

INFS 7901 Database Principle

Project Proposal

Name: Shuo Yuan

Student Number: 46920348

Email: s4692034@student.uq.edu.au

Project Description



An online bookstore system acts as a central database that contains various books in stock along with their titles, creators and selling prices. The domains of this database include the entities Customer, Shopping Basket, Warehouse, Books, Publisher, Author along with their relation. The main types are VARCHAR, CHAR, NUMERIC, INTEGER.

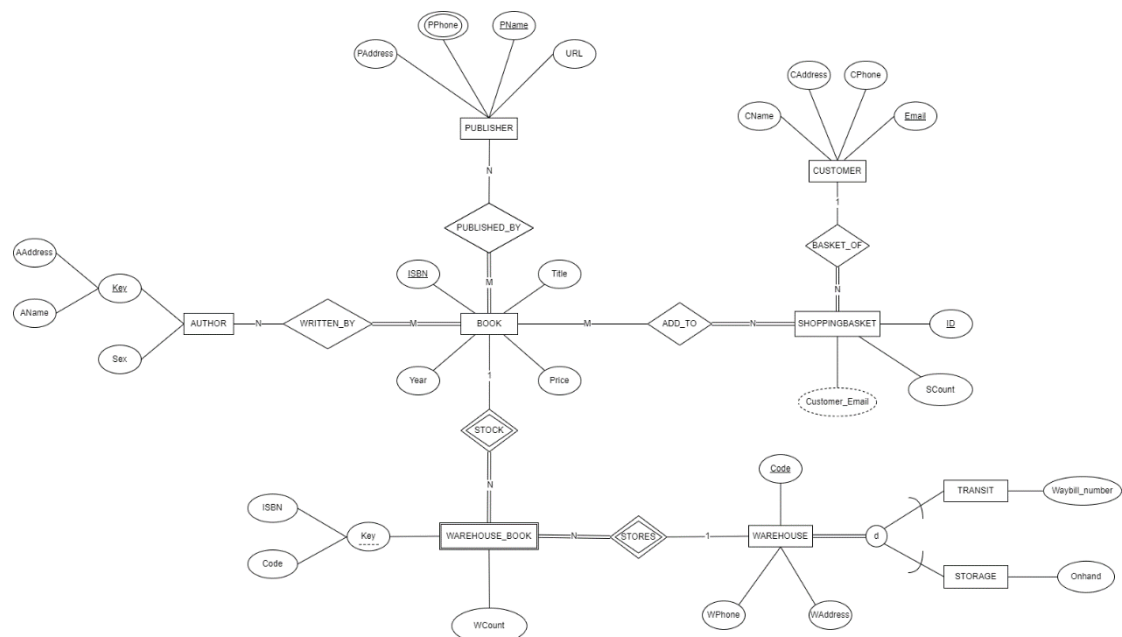
The system should manage information about books in the bookstore, registered users and books they have ordered. The database stores various book related details. A user visiting the website can see a wide range of books arranged in respective categories. The user may select desired book and view its price. The user may even search for specific books on the website. Once the user selects a book, he then has to add that into a shopping basket and the warehouse should prepare the delivery process for the user. The system has the following three main components:

1. Implement of new user to register and login
2. Implement user to choose any book
3. Implement the user to buy books



The platforms I plan for the final project be on would be MySQL + Python, I may use Flask to create a front-end for the convenience.

E/R Diagram



Schema

Relations:

- AUTHOR [AName: varchar(40), AAddress: varchar(100), Sex: char(2)]
- PUBLISHER [PName: varchar(40), URL: varchar(100), PAddress: varchar(100)]
- BOOK [ISBN: varchar(20), Year: year(4), Title: varchar(100), Price: numeric(7, 2)]
- WAREHOUSE [Code: integer(10), WPhone: varchar(40), WAddress: varchar(100)]
- CUSTOMER [Email: varchar(40), CPhone: varchar(40), CAddress: varchar(100), CName: varchar(40)]
- SHOPPINGBASKET [ID: integer(5), SCount: integer(5), Email: varchar(100)]
- WAREHOUSE_BOOK [ISBN: varchar(20), Code: integer(10), WCount: integer(10)]
- ADD_TO [ISBN: varchar(20), ID: integer(5)]
- PUBLISHED_BY [ISBN: varchar(20), PName: varchar(40)]
- WRITTEN_BY [ISBN: varchar(20), AName: varchar(40), AAddress: varchar(100)]
- PUBLISHER_PHONE [PName: varchar(40), PPhone: char(15)]
- TRANSIT [Code: integer(10), Waybill_number: integer(10)]
- STORAGE [Code: integer(10), Onhand: integer(10)]

