ASSIGNMENT 2 FRONT SHEET

Qualification	**TEC Level 5 HND Diploma in Computing**		
Unit number and title	**Unit 04: Database Design & Development**		
Submission date	02/11/2020	**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Pham Cao Nguyen	**Student ID**	GCC18074
Class	GCC0801	**Assessor name**	Nguyen Hung Dung
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	CaoNguyen

Grading grid

P2	P3	P4	P5	M2	M3	M4	M5	D2	D3

☼ **Summative Feedback:**

☼ **Resubmission Feedback:**

Grade:	**Assessor Signature:**	**Date:**

Signature & Date:

ASSIGNMENT 1 BRIEF

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number	Unit 04: Database Design & Development		
Assignment title			
Academic Year			
Unit Tutor	Pham Thuy Duong		
Issue date		**Submission date**	
IV name and date			

Submission Format:

Format: This assignment is an Individual assignment and specifically including 2 documents:

(1) *sql file of your code and represent your code to your mananger (tutor)*

(2) *a report document*

You must use font *Calibri size 12, set number of the pages and use multiple line spacing at 1.3. Margins must be: left: 1.25 cm; right: 1 cm; top: 1 cm and bottom: 1 cm.* The reference follows Harvard referencing system. The recommended word limit is *2.000-2.500 words*. You will not be penalized for exceeding the total word limit. The cover page of the report has to be the Assignment front sheet 2.

Submission Students are compulsory to submit the assignment in due date and in a way requested by the Tutors. The form of submission will be a soft copy posted on http://cms.greenwich.edu.vn/

Note: The Assignment *must* be your own work, and not copied by or from another student or from books etc. If you use ideas, quotes or data (such as diagrams) from books, journals or other sources, you must reference your sources, using the Harvard style. Make sure that you know how to reference properly, and that understand the guidelines on plagiarism. *If you do not, you definitely get fail*

Unit Learning Outcomes:

LO2 Develop a fully functional relational database system, based on an existing system design.

LO3 Test the system against user and system requirements.

LO4 Produce technical and user documentation

Assignment Brief and Guidance:

You are employed as a Database Developer for a large IT consultancy company. The company has been approached by FPT university which is expanding due to the growth of the number of students. FPT is currently facing difficulties in dealing with managing the university. It decided to develop several academic systems to manage the university easier including: **Online Library system, Student Grading System, Attendance System, CMS System, Scheduling System, Enrolment Systems, and so on.**

You are tasked to select one of those systems to develop database for FPT university. Your tasks are to:

Work with FPT to find out about current requirements for each system

Analyze the requirements and produce clear statements of user and system requirements.

Design a relational database system using appropriate design tools and techniques

Develop a fully functional relational database system, based on an existing system design.

Test the system against user and system requirements.

Produce technical and user documentation

Part 2 (Assignment 2)

Once the designs have been accepted by your manager you have been asked to:

1. Develop the database system using evidence of user interface, output and data validations and querying across multiple tables.

You want to include more than just the basics so you will implement a fully functional database system which will include system security and database maintenance features.

You have decided to implement a query language into the relational database system. The developed system will be demonstrated to your manager.

Your manager has asked you to include in the report:

2. Assessing whether meaningful data has been extracted through the use of query tools to produce appropriate management information.

3. Evaluating the effectiveness of the database solution in relation to user and system requirements, and suggest improvements.

4. Once the system has been developed, you will test the system and your manager will complete a witness statement indicating how your tests are performing against user and system requirements.

You will produce a brief report assessing the effectiveness of the testing, including an explanation of the choice of test data used.

5. Lastly you will produce technical and user documentation which will be given to the company.

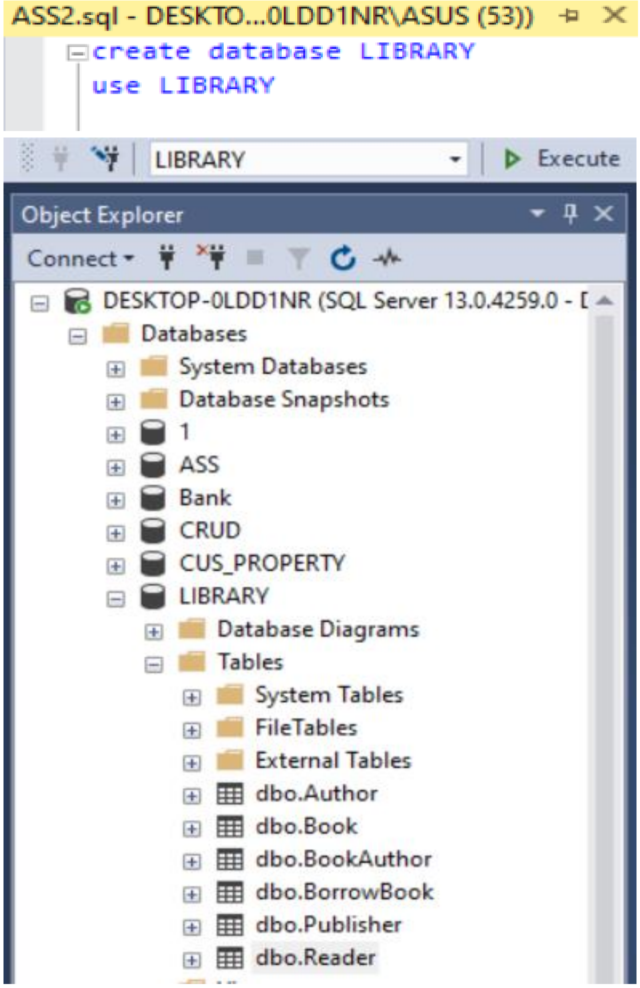
You want to provide some graphical representations for ease of reference in the technical guide, so you have decided to produce a technical and user documentation for a fully functional system, including diagrams showing movement of data through the system, and flowcharts describing how the system works.

Learning Outcomes and Assessment Criteria		
Pass	**Merit**	**Distinction**
LO2 Develop a fully functional relational database system, based on an existing system design. **LO3** Test the system against user and system requirements. **LO4** Produce technical and user documentation		
P2 Develop the database system with evidence of user interface, output and data validations, and querying across multiple tables. **P3** Implement a query language into the relational database system.	**M2** Implement a fully functional database system which includes system security and database maintenance. **M3** Assess whether meaningful data has been extracted through the use of query tools to produce appropriate management information.	**LO2 & 3** **D2** Evaluate the effectiveness of the database solution in relation to user and system requirements, and suggest improvements.
P4 Test the system against user and system requirements.	**M4** Assess the effectiveness of the testing, including an explanation of the choice of test data used.	
P5 Produce technical and user documentation for a fully functional system.	**M5** Explains why, and how, validation/verification were used, or could have been used in your final solution	**D3** Assess any future improvements that may be required to ensure the continued effectiveness of the database system.

Contents

P2 Develop the database system with evidence of user interface, output and data validations, and querying across multiple tables.

- **Code snippets to create each table**

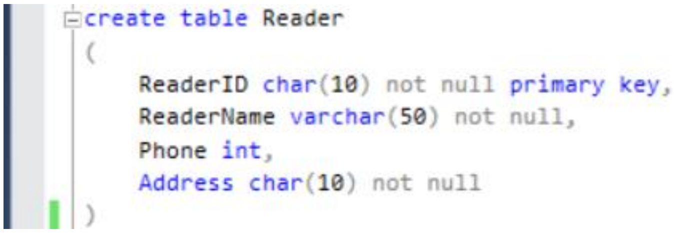


ASS2.sql - DESKTO...0LDD1NR\ASUS (53))

```sql
create database LIBRARY
use LIBRARY
```

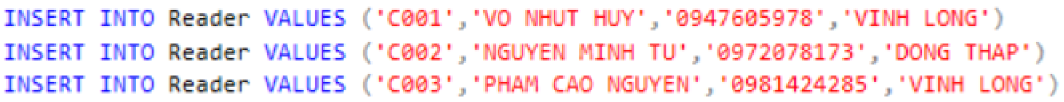
- **Code snippets to insert some sample data for each table**

_ Create table Reader



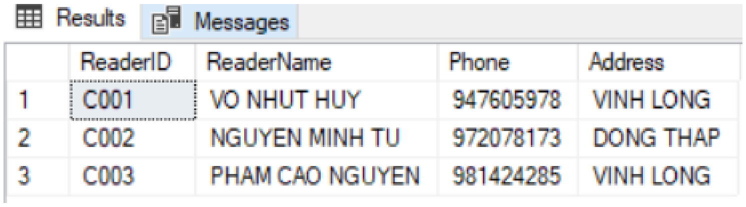
```sql
create table Reader
(
	ReaderID char(10) not null primary key,
	ReaderName varchar(50) not null,
	Phone int,
	Address char(10) not null
)
```

_ Insert information Reader



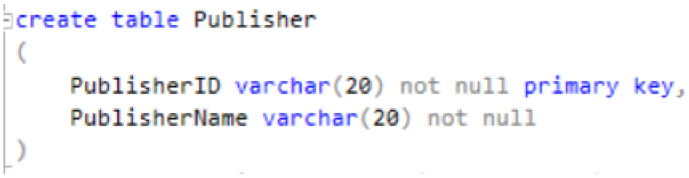
```
INSERT INTO Reader VALUES ('C001','VO NHUT HUY','0947605978','VINH LONG')
INSERT INTO Reader VALUES ('C002','NGUYEN MINH TU','0972078173','DONG THAP')
INSERT INTO Reader VALUES ('C003','PHAM CAO NGUYEN','0981424285','VINH LONG')
```

_ Select * from Reader



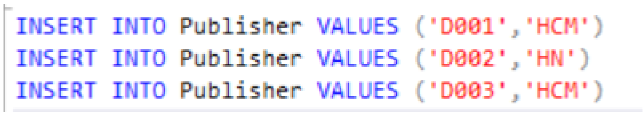
	ReaderID	ReaderName	Phone	Address
1	C001	VO NHUT HUY	947605978	VINH LONG
2	C002	NGUYEN MINH TU	972078173	DONG THAP
3	C003	PHAM CAO NGUYEN	981424285	VINH LONG

_ Create table Publisher



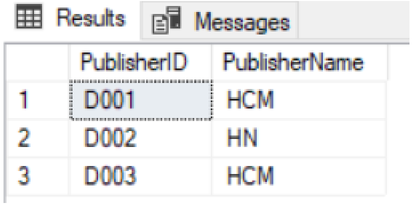
```
create table Publisher
(
    PublisherID varchar(20) not null primary key,
    PublisherName varchar(20) not null
)
```

_ Insert information Publisher



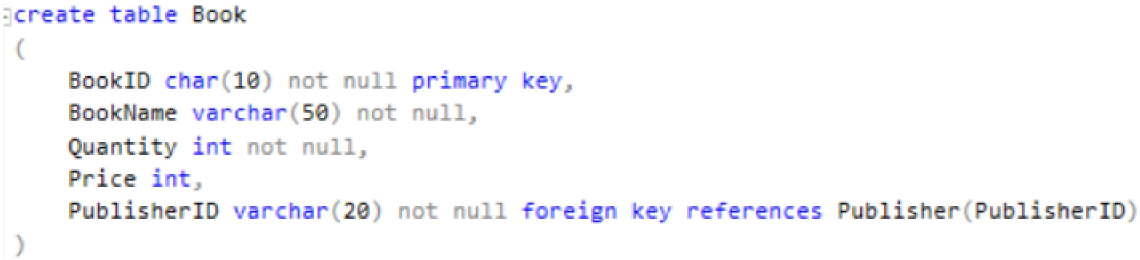
```
INSERT INTO Publisher VALUES ('D001','HCM')
INSERT INTO Publisher VALUES ('D002','HN')
INSERT INTO Publisher VALUES ('D003','HCM')
```

_ Select * from Publisher



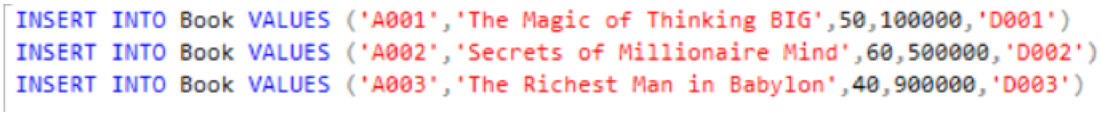
	PublisherID	PublisherName
1	D001	HCM
2	D002	HN
3	D003	HCM

_ Create table Book



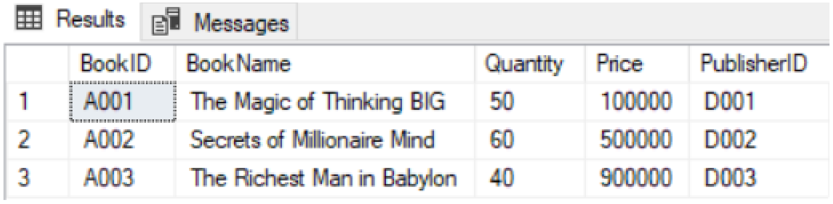
```
create table Book
(
    BookID char(10) not null primary key,
    BookName varchar(50) not null,
    Quantity int not null,
    Price int,
    PublisherID varchar(20) not null foreign key references Publisher(PublisherID)
)
```

_ Insert information Book



```
INSERT INTO Book VALUES ('A001','The Magic of Thinking BIG',50,100000,'D001')
INSERT INTO Book VALUES ('A002','Secrets of Millionaire Mind',60,500000,'D002')
INSERT INTO Book VALUES ('A003','The Richest Man in Babylon',40,900000,'D003')
```

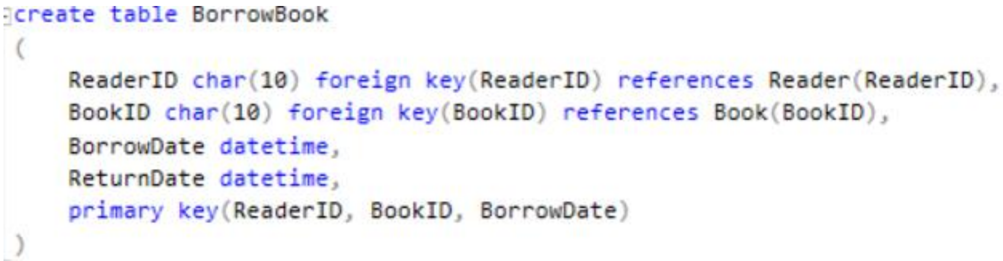
_ Select * from Book



Results | Messages

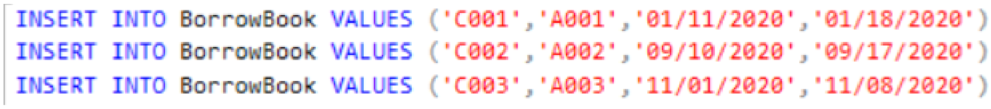
	BookID	BookName	Quantity	Price	PublisherID
1	A001	The Magic of Thinking BIG	50	100000	D001
2	A002	Secrets of Millionaire Mind	60	500000	D002
3	A003	The Richest Man in Babylon	40	900000	D003

_ Create table BorrowBook



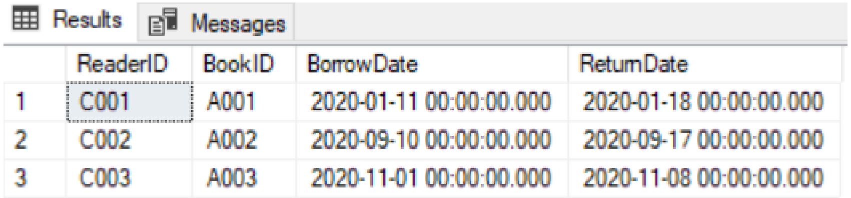
```
create table BorrowBook
(
    ReaderID char(10) foreign key(ReaderID) references Reader(ReaderID),
    BookID char(10) foreign key(BookID) references Book(BookID),
    BorrowDate datetime,
    ReturnDate datetime,
    primary key(ReaderID, BookID, BorrowDate)
)
```

