for Departments.

Final 3NF Design:

Student Table

StudentName	StudentPhone	StudentDept	
-----	-----	-----	
Tolstoy	593-3824	21	
Thoreau	644-2343	21	
James	534-2534	21	
Woolf	643-5436	18	
Shakespeare	634-6344	8	
Pushkin	534-9832	7	

Advisor Table

AdvisorName	AdvisorPhone	AdvisorDept	Term	
-----	-----	-----	-----	
Caplice	253-3233	21	Fall	
Lapide	253-1111	21	Fall	
Toomre	253-6322	18	Spring	
Smith	253-8453	8	Spring	
Griffith	253-9833	7	Spring	

StudentDepartment Table

StudentDept	StudentDeptName	
-----	-----	
21	English	

18	Mathematics	
8	Physics	
7	Biology	

AdvisorDepartment Table

AdvisorDept	AdvisorDeptName	
-----	-----	
21	ESD	
18	Mathematics	
8	Physics	
7	Bio Engr	