Foreign Key:

- WAREHOUSE_BOOK.ISBN → BOOK.ISBN
- WAREHOUSE_BOOK.Code → WAREHOUSE.Code
- SHOPPINGBASKET.Email → CUSTOMER.Email
- ADD_TO.ISBN → BOOK.ISBN
- ADD_TO.ID → SHOPPINGBASKET.ID
- PUBLISHED_BY.ISBN → BOOK.ISBN
- PUBLISHED_BY.PName → PUBLISHER.PName
- WRITTEN_BY.ISBN → BOOK.ISBN
- WRITTEN_BY.AName → AUTHOR.AName
- PUBLISHER_PHONE.PName → PUBLISHER.PName
- TRANSIT.Code → WAREHOUSE.Code
- STORAGE.Code → WAREHOUSE.Code

Functional Dependencies



1. Email \rightarrow {CName, CAddress, CPhone}
“Email” is the primary key of CUSTOMR, only one account can be created for one Email. So the system could use the Email to get other information that the customer filled in when creating the account, which means Email can determine all other attribute of CUSTOMER.
2. ID \rightarrow {SCount, Email}
“ID” is the primary key of SHOPPINGBASKET. As we all know, the shopping basket can contain many items, so it needs an ID to identify them, and at the same time, this ID can be used to know how many of the same items (SCount) have been added to the basket by the customer. Besides, once we know the ID, we can trace which Email owns this shopping basket.
3. ID \rightarrow {CName, CAddress, CPhone}
This is an implicit FD, since Email \rightarrow {CName, CAddress, CPhone}
4. AAddress, AName \rightarrow {Sex}
“AName” and “AAddress” respectively mean Author Name and Author Address. Obviously, there are many renamed people in this world, so we can't rely on the author's name alone to determine if this author wrote a certain book, and by the same token the author's address may yield many names, so we have to combine the two as the primary key.
5. PName \rightarrow {PPhone, PAddress, URL}
“PName” means Publisher Name. The publisher's name is not the same as the author's name: the publisher's name will not be repeated, so as long as you know the publisher's name you will know the publisher's contact number, address and their website.
6. ISBN \rightarrow {Title, Year, Price, AAddress, AName, PName}
“ISBN” is an abbreviation of “International Standard Book Number”, which is a numeric commercial book identifier that is intended to be unique. We can view the author, publisher, title, year of publication and price of the book stored in the system by using the ISBN code.
7. ISBN \rightarrow {Sex, PPhone, PAddress, URL}
This is an implicit FD, since AAddress, AName \rightarrow {Sex} and PName \rightarrow {PPhone, PAddress, URL}
8. Code \rightarrow {WPhone, WAddress, Waybill_number, Onhand}
“Code” is the primary key of WAREHOUSE. Each warehouse has its own code, the system can use this code to know the warehouse address and warehouse phone, and can distinguish the role of this warehouse, if it is a transit warehouse, then you can know the waybill number under processing, if it is a storage warehouse, you can know the onhand information.
9. ISBN, Code \rightarrow {WCount}
The combination of “ISBN” and “Code” yields the stock of the book to be found in the warehouse.

Normalized Schema



Based on what we have done in the ER Diagram and Functional Dependencies part, we can combine all the relation table into one universal R and 9 FDs, so the next step is to decompose R into BCNF:

- Relation: R (Email, CName, CAddress, CPhone, PName, PPhone, PAddress, URL, AAddress, AName, Sex, ISBN, Title, Year, Price, ID, SCount, Code, WPhone, WAddress, Waybill_number, Onhand, Wcount)
 - Minimal keys: (Email, PName, AAddress, AName, ISBN, ID, Code)
1. Considering: $\text{Email} \rightarrow \{\text{CName}, \text{CAddress}, \text{CPhone}\}$, Email is not a super key in R, decomposing R into:
R1(Email, CName, CAddress, CPhone) R2(Email, PName, PPhone, PAddress, URL, AAddress, AName, Sex, ISBN, Title, Year, Price, ID, SCount, Code, WPhone, WAddress, Waybill_number, Onhand, Wcount)
 2. Considering: $\text{ID} \rightarrow \{\text{SCount}\}$, ID is not a super key in R2, decomposing R2 into:
R3(ID, SCount) R4(Email, PName, PPhone, PAddress, URL, AAddress, AName, Sex, ISBN, Title, Year, Price, ID, Code, WPhone, WAddress, Waybill_number, Onhand, Wcount)
 3. Considering: $\text{AAddress}, \text{AName} \rightarrow \{\text{Sex}\}$, AAddress, AName is not a super key in R4, decomposing R4 into:
R5(AAddress, AName, Sex) R6(Email, PName, PPhone, PAddress, URL, AAddress, AName, ISBN, Title, Year, Price, ID, Code, WPhone, WAddress, Waybill_number, Onhand, Wcount)
 4. Considering: $\text{ISBN} \rightarrow \{\text{Title}, \text{Year}, \text{Price}\}$, ISBN is not a super key in R6, decomposing R6 into:
R7(ISBN, Title, Year, Price) R8(Email, PName, PPhone, PAddress, URL, AAddress, AName, ISBN, ID, Code, WPhone, WAddress, Waybill_number, Onhand, Wcount)
 5. Considering: $\text{PName} \rightarrow \{\text{PPhone}, \text{PAddress}, \text{URL}\}$, PName is not a super key in R8, decomposing R8 into:
R9(PName, PPhone, PAddress, URL) R10(Email, PName, AAddress, AName, ISBN, ID, Code, WPhone, WAddress, Waybill_number, Onhand, Wcount)
 6. Considering: $\text{Code} \rightarrow \{\text{WPhone}, \text{WAddress}, \text{Waybill_number}, \text{Onhand}\}$, Code is not a super key in R10, decomposing R10 into:
R11(Code, WPhone, WAddress, Waybill_number, Onhand) R12(Email, PName, AAddress, AName, ISBN, ID, Code, Wcount)
 7. Considering: $\text{ISBN}, \text{Code} \rightarrow \{\text{WCount}\}$, ISBN, Code is not a super key in R12, decomposing R12 into:
R13(ISBN, Code, WCount) R14(Email, PName, AAddress, AName, ISBN, ID, Code)

After the decomposition, we have R1, R3, R5, R7, R8, R9, R11, R13, R14. I'm going to rename and put them in one table:

Normalized Schema	Primary Key	Foreign Key
CUSTOMER (Email, CName, CAddress, CPhone)	Email	None
SHOPPINGBASKET (ID, SCount)	ID	None
AUTHOR (AAddress, AName, Sex)	AAddress, AName	None
BOOK (ISBN, Title, Year, Price)	ISBN	None
PUBLISHER (PName, PPhone, PAddress, URL)	PName, PPhone	None
WAREHOUSE (Code, WPhone, WAddress, Waybill_number, Onhand)	Code	None
WAREHOUSE_BOOK (ISBN, Code, WCount)	ISBN, Code (Foreign key as Primary key)	ISBN, Code
R14(Email, PName, AAddress, AName, ISBN, ID, Code)	Email, PName, AAddress, AName, ISBN, ID, Code	None