_ Insert information BorrowBook



```
INSERT INTO BorrowBook VALUES ('C001','A001','01/11/2020','01/18/2020')
INSERT INTO BorrowBook VALUES ('C002','A002','09/10/2020','09/17/2020')
INSERT INTO BorrowBook VALUES ('C003','A003','11/01/2020','11/08/2020')
```

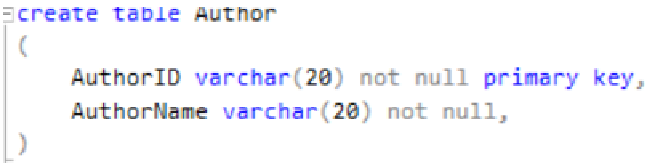
_Select * from BorrowBook



Results | Messages

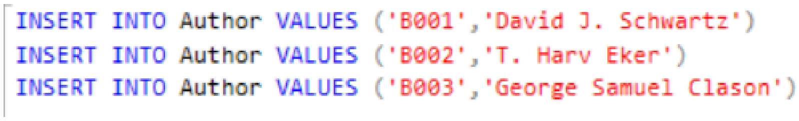
	ReaderID	BookID	BorrowDate	ReturnDate
1	C001	A001	2020-01-11 00:00:00.000	2020-01-18 00:00:00.000
2	C002	A002	2020-09-10 00:00:00.000	2020-09-17 00:00:00.000
3	C003	A003	2020-11-01 00:00:00.000	2020-11-08 00:00:00.000

+ Create table Author



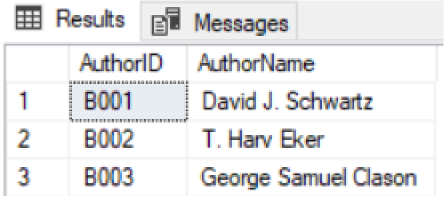
```
create table Author
(
    AuthorID varchar(20) not null primary key,
    AuthorName varchar(20) not null,
)
```

_ Insert information Author



```
INSERT INTO Author VALUES ('B001','David J. Schwartz')
INSERT INTO Author VALUES ('B002','T. Harv Eker')
INSERT INTO Author VALUES ('B003','George Samuel Clason')
```

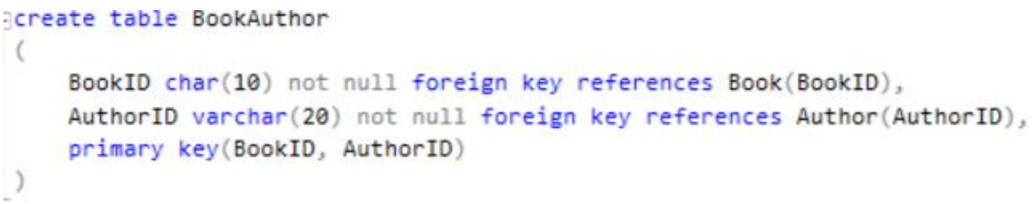
_ Select * from Author



Results | Messages

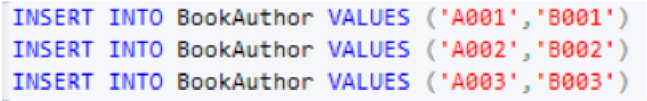
	AuthorID	AuthorName
1	B001	David J. Schwartz
2	B002	T. Harv Eker
3	B003	George Samuel Clason

_ Create table BookAuthor



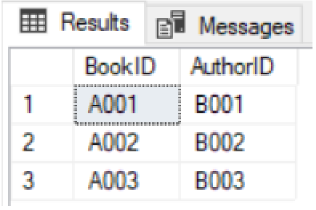
```
create table BookAuthor
(
    BookID char(10) not null foreign key references Book(BookID),
    AuthorID varchar(20) not null foreign key references Author(AuthorID),
    primary key(BookID, AuthorID)
)
```

_ Insert information BookAuthor



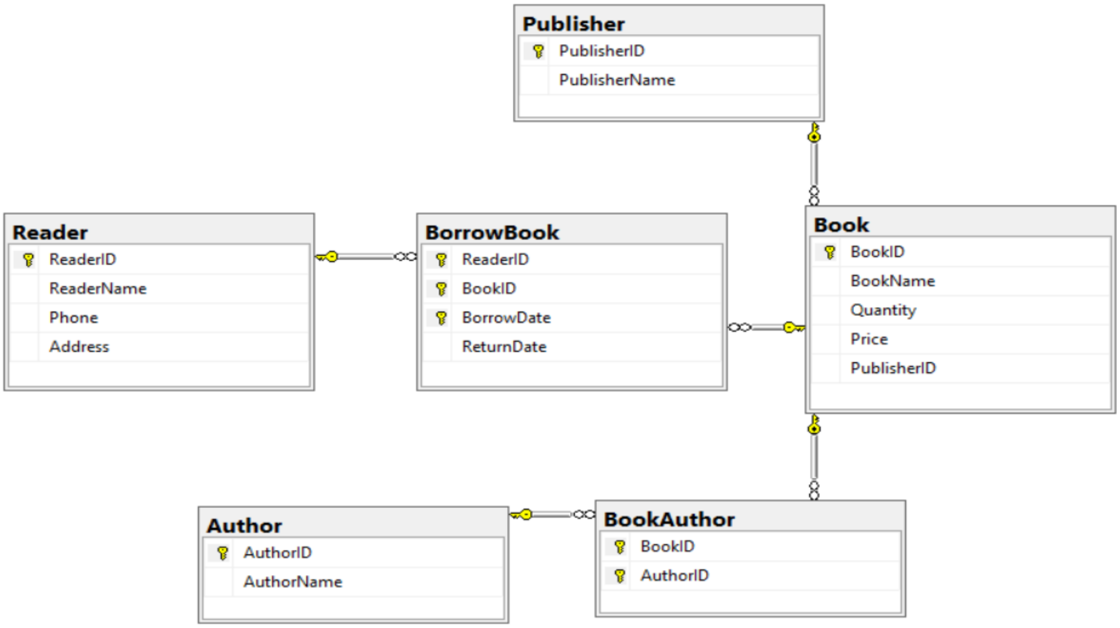
```
INSERT INTO BookAuthor VALUES ('A001','B001')
INSERT INTO BookAuthor VALUES ('A002','B002')
INSERT INTO BookAuthor VALUES ('A003','B003')
```

_ Select * from BookAuthor



	BookID	AuthorID
1	A001	B001
2	A002	B002
3	A003	B003

◆ Generated Database Diagram of your Implementation



Publisher
- PublisherID
- PublisherName

Reader
- ReaderID
- ReaderName
- Phone
- Address

BorrowBook
- ReaderID
- BookID
- BorrowDate
- ReturnDate

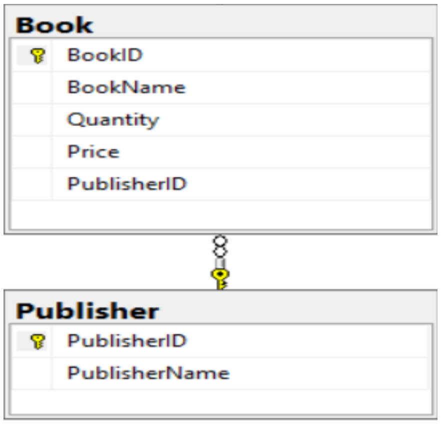
Book
- BookID
- BookName
- Quantity
- Price
- PublisherID

Author
- AuthorID
- AuthorName

BookAuthor
- BookID
- AuthorID

◆ Explanations about any changes comparing to your design (if there is)

- Publisher entity: does not have BookID attribute.
- Book entity: PublisherID property is required.
- The Book table and the Publisher table have a many to one relationship, I have to remove the BookID in the Publisher table, if we let bookID be the primary key in the Publisher table, there is one to many relations between the Book, and the Publisher, because one Publisher provides many Books.

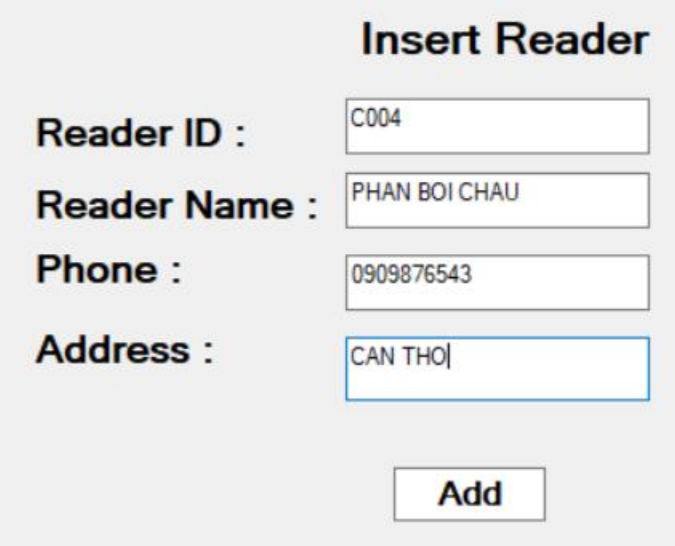


P3 Implement a query language into the relational database system.

- Queries to support the functionalities (insert, update, delete)

◆ [dbo].[Reader]

+ Insert

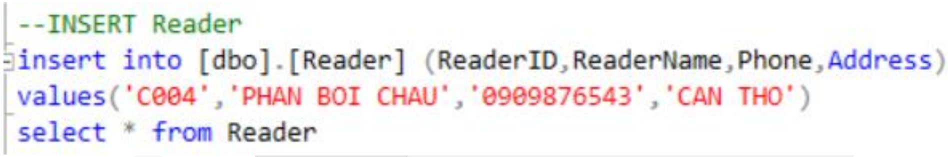


Insert Reader

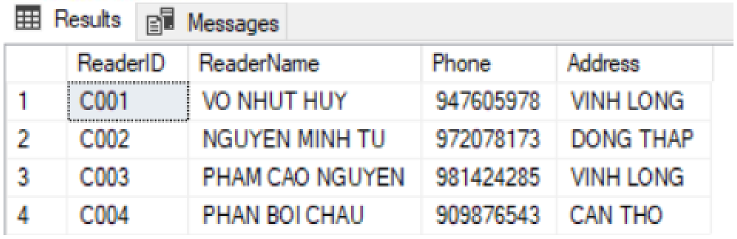
Reader ID :	C004
Reader Name :	PHAN BOI CHAU
Phone :	0909876543
Address :	CAN THO

The user enters Reader information need insert into the textbox and clicks 'Add'.

We will have the command :

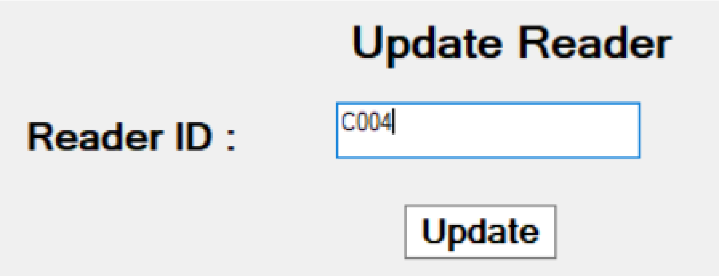


```
--INSERT Reader
insert into [dbo].[Reader] (ReaderID,ReaderName,Phone,Address)
values('C004','PHAN BOI CHAU','0909876543','CAN THO')
select * from Reader
```



	ReaderID	ReaderName	Phone	Address
1	C001	VO NHUT HUY	947605978	VINH LONG
2	C002	NGUYEN MINH TU	972078173	DONG THAP
3	C003	PHAM CAO NGUYEN	981424285	VINH LONG
4	C004	PHAN BOI CHAU	909876543	CAN THO

+ Update

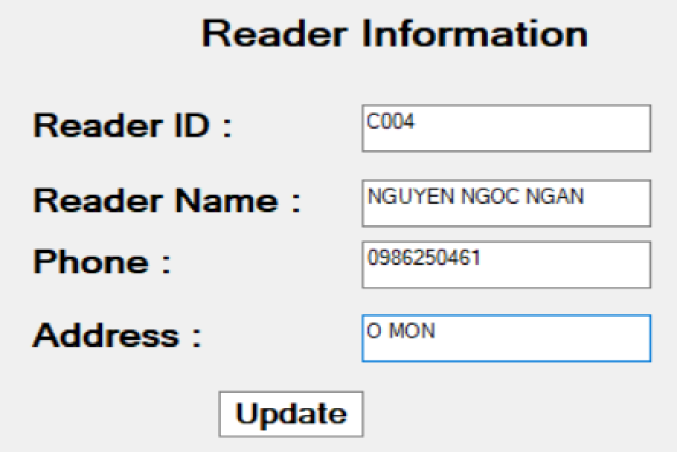


Update Reader

Reader ID : C004

The user enters Reader ID into the textbox and clicks 'update'. Reader information and we will have the command :

SELECT * from Reader
Where ReaderID = ‘C004’



Reader Information

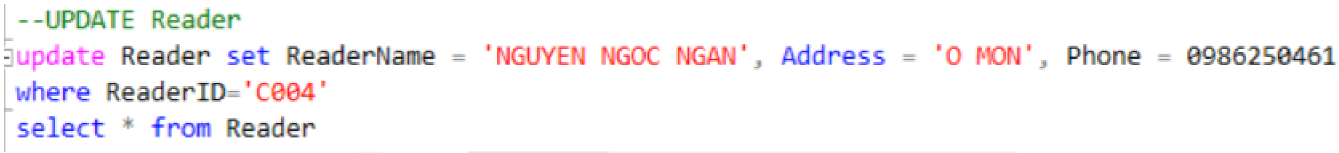
Reader ID : C004

Reader Name : NGUYEN NGOC NGAN

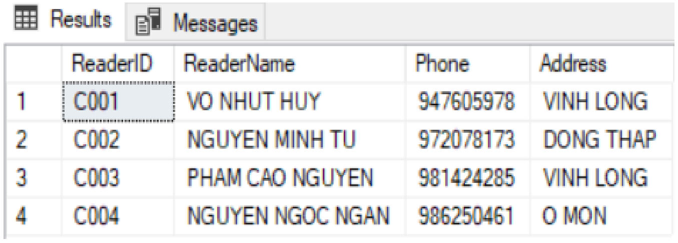
Phone : 0986250461

Address : O MON

The user enters Reader information need update into the textbox and clicks 'update'.
We will have the command :

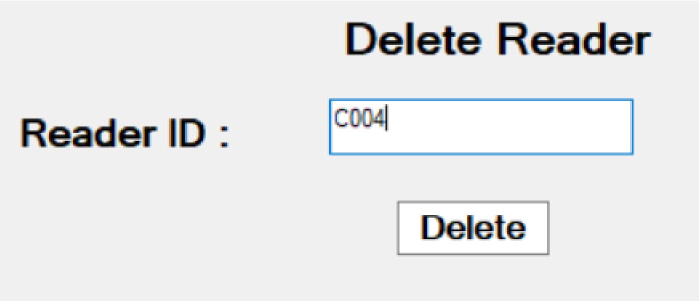


```
--UPDATE Reader
update Reader set ReaderName = 'NGUYEN NGOC NGAN', Address = 'O MON', Phone = 0986250461
where ReaderID='C004'
select * from Reader
```



	ReaderID	ReaderName	Phone	Address
1	C001	VO NHUT HUY	947605978	VINH LONG
2	C002	NGUYEN MINH TU	972078173	DONG THAP
3	C003	PHAM CAO NGUYEN	981424285	VINH LONG
4	C004	NGUYEN NGOC NGAN	986250461	O MON

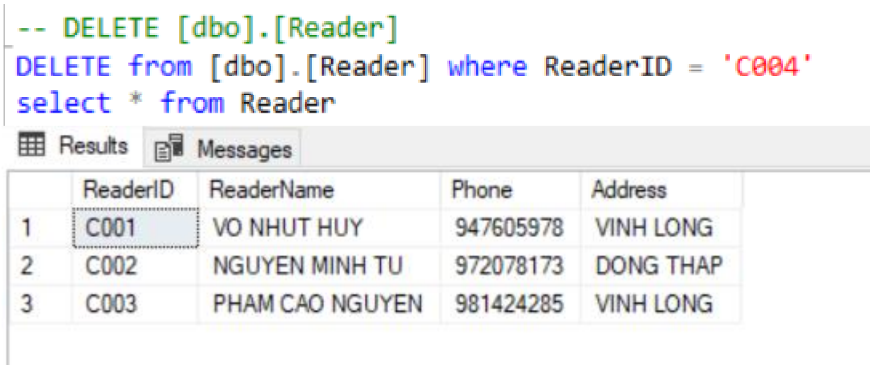
+ Delete



Delete Reader

Reader ID : C004

The user enters ReaderID need delete into the textbox and clicks 'Delete'.

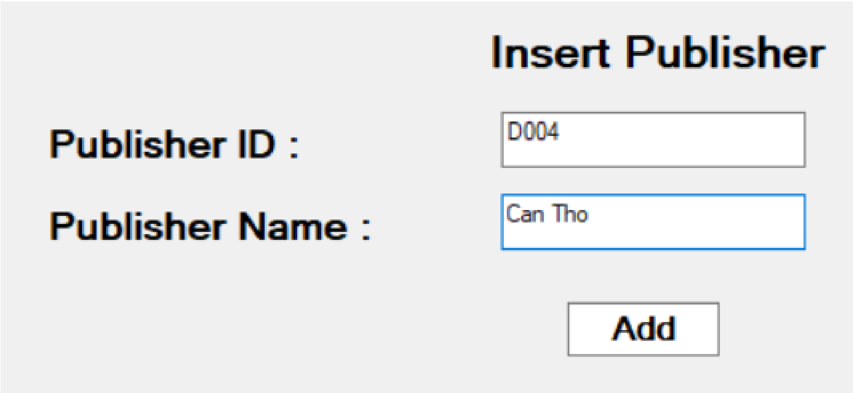


```
-- DELETE [dbo].[Reader]
DELETE from [dbo].[Reader] where ReaderID = 'C004'
select * from Reader
```

	ReaderID	ReaderName	Phone	Address
1	C001	VO NHUT HUY	947605978	VINH LONG
2	C002	NGUYEN MINH TU	972078173	DONG THAP
3	C003	PHAM CAO NGUYEN	981424285	VINH LONG

◆ **[dbo].[Publisher]**

+ **Insert**

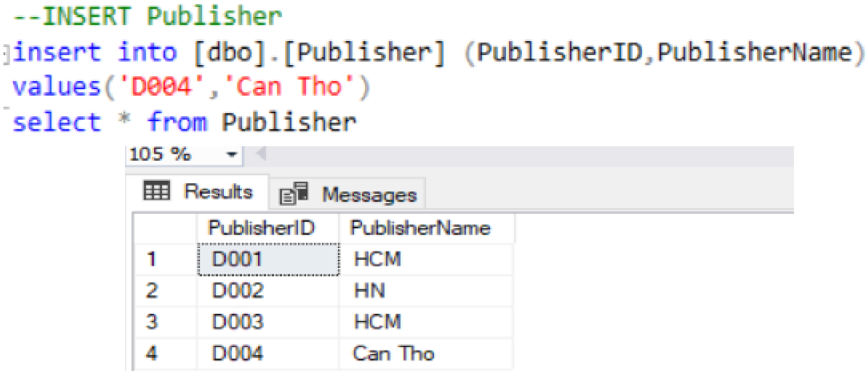


Insert Publisher

Publisher ID : D004

Publisher Name : Can Tho

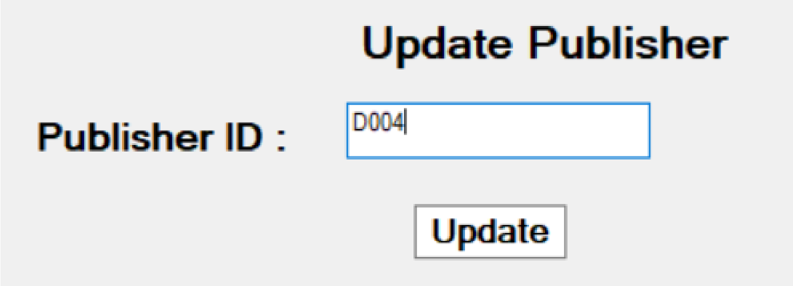
The user enters Publisher ID and Publisher Name need to insert into the textbox and clicks 'Add'. We will have the command :



```
--INSERT Publisher
insert into [dbo].[Publisher] (PublisherID,PublisherName)
values('D004','Can Tho')
select * from Publisher
```

	PublisherID	PublisherName
1	D001	HCM
2	D002	HN
3	D003	HCM
4	D004	Can Tho

+ Update

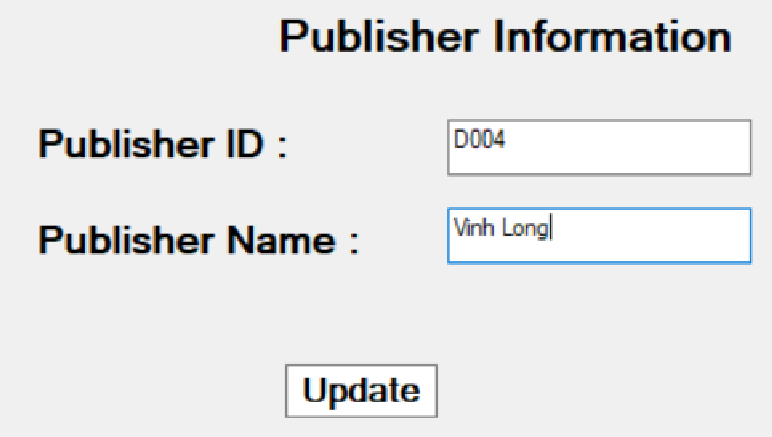


Update Publisher

Publisher ID : D004

The user enters Publisher ID into the textbox and clicks 'update'. Publisher information and we will have the command :

SELECT * from Publisher
Where PublisherID = ‘D004’

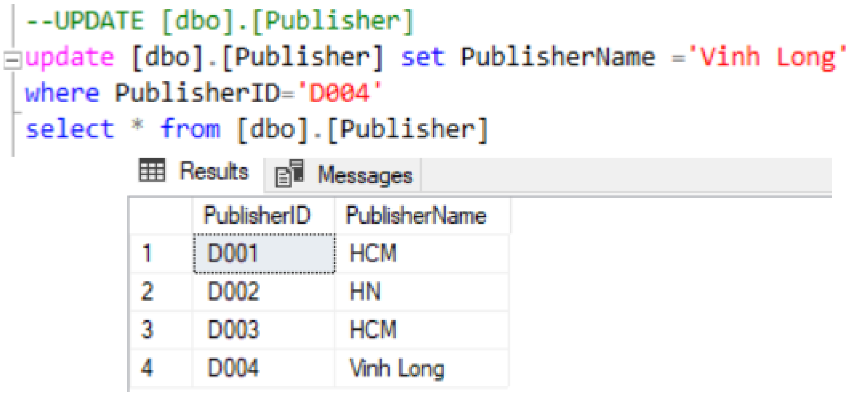


Publisher Information

Publisher ID : D004

Publisher Name : Vinh Long

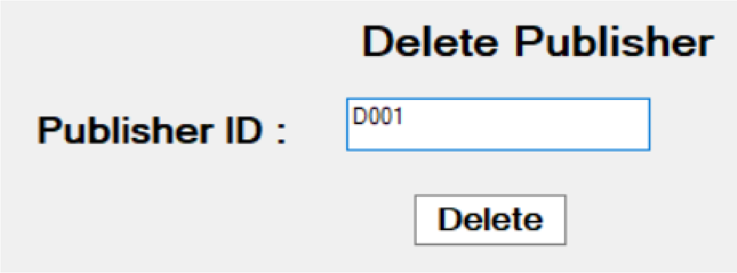
The user enters Publisher information need update into the textbox and clicks 'update'.
We will have the command :



```
--UPDATE [dbo].[Publisher]
update [dbo].[Publisher] set PublisherName ='Vinh Long'
where PublisherID='D004'
select * from [dbo].[Publisher]
```

	PublisherID	PublisherName
1	D001	HCM
2	D002	HN
3	D003	HCM
4	D004	Vinh Long

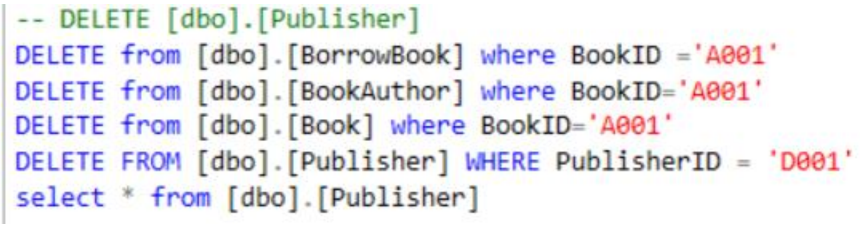
+ Delete



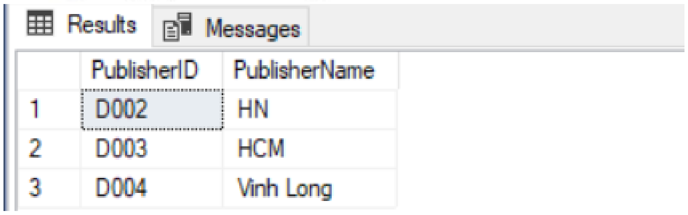
Delete Publisher

Publisher ID : D001

The user enters PublisherID need delete into the textbox and clicks 'Delete'.



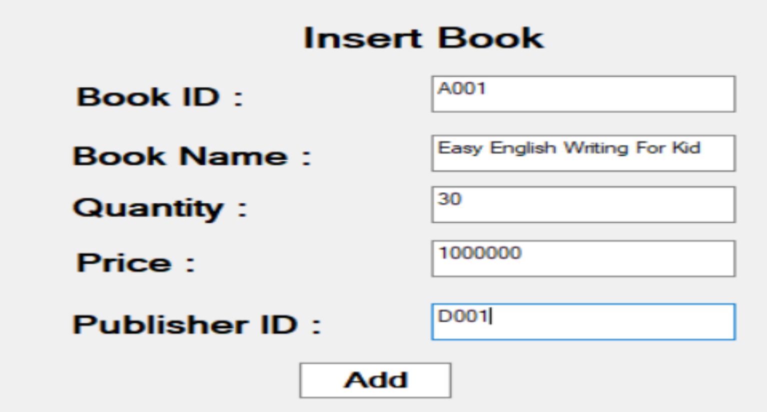
```
-- DELETE [dbo].[Publisher]
DELETE from [dbo].[BorrowBook] where BookID ='A001'
DELETE from [dbo].[BookAuthor] where BookID='A001'
DELETE from [dbo].[Book] where BookID='A001'
DELETE FROM [dbo].[Publisher] WHERE PublisherID = 'D001'
select * from [dbo].[Publisher]
```



	PublisherID	PublisherName
1	D002	HN
2	D003	HCM
3	D004	Vinh Long

- **[dbo].[Book]**

+ Insert



Insert Book

Book ID : A001

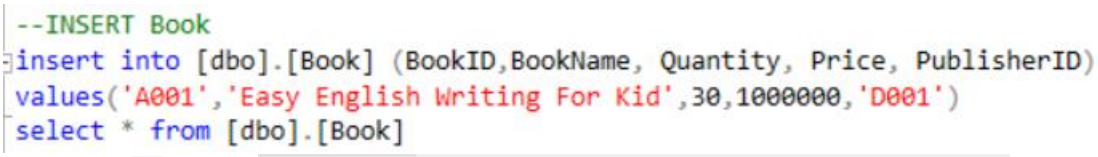
Book Name : Easy English Writing For Kid

Quantity : 30

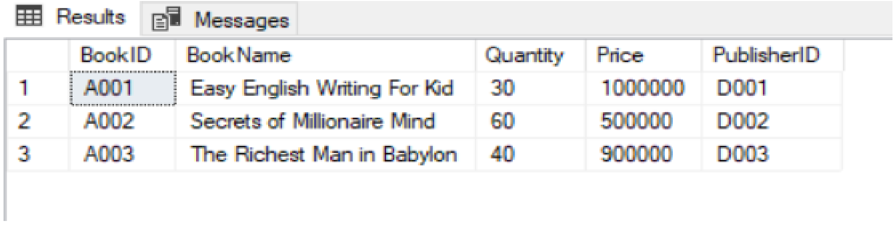
Price : 1000000

Publisher ID : D001

The user enters Book information need to insert and PublisherID into the textbox and clicks 'Add'. We will have the command :

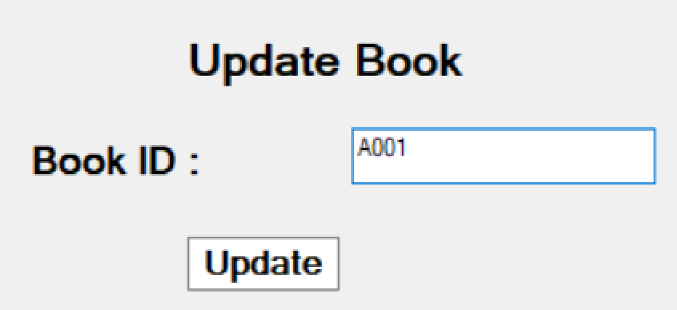


```
--INSERT Book
insert into [dbo].[Book] (BookID,BookName, Quantity, Price, PublisherID)
values('A001','Easy English Writing For Kid',30,1000000,'D001')
select * from [dbo].[Book]
```



	BookID	BookName	Quantity	Price	PublisherID
1	A001	Easy English Writing For Kid	30	1000000	D001
2	A002	Secrets of Millionaire Mind	60	500000	D002
3	A003	The Richest Man in Babylon	40	900000	D003

+ Update

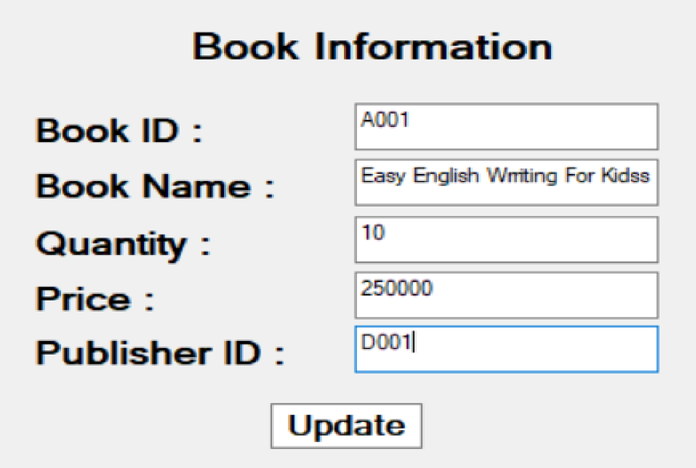


Update Book

Book ID :	A001

The user enters Book ID into the textbox and clicks 'update'. Book information and we will have the command :

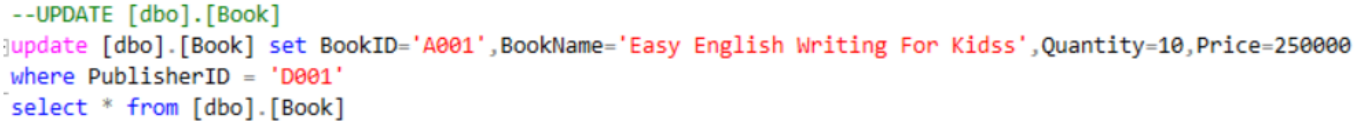
SELECT * from Book
Where BookID = ‘A001’