SQL Dump

```
CREATE TABLE `AUTHOR` (  
  `AName` varchar(40) NOT NULL,  
  `AAddress` varchar(100) NOT NULL,  
  `Sex` char(2) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
INSERT INTO `AUTHOR` (`AName`, `AAddress`, `Sex`) VALUES  
(  
'Anthony Molinaro', '9279 Johnson Ave. New Orleans, LA 70115', 'M'),  
(  
'Chris Date', '844 Primrose Drive Waterford, MI 48329', 'M'),  
(  
'Joel Grus', 'San Antonio, Texas, United States', 'M'),  
(  
'John M. Zelle', 'Department of Math/CS/Physics Wartburg College 100  
Wartburg Blvd.Waverly, IA 50677', 'M'),  
(  
'Tony Gaddis', '645 Chestnut St, Alto Pass, IL 62905', 'M');
```

```
CREATE TABLE `BOOK` (  
  `ISBN` varchar(20) NOT NULL,  
  `Year` year DEFAULT NULL,  
  `Title` varchar(100) DEFAULT NULL,  
  `Price` decimal(7,2) DEFAULT NULL,  
  `PublisherName` varchar(40) NOT NULL,  
  `AuthorName` varchar(40) NOT NULL,  
  `AuthorAddress` varchar(100) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
INSERT INTO `BOOK` (`ISBN`, `Year`, `Title`, `Price`, `PublisherName`,  
`AuthorName`, `AuthorAddress`) VALUES  
(  
'9780321197849', 2003, 'An Introduction to Database Systems',  
'141.10', 'Pearson', 'Chris Date', '844 Primrose Drive Waterford, MI  
48329'),  
(  
'9780596009762', 2006, 'SQL Cookbook: Query Solutions and Techniques  
for Database Developers', '69.20', 'O\'Reilly Media', 'Anthony  
Molinaro', '9279 Johnson Ave. New Orleans, LA 70115'),  
(  
'9781491901427', 2015, 'Data Science from Scratch: First Principles  
with Python', '48.49', 'O\'Reilly Media', 'Joel Grus', 'San Antonio,  
Texas, United States'),  
(  
'9781590282755', 2016, 'Python Programming: An Introduction to  
Computer Science', '45.00', 'O\'Reilly Media', 'John M. Zelle',  
'Department of Math/CS/Physics Wartburg College 100 Wartburg  
Blvd.Waverly, IA 50677'),  
(  
'9789353066888', 2019, 'STARTING OUT WITH PYTHON', '44.51', 'Franklin,  
Beedle & Associates Inc', 'Tony Gaddis', '645 Chestnut St, Alto Pass,  
IL 62905');
```

```
CREATE TABLE `CUSTOMER` (
  `Email` varchar(40) NOT NULL,
  `CName` varchar(40) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci
  DEFAULT NULL,
  `CAddress` varchar(100) DEFAULT NULL,
  `CPhone` varchar(40) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
INSERT INTO `CUSTOMER` (`Email`, `CName`, `CAddress`, `CPhone`) VALUES
('fsalierno@tubidu.com', 'Izabelle Gruffud', '9576 Alderwood Dr.Glen
Allen, VA 23059', '276-200-1771'),
('happy400@tchoeo.com', 'Balbinus Ortwin', '8232 Winchester Street
Hackettstown, NJ 07840', '201-200-3543'),
('henrikgasten@asistx.net', 'Tewodros Birgitte', '9416 Wayne Ave.
Beltsville, MD 20705', '240-200-3623'),
('ricky31b@devii.site', 'Abd al-Qadir Brighid', '7112 Sunset Drive
Skokie, IL 60076', '217-200-3621'),
('vasuchin@bbtspage.com', 'Aina Shamash', '9885 Maple Street New
Britain, CT 06051', '203-200-9841');
```

```
CREATE TABLE `PUBLISHER` (
  `PName` varchar(40) NOT NULL,
  `URL` varchar(100) DEFAULT NULL,
  `PAddress` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
INSERT INTO `PUBLISHER` (`PName`, `URL`, `PAddress`) VALUES
('Franklin, Beedle & Associates Inc', 'https://fbeedle.com/',
'Franklin, Beedle & Associates Inc.10350 N Vancouver Way #5012 Portland
Oregon 97217 United States'),
('O\'Reilly Media', 'https://www.oreilly.com/', '1005 Gravenstein
Highway North Sebastopol, CA 95472 USA'),
('Pearson', 'https://www.pearson.com/', '1001/151 Castlereagh St,
Sydney NSW 2000, Australia'),
('Relx', 'https://www.relx.com/', '230 Park Avenue New York NY 10169
USA'),
('Thomson Reuters', 'https://www.thomsonreuters.com/en.html', '19
Harris Street Pyrmont Sydney 2009 Australia');
```

```
CREATE TABLE `PUB_PHONE` (
  `Pub_Name` varchar(40) NOT NULL,
  `Phone` char(15) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```



```

INSERT INTO `PUB_PHONE` (`Pub_Name`, `Phone`) VALUES
('Franklin, Beedle & Associates Inc', '800-322-2665'),
('O\Reilly Media', '827-7000'),
('Pearson', '1800 882 385'),
('Relx', '212 309 8100'),
('Thomson Reuters', '2 9171 7100');

```

```

CREATE TABLE `SHOPPINGBASKET` (
  `ID` int NOT NULL,
  `SCount` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

```

INSERT INTO `SHOPPINGBASKET` (`ID`, `SCount`) VALUES
(1, 3),
(2, 4),
(3, 1),
(4, 2),
(5, 1);

```

```

CREATE TABLE `WAREHOUSE` (
  `Code` int NOT NULL,
  `WPhone` varchar(40) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci
  DEFAULT NULL,
  `WAddress` varchar(100) DEFAULT NULL,
  `Waybill_number` int DEFAULT NULL,
  `Onhand` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

```

INSERT INTO `WAREHOUSE` (`Code`, `WPhone`, `WAddress`,
`Waybill_number`, `Onhand`) VALUES
(1001, '217-206-9523', '990 Cypress St. Orland Park, IL 60462', 547,
NULL),
(1002, '201-205-6282', '9645 Colonial Court Bergenfield, NJ 07621',
903, NULL),
(1003, '319-200-9841', '190 Logan St. Cedar Falls, IA 50613', 1475,
NULL),
(2001, '218-201-7348', '9264 Hilldale Court Lakeville, MN 55044', NULL,
3533),
(2002, '212-200-4034', '8712 East Howard Lane Huntington, NY 11743',
NULL, 5633);

```