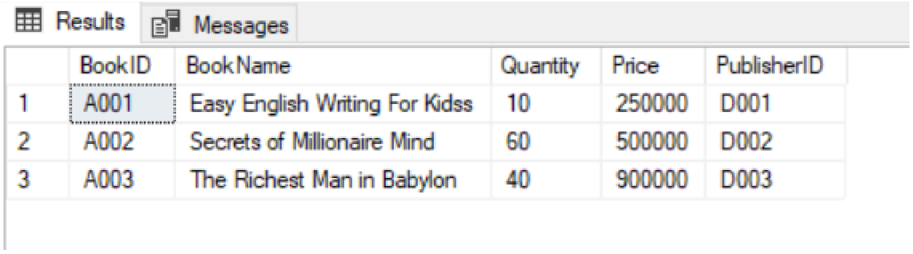
Book Information

Book ID :	A001
Book Name :	Easy English Writing For Kidss
Quantity :	10
Price :	250000
Publisher ID :	D001

The user enters Book information need update into the textbox and clicks 'update'. We will have the command :

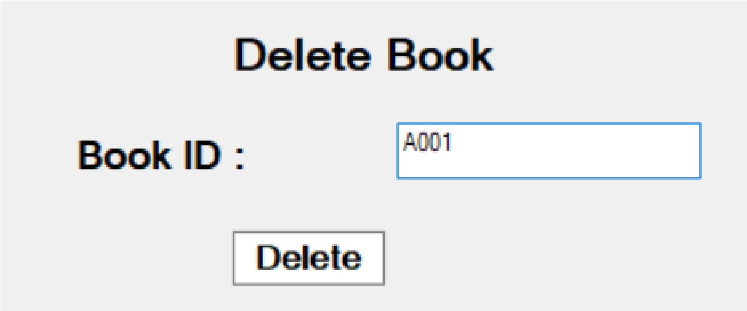


```
--UPDATE [dbo].[Book]
update [dbo].[Book] set BookID='A001',BookName='Easy English Writing For Kidss',Quantity=10,Price=250000
where PublisherID = 'D001'
select * from [dbo].[Book]
```



	BookID	BookName	Quantity	Price	PublisherID
1	A001	Easy English Writing For Kidss	10	250000	D001
2	A002	Secrets of Millionaire Mind	60	500000	D002
3	A003	The Richest Man in Babylon	40	900000	D003

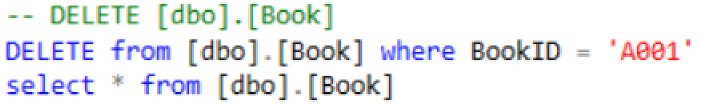
+ Delete



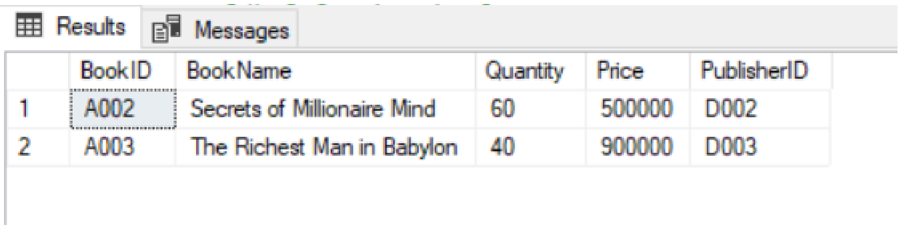
Delete Book

Book ID : A001

The user enters BookID need delete into the textbox and clicks 'Delete'.



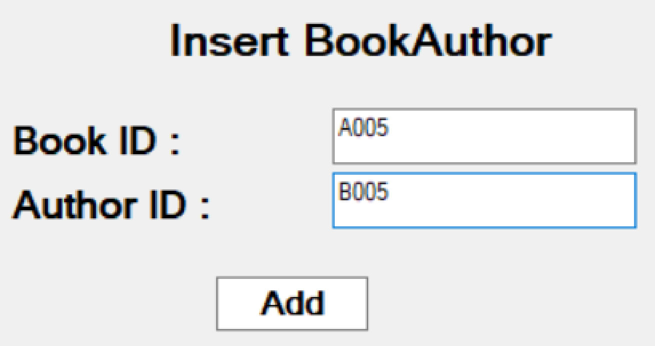
```
-- DELETE [dbo].[Book]
DELETE from [dbo].[Book] where BookID = 'A001'
select * from [dbo].[Book]
```



	BookID	BookName	Quantity	Price	PublisherID
1	A002	Secrets of Millionaire Mind	60	500000	D002
2	A003	The Richest Man in Babylon	40	900000	D003

- ◆ **[dbo].[BookAuthor]**

+ Insert

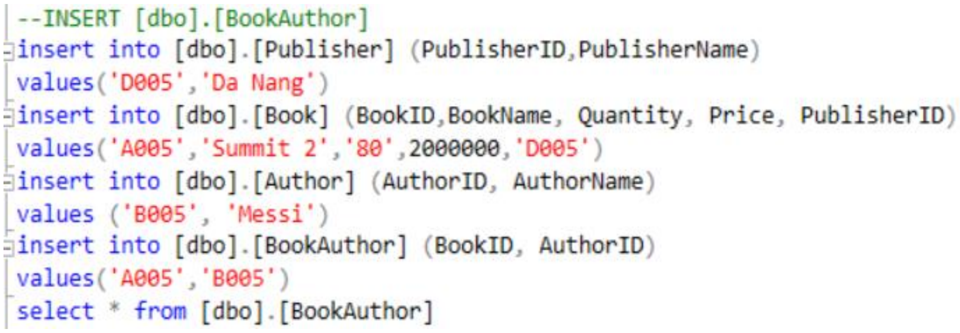


Insert BookAuthor

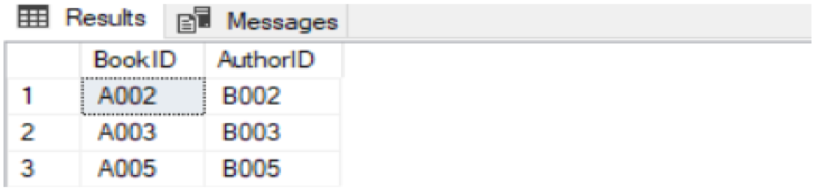
Book ID : A005

Author ID : B005

The user enters Book ID and Author Name needs to insert into the textbox and clicks 'Add'.
We will have the command :

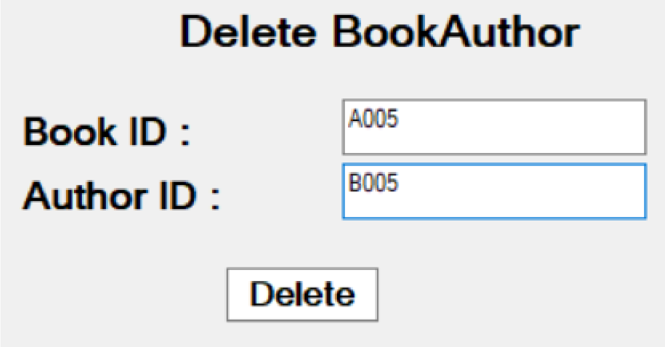


```
--INSERT [dbo].[BookAuthor]
insert into [dbo].[Publisher] (PublisherID,PublisherName)
values('D005','Da Nang')
insert into [dbo].[Book] (BookID,BookName, Quantity, Price, PublisherID)
values('A005','Summit 2','80',2000000,'D005')
insert into [dbo].[Author] (AuthorID, AuthorName)
values ('B005', 'Messi')
insert into [dbo].[BookAuthor] (BookID, AuthorID)
values('A005','B005')
select * from [dbo].[BookAuthor]
```



	BookID	AuthorID
1	A002	B002
2	A003	B003
3	A005	B005

+ Delete

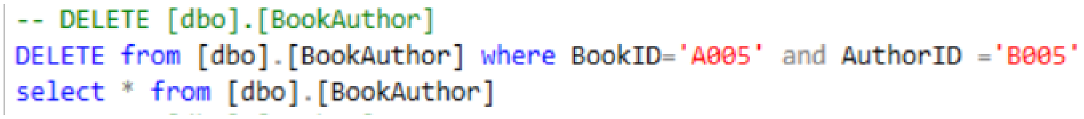


Delete BookAuthor

Book ID : A005

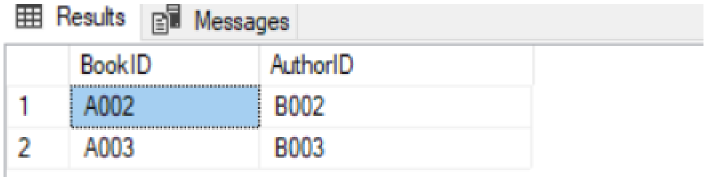
Author ID : B005

The user enters BookID and AuthorID need to delete into the textbox and clicks 'Delete'.



```
-- DELETE [dbo].[BookAuthor]
DELETE from [dbo].[BookAuthor] where BookID='A005' and AuthorID ='B005'
select * from [dbo].[BookAuthor]
```

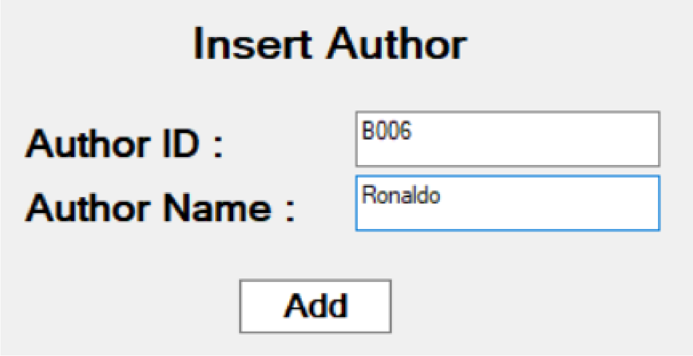
s



	BookID	AuthorID
1	A002	B002
2	A003	B003

◆ **[dbo].[Author]**

+ Insert



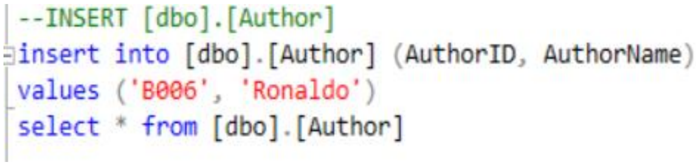
Insert Author

Author ID : B006

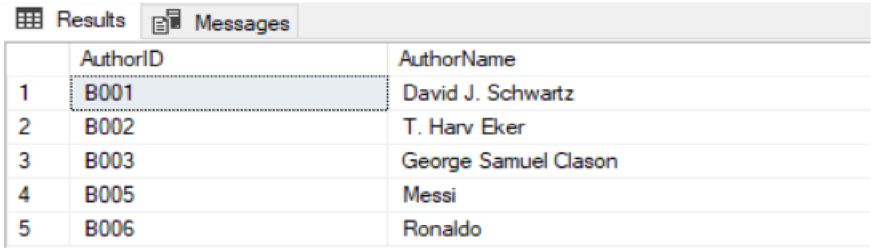
Author Name : Ronaldo

The user enters Author ID and Author Name needs to insert into the textbox and clicks 'Add'.

We will have the command :

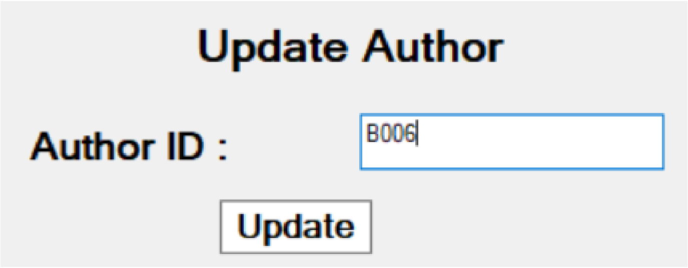


```
--INSERT [dbo].[Author]
insert into [dbo].[Author] (AuthorID, AuthorName)
values ('B006', 'Ronaldo')
select * from [dbo].[Author]
```



	AuthorID	AuthorName
1	B001	David J. Schwartz
2	B002	T. Harv Eker
3	B003	George Samuel Clason
4	B005	Messi
5	B006	Ronaldo

+ Update

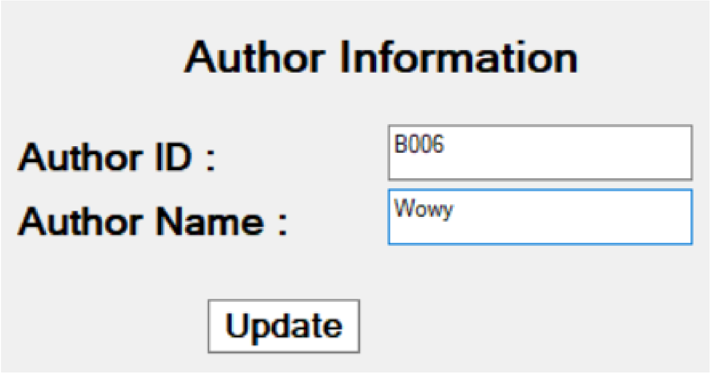


Update Author

Author ID : B006

The user enters Author ID into the textbox and clicks 'update'. Author information and we will have the command :

SELECT * from Author
Where AuthorID = ‘B006’

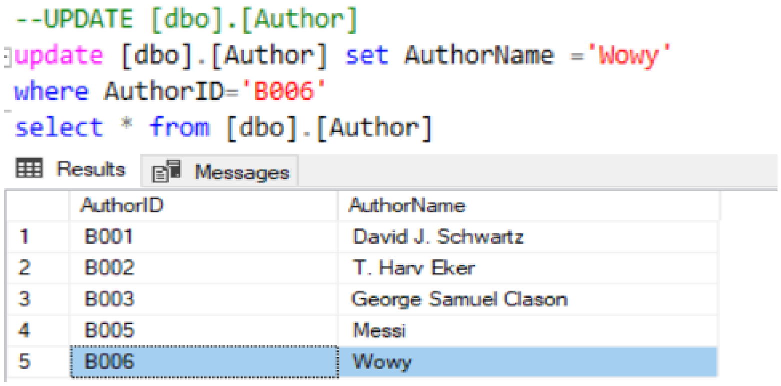


Author Information

Author ID : B006

Author Name : Wowy

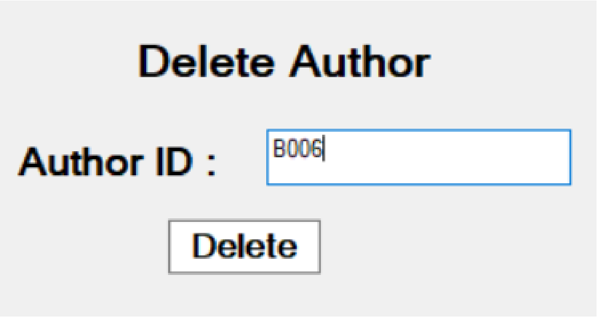
The user enters Author information need update into the textbox and clicks 'update'.
We will have the command :



```
--UPDATE [dbo].[Author]
update [dbo].[Author] set AuthorName ='Wowy'
where AuthorID='B006'
select * from [dbo].[Author]
```

	AuthorID	AuthorName
1	B001	David J. Schwartz
2	B002	T. Harv Eker
3	B003	George Samuel Clason
4	B005	Messi
5	B006	Wowy

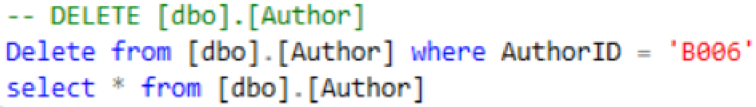
+ Delete



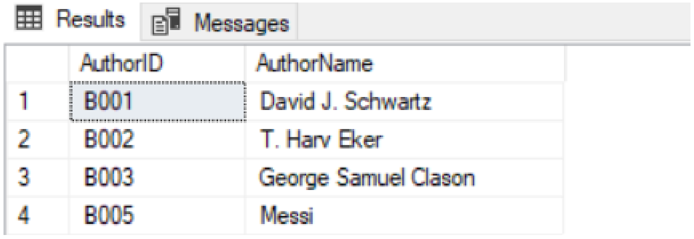
Delete Author

Author ID : B006

The user enters AuthorID need to delete into the textbox and clicks 'Delete'.



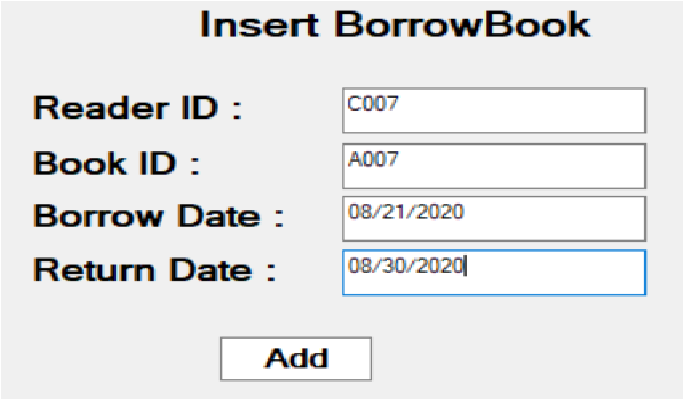
```
-- DELETE [dbo].[Author]
Delete from [dbo].[Author] where AuthorID = 'B006'
select * from [dbo].[Author]
```



	AuthorID	AuthorName
1	B001	David J. Schwartz
2	B002	T. Harv Eker
3	B003	George Samuel Clason
4	B005	Messi

- **[dbo].[BorrowBook]**

+ Insert



Insert BorrowBook

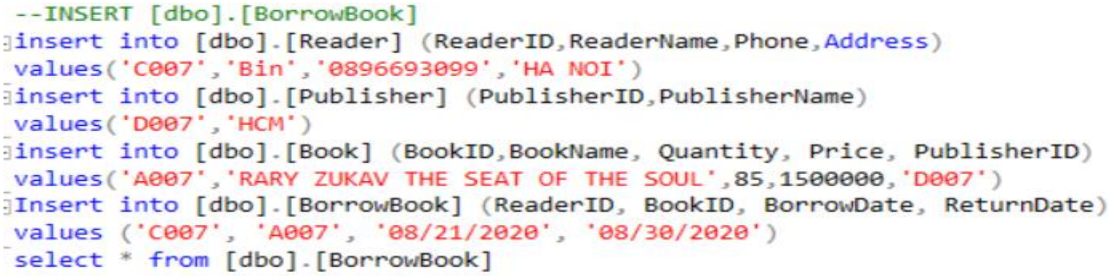
Reader ID : C007

Book ID : A007

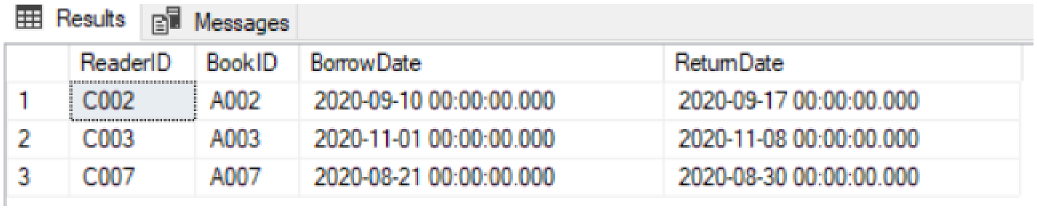
Borrow Date : 08/21/2020

Return Date : 08/30/2020

The user enters Borrow book information needs to insert into the textbox and clicks 'Add'. We will have the command :

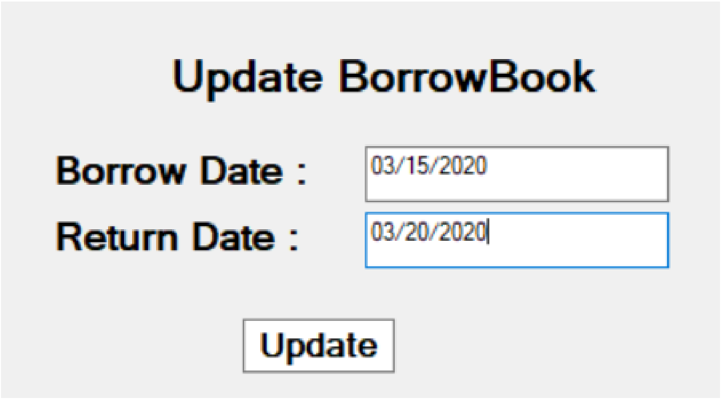


```
--INSERT [dbo].[BorrowBook]
insert into [dbo].[Reader] (ReaderID,ReaderName,Phone,Address)
values('C007','Bin','0896693099','HA NOI')
insert into [dbo].[Publisher] (PublisherID,PublisherName)
values('D007','HCM')
insert into [dbo].[Book] (BookID,BookName, Quantity, Price, PublisherID)
values('A007','RARY ZUKAV THE SEAT OF THE SOUL',85,1500000,'D007')
Insert into [dbo].[BorrowBook] (ReaderID, BookID, BorrowDate, ReturnDate)
values ('C007', 'A007', '08/21/2020', '08/30/2020')
select * from [dbo].[BorrowBook]
```



	ReaderID	BookID	BorrowDate	ReturnDate
1	C002	A002	2020-09-10 00:00:00.000	2020-09-17 00:00:00.000
2	C003	A003	2020-11-01 00:00:00.000	2020-11-08 00:00:00.000
3	C007	A007	2020-08-21 00:00:00.000	2020-08-30 00:00:00.000

+ Update

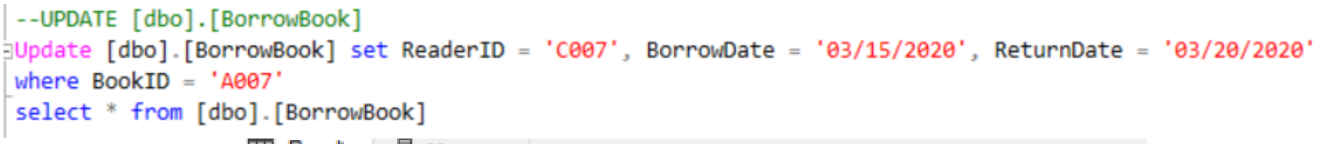


Update BorrowBook

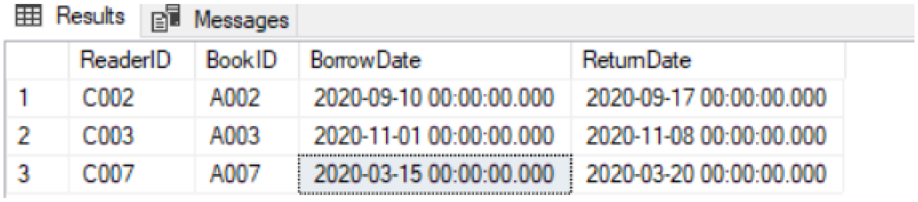
Borrow Date : 03/15/2020

Return Date : 03/20/2020

The user enters Borrow Date and Return Date need update into the textbox and clicks 'update'. We will have the command :

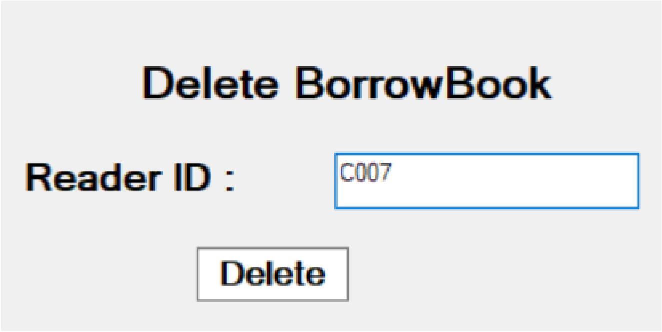


```
--UPDATE [dbo].[BorrowBook]
Update [dbo].[BorrowBook] set ReaderID = 'C007', BorrowDate = '03/15/2020', ReturnDate = '03/20/2020'
where BookID = 'A007'
select * from [dbo].[BorrowBook]
```



	ReaderID	BookID	BorrowDate	ReturnDate
1	C002	A002	2020-09-10 00:00:00.000	2020-09-17 00:00:00.000
2	C003	A003	2020-11-01 00:00:00.000	2020-11-08 00:00:00.000
3	C007	A007	2020-03-15 00:00:00.000	2020-03-20 00:00:00.000

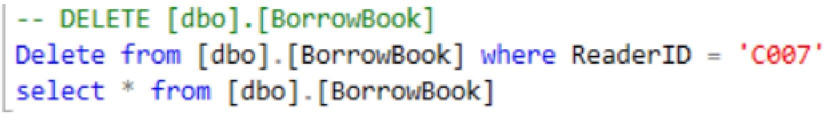
+ Delete



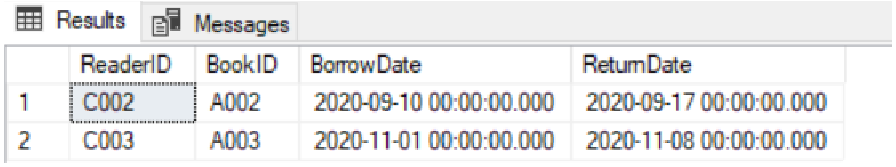
Delete BorrowBook

Reader ID : C007

The user enters ReaderID need to delete into the textbox and clicks 'Delete'.



```
-- DELETE [dbo].[BorrowBook]
Delete from [dbo].[BorrowBook] where ReaderID = 'C007'
select * from [dbo].[BorrowBook]
```

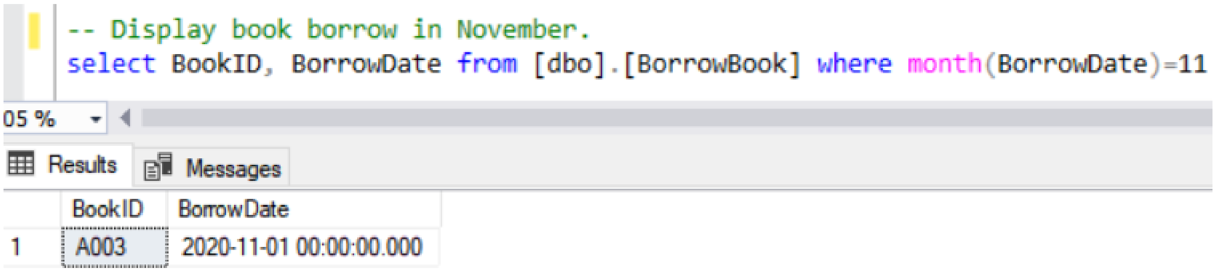


Results | Messages

	ReaderID	BookID	BorrowDate	ReturnDate
1	C002	A002	2020-09-10 00:00:00.000	2020-09-17 00:00:00.000
2	C003	A003	2020-11-01 00:00:00.000	2020-11-08 00:00:00.000

◆ **Report statistical**

- **Report listings to be borrowed book in November**



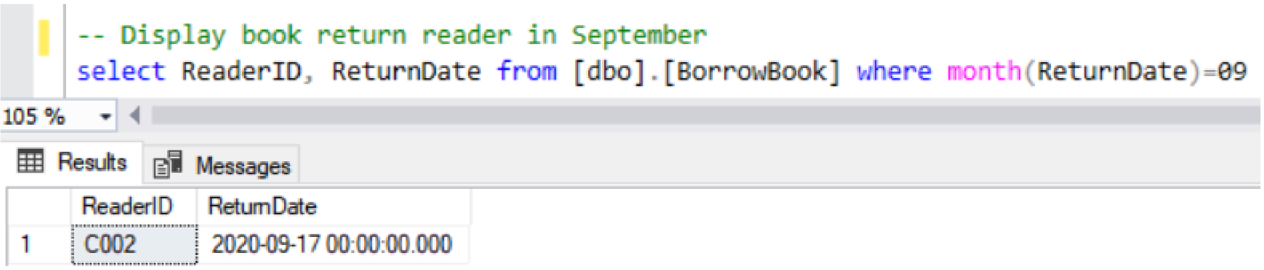
```
-- Display book borrow in November.
select BookID, BorrowDate from [dbo].[BorrowBook] where month(BorrowDate)=11
```

05 %

Results | Messages

	BookID	BorrowDate
1	A003	2020-11-01 00:00:00.000

- **Report book return reader in September.**



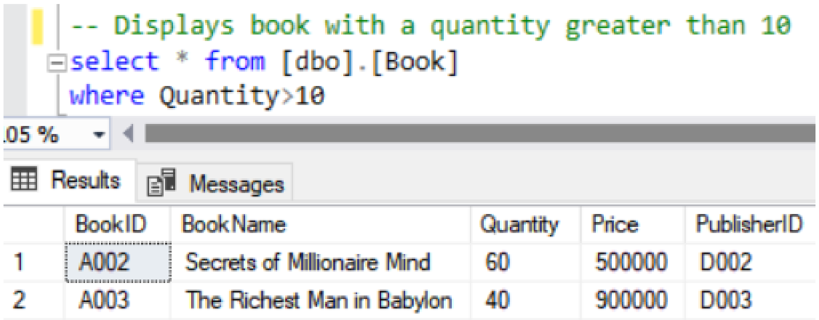
```
-- Display book return reader in September
select ReaderID, ReturnDate from [dbo].[BorrowBook] where month(ReturnDate)=09
```

105 %

Results | Messages

	ReaderID	ReturnDate
1	C002	2020-09-17 00:00:00.000

- **Report all books with a quantity greater than 10.**



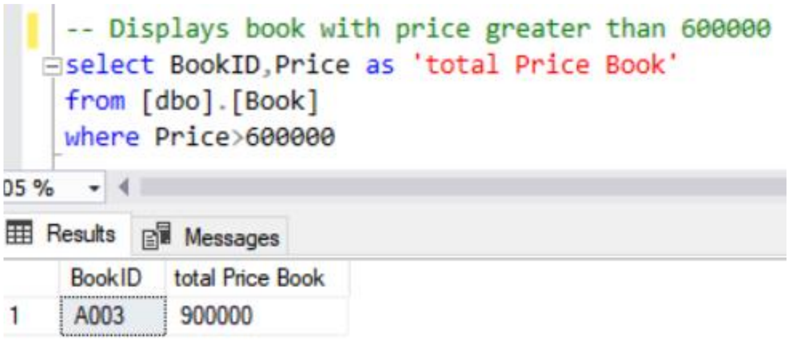
```
-- Displays book with a quantity greater than 10
select * from [dbo].[Book]
where Quantity>10
```

05 %

Results | Messages

	BookID	BookName	Quantity	Price	PublisherID
1	A002	Secrets of Millionaire Mind	60	500000	D002
2	A003	The Richest Man in Babylon	40	900000	D003

- **Report books with a price greater than 600000**



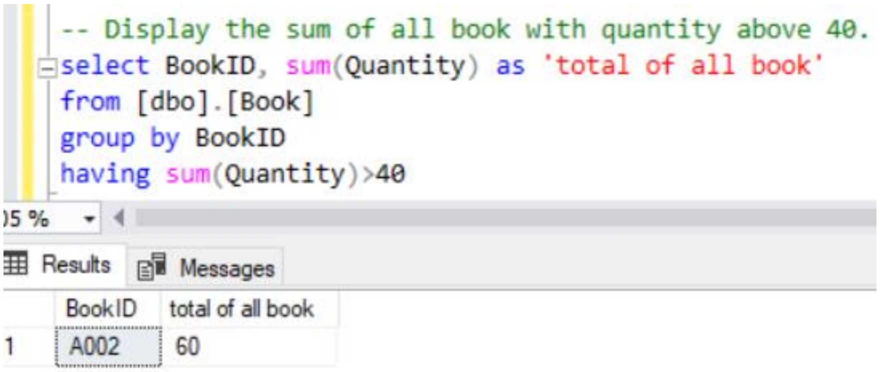
```
-- Displays book with price greater than 600000
select BookID,Price as 'total Price Book'
from [dbo].[Book]
where Price>600000
```