```

CREATE TABLE `WAREHOUSE_BOOK` (
  `Warehouse_Code` int NOT NULL,

```

```

        `Book_ISBN` varchar(20) NOT NULL,
        `WCount` int DEFAULT NULL
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `WAREHOUSE_BOOK` (`Warehouse_Code`, `Book_ISBN`, `WCount`)
VALUES
(2001, '9780321197849', 654),
(2001, '9780596009762', 861),
(2002, '9781491901427', 4561),
(2002, '9781590282755', 3861),
(2002, '9789353066888', 999);

ALTER TABLE `AUTHOR`
    ADD PRIMARY KEY (`AName`,`AAddress`);

ALTER TABLE `BOOK`
    ADD PRIMARY KEY (`ISBN`),
    ADD KEY `PublisherName` (`PublisherName`),
    ADD KEY `AuthorInfo` (`AuthorName`,`AuthorAddress`);

ALTER TABLE `CUSTOMER`
    ADD PRIMARY KEY (`Email`);

ALTER TABLE `PUBLISHER`
    ADD PRIMARY KEY (`PName`);

ALTER TABLE `PUB_PHONE`
    ADD PRIMARY KEY (`Pub_Name`,`Phone`);

ALTER TABLE `SHOPPINGBASKET`
    ADD PRIMARY KEY (`ID`);

ALTER TABLE `WAREHOUSE`
    ADD PRIMARY KEY (`Code`);

ALTER TABLE `WAREHOUSE_BOOK`
    ADD PRIMARY KEY (`Warehouse_Code`,`Book_ISBN`),
    ADD KEY `Book_ISBN` (`Book_ISBN`);

ALTER TABLE `BOOK`
    ADD CONSTRAINT `BOOK_ibfk_1` FOREIGN KEY (`PublisherName`) REFERENCES
`PUBLISHER` (`PName`) ON DELETE RESTRICT ON UPDATE RESTRICT,

```

```
ADD CONSTRAINT `BOOK_ibfk_2` FOREIGN KEY  
(`AuthorName`,`AuthorAddress`) REFERENCES `AUTHOR` (`AName`,  
`AAddress`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `PUB_PHONE`  
ADD CONSTRAINT `PUB_PHONE_ibfk_1` FOREIGN KEY (`Pub_Name`) REFERENCES  
`PUBLISHER` (`PName`);
```

```
ALTER TABLE `WAREHOUSE_BOOK`  
ADD CONSTRAINT `WAREHOUSE_BOOK_ibfk_1` FOREIGN KEY (`Warehouse_Code`) REFERENCES  
`WAREHOUSE` (`Code`),  
ADD CONSTRAINT `WAREHOUSE_BOOK_ibfk_2` FOREIGN KEY (`Book_ISBN`) REFERENCES  
`BOOK` (`ISBN`);  
COMMIT;
```

Screenshot

- 
- ```
SELECT ISBN FROM BOOK;
```

Find all of the book ISBN
  - ```
SELECT * FROM BOOK WHERE Price > 44.00;
```

Find all books that cost more than 44.00
 - ```
SELECT AVG(Price) FROM BOOK;
```

Calculate the average of book price
  - ```
SELECT COUNT(*) FROM BOOK WHERE Year > 2003;
```

Counts the number of books which published after 2003.
 - ```
SELECT ID, SCount FROM SHOPPINGBASKET S1 WHERE exists (SELECT * FROM SHOPPINGBASKET S2 WHERE S2.SCount = S1.SCount AND S2.ID <> S1.ID);
```

For each shopping basket, check if there is another basket with the same item count.
  - ```
SELECT PName FROM PUBLISHER P WHERE NOT EXISTS (SELECT A.AName, A.AAddress FROM AUTHOR A WHERE NOT EXISTS (SELECT B.AuthorName, B.AuthorAddress FROM BOOK B WHERE B.AuthorName = A.AName AND B.AuthorAddress = A.AAddress AND B.PublisherName=P.PName));
```

Find the publisher's name who have published in all of the books.
 - ```
DELETE FROM SHOPPINGBASKET WHERE ID = 1
```

Delete all shopping basket related to ID 1.
  - ```
DELETE FROM BOOK WHERE Year < 2005 AND Price > 70.00
```

Delete all book which published before 2005 and cost more than 70.00
 - ```
UPDATE BOOK SET Price = 43.21 WHERE Price <= 60.00
```

Set the price of all books below 60.00 to 43.21
  - ```
UPDATE BOOK SET Price = Price + 4.99 WHERE Price < 46.00
```

Increase the book price of all books by 4.99 (should not be more than 46.00)