05 %

Results | Messages

	BookID	total Price Book
1	A003	900000

- **Report the sum of all book with quantity above 40.**



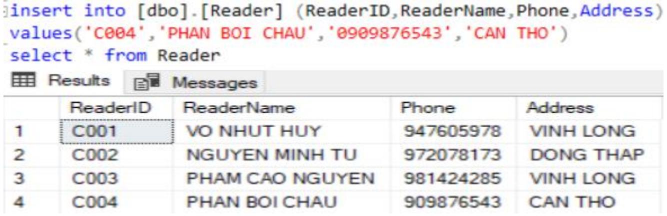
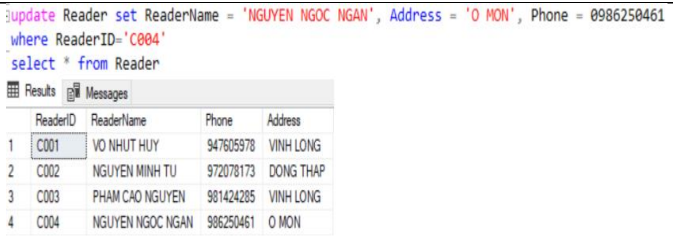
```
-- Display the sum of all book with quantity above 40.
select BookID, sum(Quantity) as 'total of all book'
from [dbo].[Book]
group by BookID
having sum(Quantity)>40
```

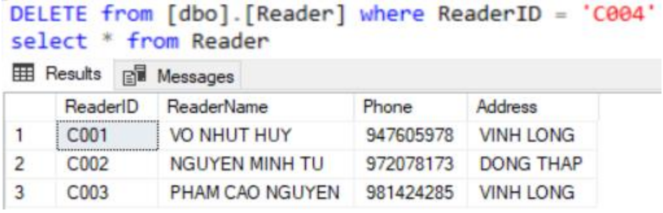
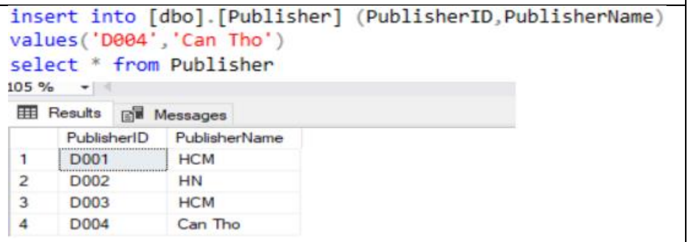
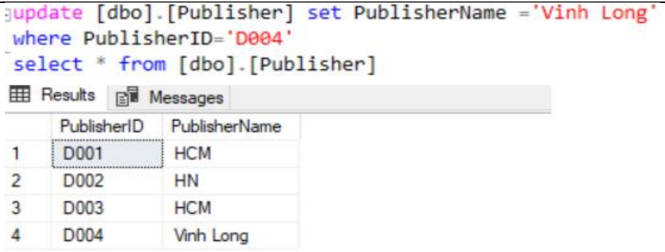
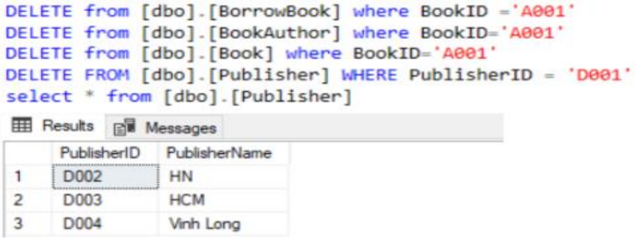
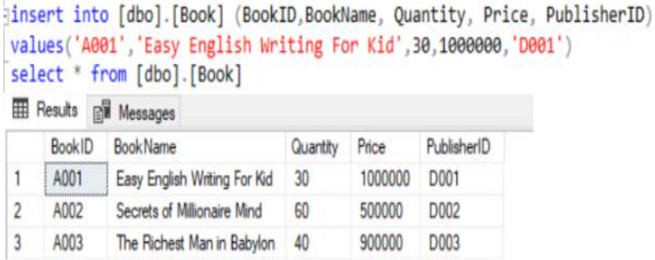
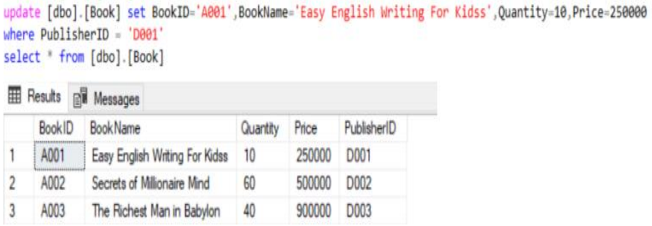
```
   BookID   total of all book
1  A002     60
```

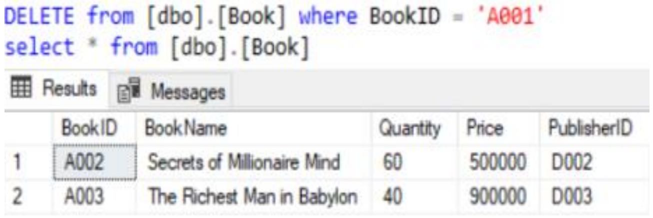
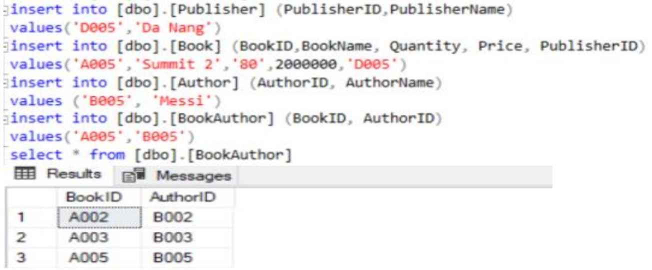
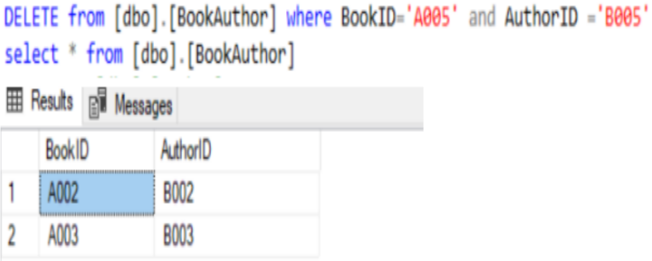
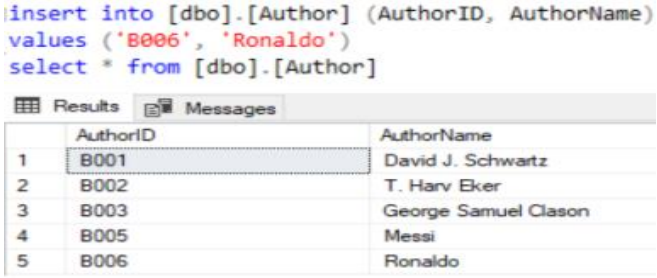
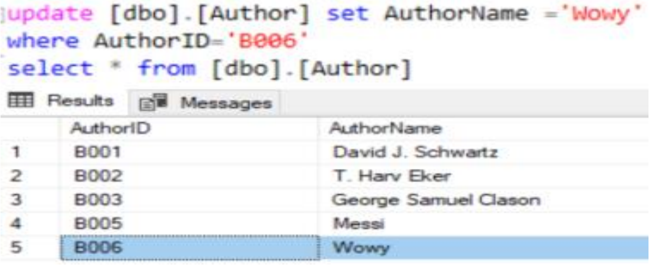
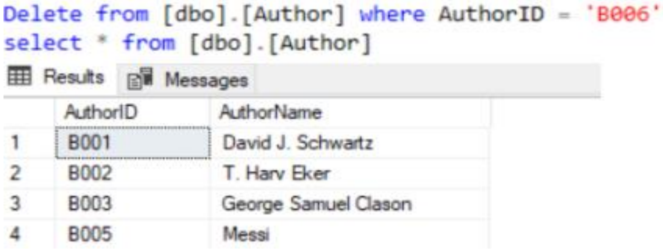
Benefits of statistical

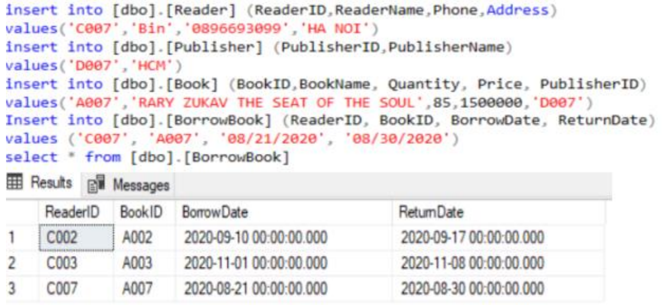
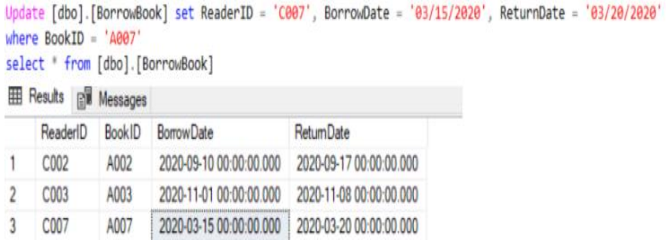
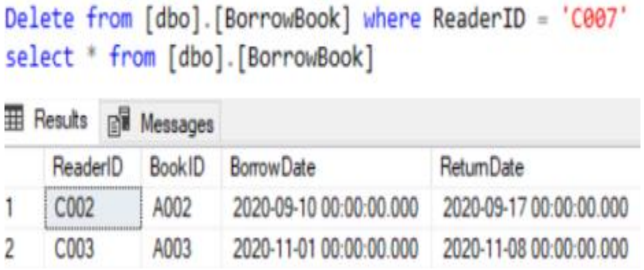
Statistical reports make it easier to manage for to borrow and return books, date of borrowing and return of books, number of books, and prices of books in the library. and can easily all display readers borrowing books in November, and easily display all books with prices greater than 600000. Display the sum of all books with a quantity above 40. Display all book return readers in September.

P4 Test the system against user and system requirements.

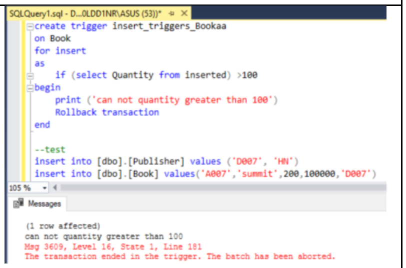
No.	Test case	Function	Execute
1.	Verify that reader's information will save into reader table when user enters valid information.	Adding Reader	 insert into [dbo].[Reader] (ReaderID,ReaderName,Phone,Address) values('C004','PHAN BOI CHAU','0909876543','CAN THO') select * from Reader Results Messages ReaderID ReaderName Phone Address 1 C001 VO NHUT HUY 947605978 VINH LONG 2 C002 NGUYEN MINH TU 972078173 DONG THAP 3 C003 PHAM CAO NGUYEN 981424285 VINH LONG 4 C004 PHAN BOI CHAU 909876543 CAN THO
	Verify that reader information will update into Reader table when user update valid information	Update Reader	 update Reader set ReaderName = 'NGUYEN NGOC NGAN', Address = 'O MON', Phone = 0986250461 where ReaderID='C004' select * from Reader Results Messages ReaderID ReaderName Phone Address 1 C001 VO NHUT HUY 947605978 VINH LONG 2 C002 NGUYEN MINH TU 972078173 DONG THAP 3 C003 PHAM CAO NGUYEN 981424285 VINH LONG 4 C004 NGUYEN NGOC NGAN 986250461 O MON

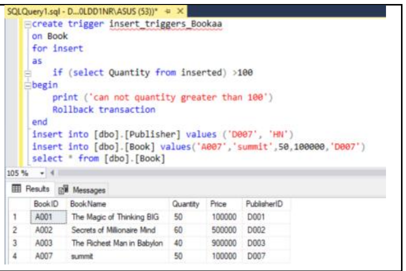
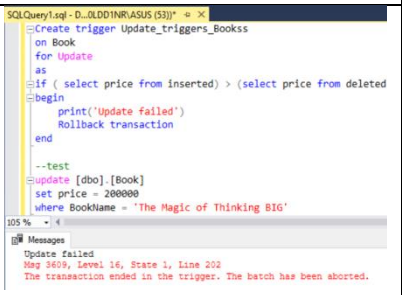
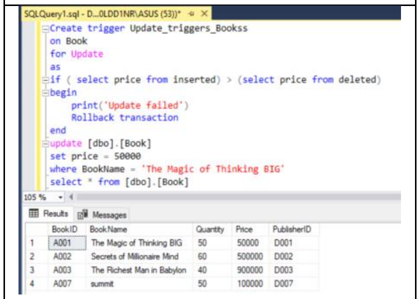
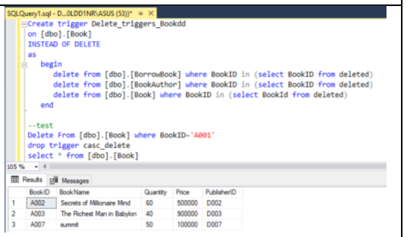
	Verify that reader information will be deleted when user enters a valid ReaderID code.	Delete Reader	 DELETE from [dbo].[Reader] where ReaderID = 'C004' select * from Reader Results Messages ReaderID ReaderName Phone Address 1 C001 VO NHUT HUY 947605978 VINH LONG 2 C002 NGUYEN MINH TU 972078173 DONG THAP 3 C003 PHAM CAO NGUYEN 981424285 VINH LONG
2.	Verify that publisher information will save into publisher table when user enters valid information.	Adding Publisher	 insert into [dbo].[Publisher] (PublisherID,PublisherName) values('D004','Can Tho') select * from Publisher 105 % Results Messages PublisherID PublisherName 1 D001 HCM 2 D002 HN 3 D003 HCM 4 D004 Can Tho
	Verify that publisher information will update into Publisher table when user update valid information.	Update Publisher	 update [dbo].[Publisher] set PublisherName ='Vinh Long' where PublisherID='D004' select * from [dbo].[Publisher] Results Messages PublisherID PublisherName 1 D001 HCM 2 D002 HN 3 D003 HCM 4 D004 Vinh Long
	Verify that publisher information will be deleted when user enters a valid PublisherID code.	Delete Publisher	 DELETE from [dbo].[BorrowBook] where BookID ='A001' DELETE from [dbo].[BookAuthor] where BookID='A001' DELETE from [dbo].[Book] where BookID='A001' DELETE FROM [dbo].[Publisher] WHERE PublisherID = 'D001' select * from [dbo].[Publisher] Results Messages PublisherID PublisherName 1 D002 HN 2 D003 HCM 3 D004 Vinh Long
3.	Verify that book information will save into book table when user enters valid information.	Adding Book	 insert into [dbo].[Book] (BookID,BookName, Quantity, Price, PublisherID) values('A001','Easy English Writing For Kid',30,1000000,'D001') select * from [dbo].[Book] Results Messages BookID BookName Quantity Price PublisherID 1 A001 Easy English Writing For Kid 30 1000000 D001 2 A002 Secrets of Millionaire Mind 60 500000 D002 3 A003 The Richest Man in Babylon 40 900000 D003
	Verify that Book information will update into Book table when user update valid information.	Update Book	 update [dbo].[Book] set BookID='A001',BookName='Easy English Writing For Kidss',Quantity=10,Price=250000 where PublisherID = 'D001' select * from [dbo].[Book] Results Messages BookID BookName Quantity Price PublisherID 1 A001 Easy English Writing For Kidss 10 250000 D001 2 A002 Secrets of Millionaire Mind 60 500000 D002 3 A003 The Richest Man in Babylon 40 900000 D003

	Verify that book information will be deleted when user enters a valid BookID code.	Delete Book	 DELETE from [dbo].[Book] where BookID = 'A001' select * from [dbo].[Book] Results Messages BookID BookName Quantity Price PublisherID 1 A002 Secrets of Millionaire Mind 60 500000 D002 2 A003 The Richest Man in Babylon 40 900000 D003
4.	Verify that book author information will save into BookAuthor table when user enters valid information.	Adding BookAuthor	 insert into [dbo].[Publisher] (PublisherID,PublisherName) values('D005','Da Nang') insert into [dbo].[Book] (BookID,BookName, Quantity, Price, PublisherID) values('A005','Summit 2','80',2000000,'D005') insert into [dbo].[Author] (AuthorID, AuthorName) values ('B005', 'Messi') insert into [dbo].[BookAuthor] (BookID, AuthorID) values('A005','B005') select * from [dbo].[BookAuthor] Results Messages BookID AuthorID 1 A002 B002 2 A003 B003 3 A005 B005
	Verify that book author information will be deleted when user enters a valid BookID and AuthorID code.	Delete BookAuthor	 DELETE from [dbo].[BookAuthor] where BookID='A005' and AuthorID ='B005' select * from [dbo].[BookAuthor] Results Messages BookID AuthorID 1 A002 B002 2 A003 B003
5.	Verify that author information will save into author table when user enters valid information.	Adding Author	 insert into [dbo].[Author] (AuthorID, AuthorName) values ('B006', 'Ronaldo') select * from [dbo].[Author] Results Messages AuthorID AuthorName 1 B001 David J. Schwartz 2 B002 T. Harv Eker 3 B003 George Samuel Clason 4 B005 Messi 5 B006 Ronaldo
	Verify that author information will update into Author table when user update valid information.	Update Author	 update [dbo].[Author] set AuthorName ='Wowy' where AuthorID='B006' select * from [dbo].[Author] Results Messages AuthorID AuthorName 1 B001 David J. Schwartz 2 B002 T. Harv Eker 3 B003 George Samuel Clason 4 B005 Messi 5 B006 Wowy
	Verify that author information will be deleted when user enters a valid AuthorID code.	Delete Author	 Delete from [dbo].[Author] where AuthorID = 'B006' select * from [dbo].[Author] Results Messages AuthorID AuthorName 1 B001 David J. Schwartz 2 B002 T. Harv Eker 3 B003 George Samuel Clason 4 B005 Messi

6.	Verify that borrow book information will save into borrowbook table when user enters valid information.	Adding BorrowBook	 insert into [dbo].[Reader] (ReaderID,ReaderName,Phone,Address) values('C007','Bin','0896693099','HA NOI') insert into [dbo].[Publisher] (PublisherID,PublisherName) values('D007','HCM') insert into [dbo].[Book] (BookID,BookName, Quantity, Price, PublisherID) values('A007','RARY ZUKAV THE SEAT OF THE SOUL',85,1500000,'D007') Insert into [dbo].[BorrowBook] (ReaderID, BookID, BorrowDate, ReturnDate) values ('C007', 'A007', '08/21/2020', '08/30/2020') select * from [dbo].[BorrowBook] Results Messages ReaderID BookID BorrowDate ReturnDate 1 C002 A002 2020-09-10 00:00:00.000 2020-09-17 00:00:00.000 2 C003 A003 2020-11-01 00:00:00.000 2020-11-08 00:00:00.000 3 C007 A007 2020-08-21 00:00:00.000 2020-08-30 00:00:00.000
	Verify that borrow book information will update into Borrowbook table when user update valid information.	Update BorrowBook	 Update [dbo].[BorrowBook] set ReaderID = 'C007', BorrowDate = '03/15/2020', ReturnDate = '03/20/2020' where BookID = 'A007' select * from [dbo].[BorrowBook] Results Messages ReaderID BookID BorrowDate ReturnDate 1 C002 A002 2020-09-10 00:00:00.000 2020-09-17 00:00:00.000 2 C003 A003 2020-11-01 00:00:00.000 2020-11-08 00:00:00.000 3 C007 A007 2020-03-15 00:00:00.000 2020-03-20 00:00:00.000
	Verify that borrow book information will be deleted when user enters a valid ReaderID code.	Delete BorrowBook	 Delete from [dbo].[BorrowBook] where ReaderID = 'C007' select * from [dbo].[BorrowBook] Results Messages ReaderID BookID BorrowDate ReturnDate 1 C002 A002 2020-09-10 00:00:00.000 2020-09-17 00:00:00.000 2 C003 A003 2020-11-01 00:00:00.000 2020-11-08 00:00:00.000

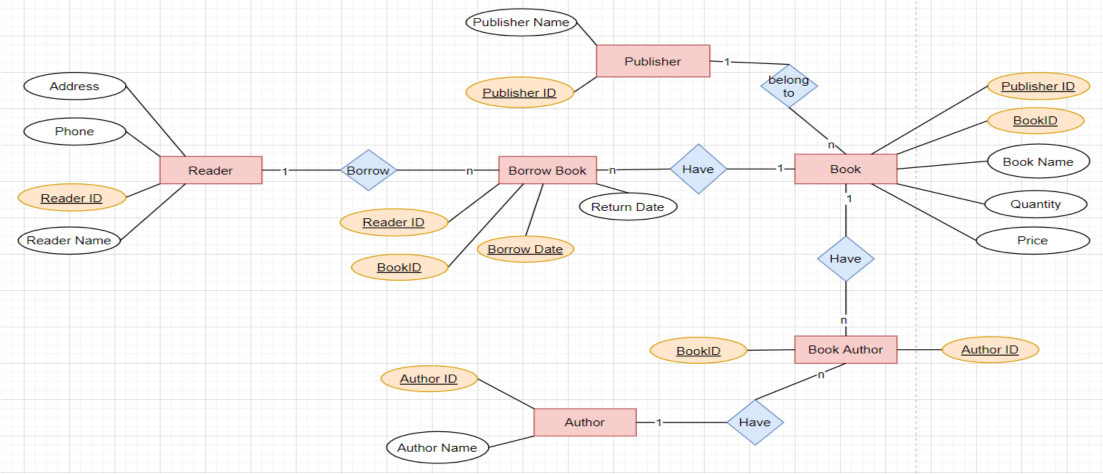
M2 Implement a fully functional database system which includes system security and database maintenance.

No.	Test case	Function	Execute
1.	Verify that error message is displayed and the transaction will be canceled if a user enters the number of books greater than 100.	Adding Book.	 SQLQuery1.sql - D...0LDD1NR\ASUS (53))* create trigger insert_triggers_Bookaa on Book for insert as if (select Quantity from inserted) >100 begin print ('can not quantity greater than 100') Rollback transaction end --test insert into [dbo].[Publisher] values ('D007', 'HN') insert into [dbo].[Book] values('A007','summit',200,100000,'D007') 105 % Messages (1 row affected) can not quantity greater than 100 Msg 3609, Level 16, State 1, Line 181 The transaction ended in the trigger. The batch has been aborted.

	Verify that successfully message is displayed and the transaction valid if a user enters the number of books greater less 100.		SQLQuery1.sql - D...0LDD1NR\ASUS (53))* create trigger insert_triggers_Bookaa on Book for insert as if (select Quantity from inserted) >100 begin print ('can not quantity greater than 100') Rollback transaction end insert into [dbo].[Publisher] values ('D007', 'HN') insert into [dbo].[Book] values('A007','summit',50,100000,'D007') select * from [dbo].[Book] 105 % Results Messages BookID BookName Quantity Price PublisherID 1 A001 The Magic of Thinking BIG 50 100000 D001 2 A002 Secrets of Millionaire Mind 60 500000 D002 3 A003 The Richest Man in Babylon 40 900000 D003 4 A007 summit 50 100000 D007
2.	Verify that update operation will be performed canceled when user inputs the updated book price is greater than the current book price.	Update Price Book	SQLQuery1.sql - D...0LDD1NR\ASUS (53))* Create trigger Update_triggers_Bookss on Book for Update as if (select price from inserted) > (select price from deleted begin print('Update failed') Rollback transaction end --test update [dbo].[Book] set price = 200000 where BookName = 'The Magic of Thinking BIG' 105 % Messages Update failed Msg 3609, Level 16, State 1, Line 202 The transaction ended in the trigger. The batch has been aborted.
	Verify that the book information is saved to the Book table when the user updates the valid information.(The updated book price is smaller than the current one).		SQLQuery1.sql - D...0LDD1NR\ASUS (53))* Create trigger Update_triggers_Bookss on Book for Update as if (select price from inserted) > (select price from deleted) begin print('Update failed') Rollback transaction end update [dbo].[Book] set price = 50000 where BookName = 'The Magic of Thinking BIG' select * from [dbo].[Book] 105 % Results Messages BookID BookName Quantity Price PublisherID 1 A001 The Magic of Thinking BIG 50 50000 D001 2 A002 Secrets of Millionaire Mind 60 500000 D002 3 A003 The Richest Man in Babylon 40 900000 D003 4 A007 summit 50 100000 D007
3.	Verify that book information will be deleted when user enters a valid BookID code.	Delete Book	SQLQuery1.sql - D...0LDD1NR\ASUS (53))* Create trigger Delete_triggers_Bookdd on [dbo].[Book] INSTEAD OF DELETE as begin delete from [dbo].[BorrowBook] where BookID in (select BookID from deleted) delete from [dbo].[BookAuthor] where BookID in (select BookID from deleted) delete from [dbo].[Book] where BookID in (select BookId from deleted) end --test Delete From [dbo].[Book] where BookID='A001' drop trigger casc_delete select * from [dbo].[Book] 105 % Results Messages BookID BookName Quantity Price PublisherID 1 A002 Secrets of Millionaire Mind 60 500000 D002 2 A003 The Richest Man in Babylon 40 900000 D003 3 A007 summit 50 100000 D007

P5 Produce technical and user documentation for a fully functional system.

- ERD

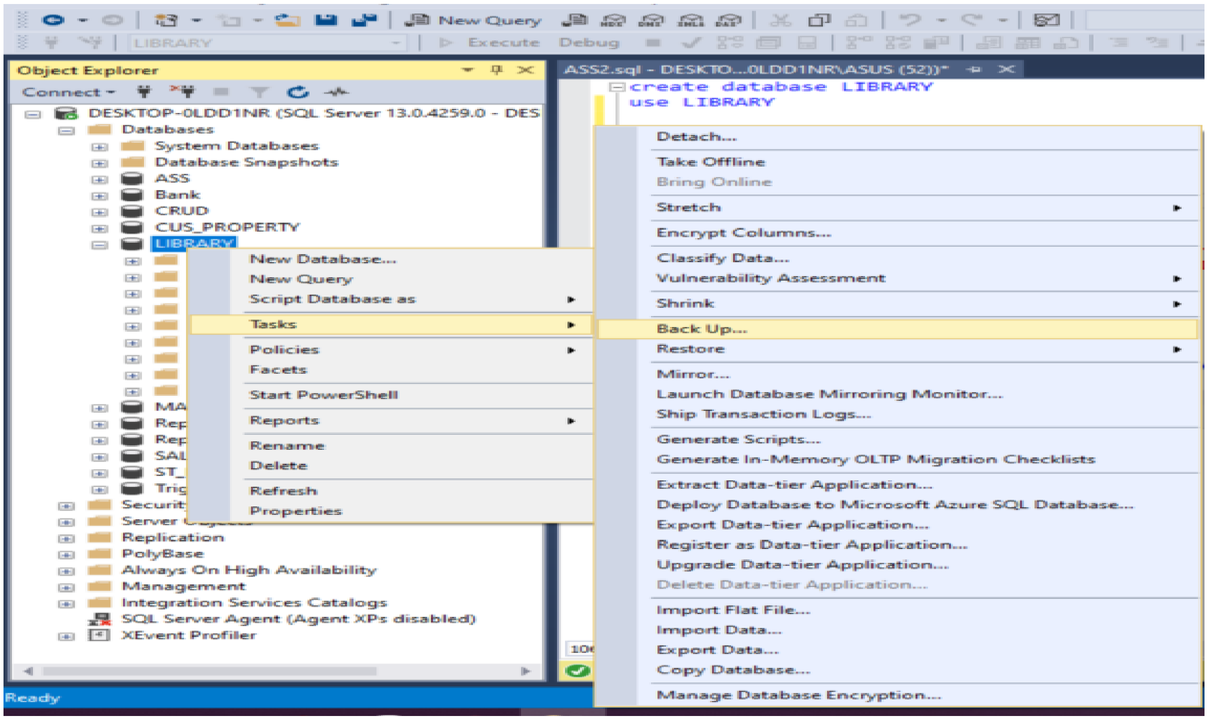


- Current version number : 1.0

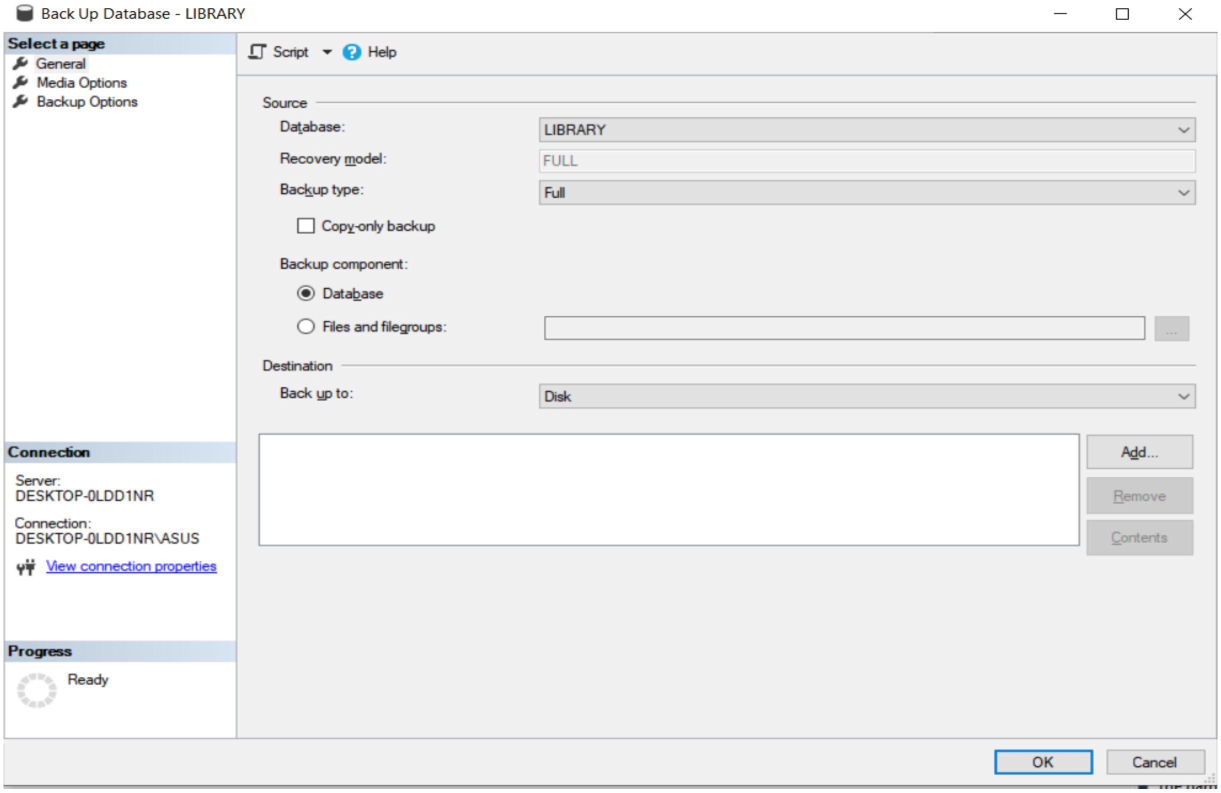
- Date last changed and implemented: 11/02/2020

- Back up Database :

+ step1: choose 'Tasks' => choose 'Back Up…


```
create database LIBRARY
use LIBRARY
```

+Step2 : Choose ‘Remove’ then select choose ‘add’



Back Up Database - LIBRARY

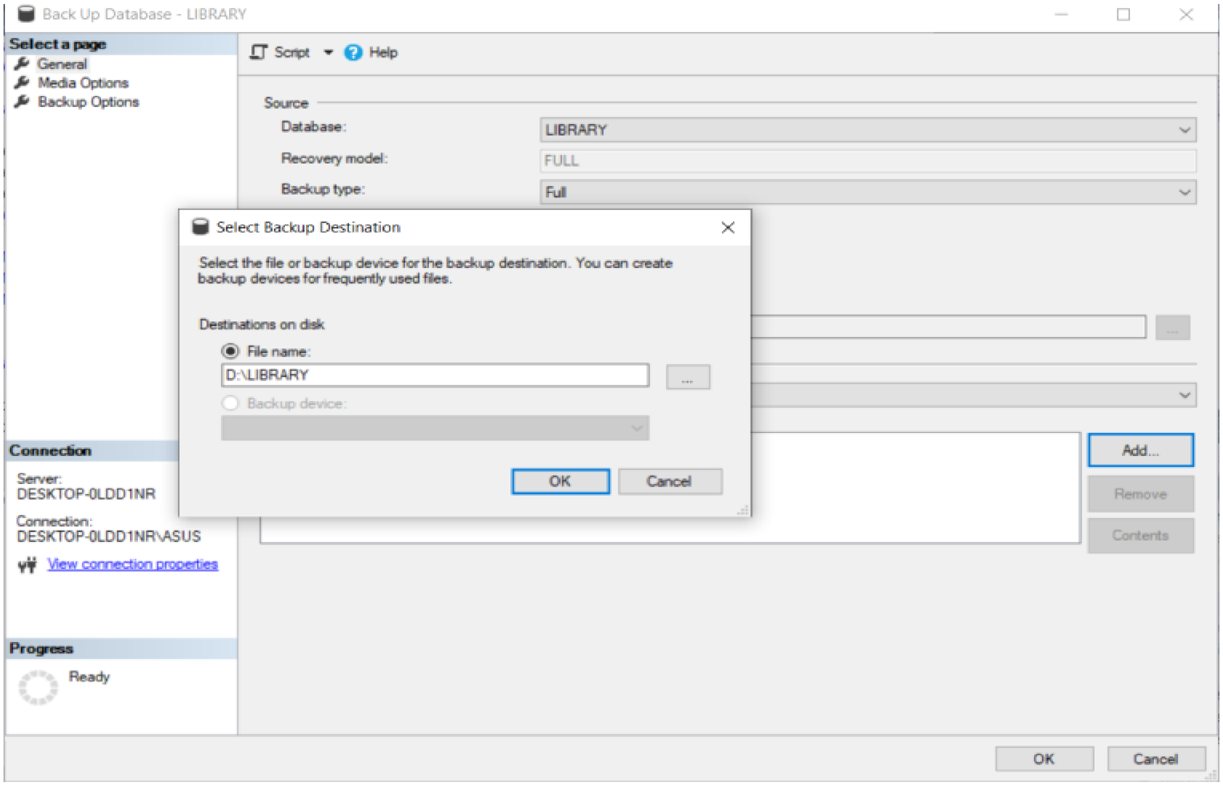
Source
Database: LIBRARY
Recovery model: FULL
Backup type: Full
□ Copy-only backup
Backup component:
● Database
○ Files and filegroups:

Destination
Back up to: Disk

Connection
Server:
DESKTOP-0LDD1NR
Connection:
DESKTOP-0LDD1NR\ASUS
View connection properties

Progress
Ready

+ Step 3: Add Save to data D:\LIBRARY



Back Up Database - LIBRARY

Source
Database: LIBRARY
Recovery model: FULL
Backup type: Full

Select Backup Destination

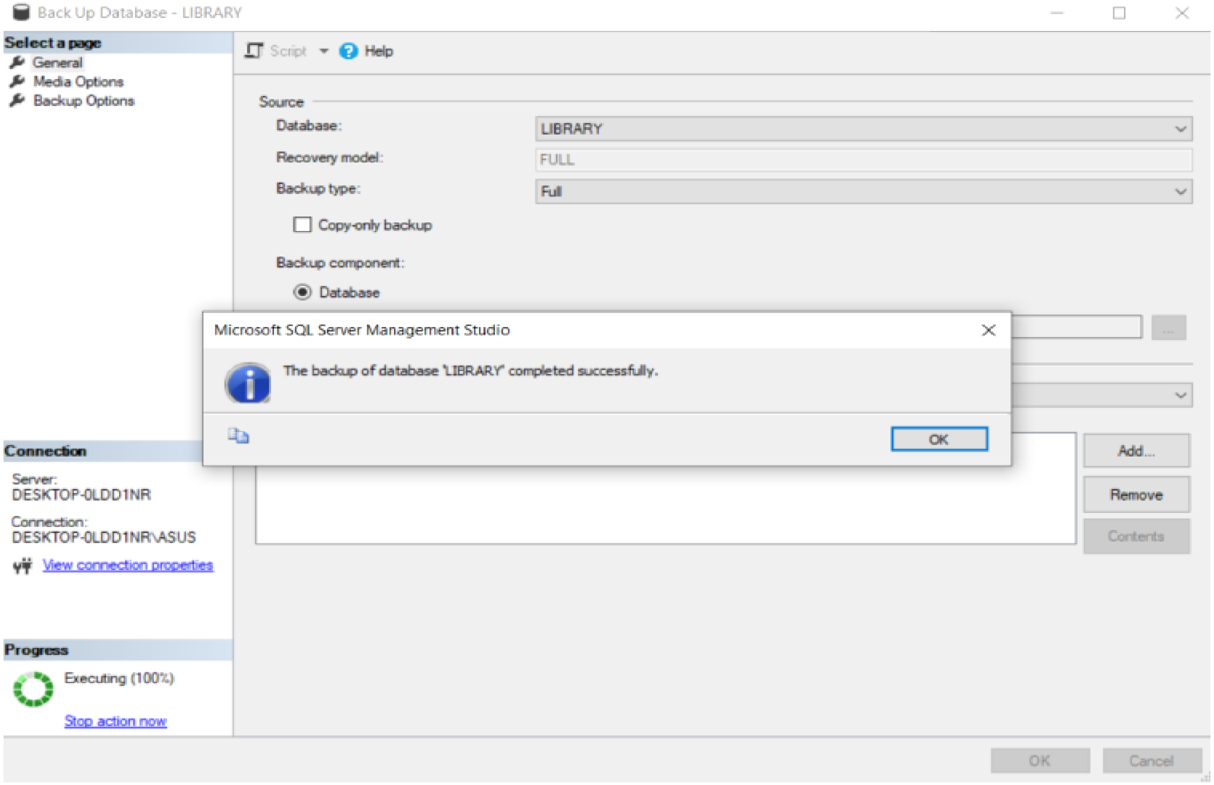
Select the file or backup device for the backup destination. You can create backup devices for frequently used files.

Destinations on disk
● File name:
D:\LIBRARY
○ Backup device:

Connection
Server:
DESKTOP-0LDD1NR
Connection:
DESKTOP-0LDD1NR\ASUS
View connection properties

Progress
Ready

+ Step 4: choose ok



Back Up Database - LIBRARY

Source

Database: LIBRARY

Recovery model: FULL

Backup type: Full

□ Copy-only backup

Backup component:

● Database

Microsoft SQL Server Management Studio

The backup of database 'LIBRARY' completed successfully.

OK

Server:
DESKTOP-0LDD1NR

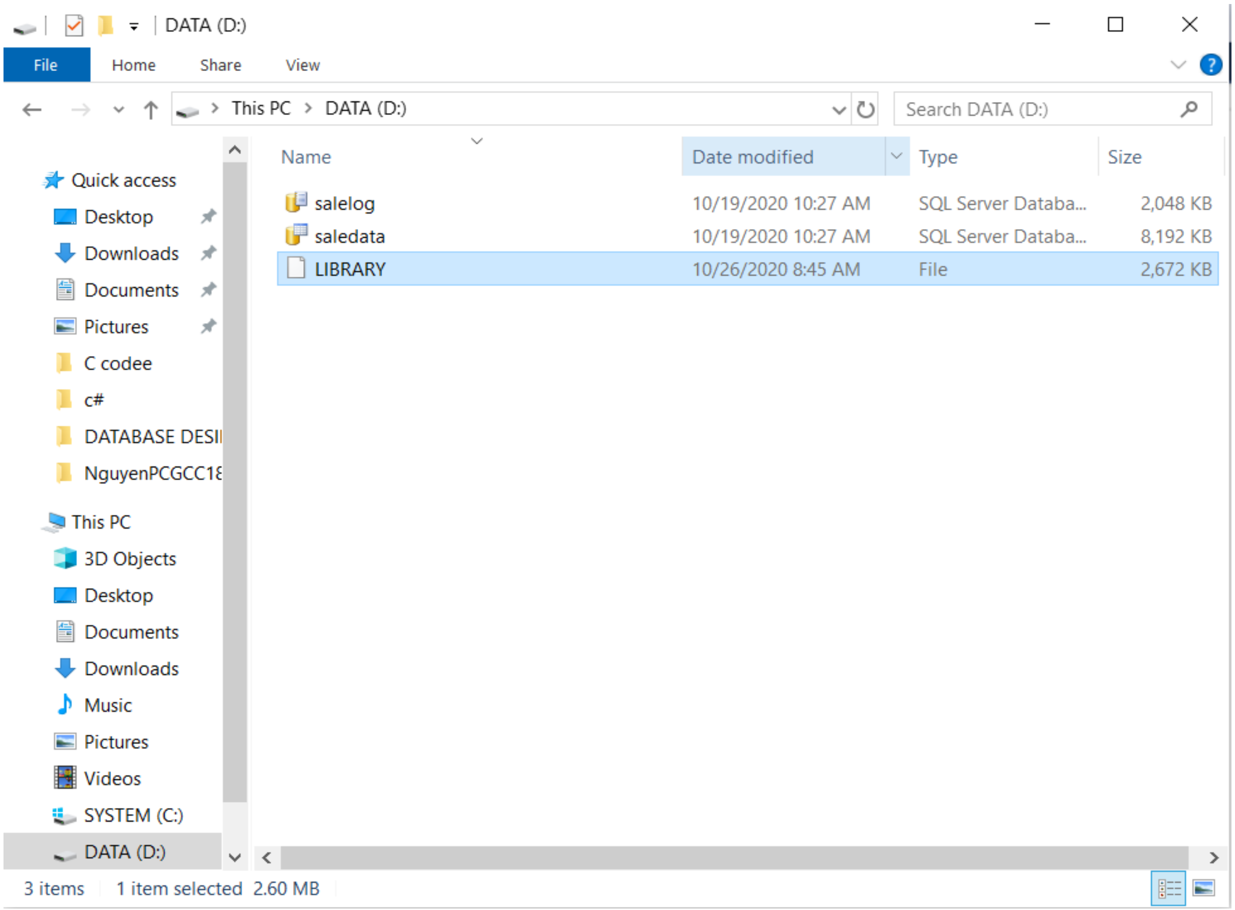
Connection:
DESKTOP-0LDD1NR\ASUS

View connection properties

Executing (100%)

Stop action now

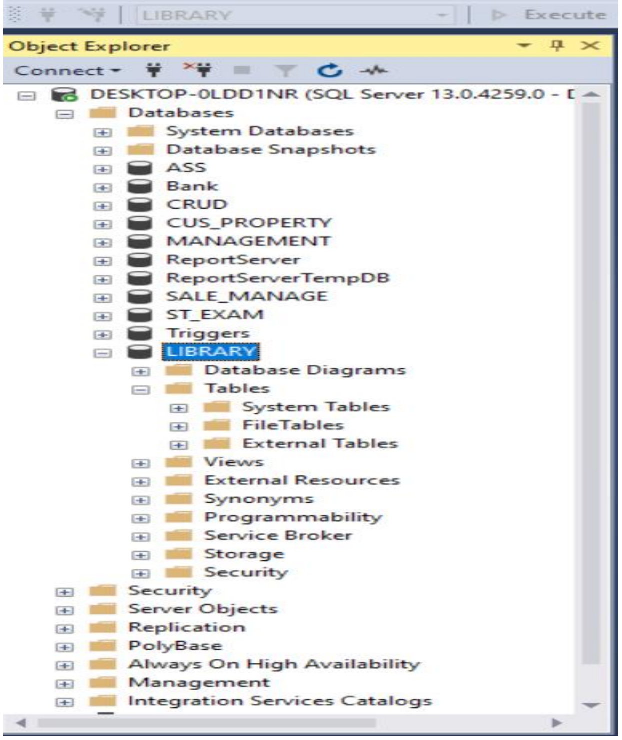
+ Step 5: Check in drive data(D:) have LIBRARY



Name	Date modified	Type	Size
salelog	10/19/2020 10:27 AM	SQL Server Databa...	2,048 KB
saledata	10/19/2020 10:27 AM	SQL Server Databa...	8,192 KB
LIBRARY	10/26/2020 8:45 AM	File	2,672 KB

3 items 1 item selected 2.60 MB

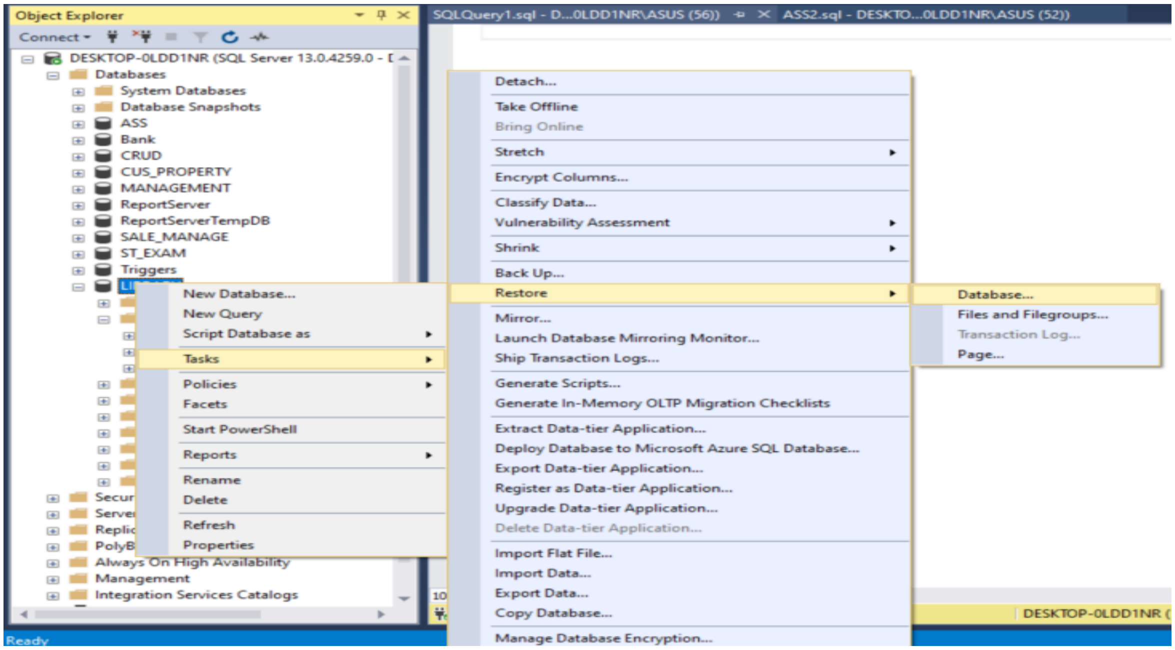
+ Step 6: Delete databases LIBRARY



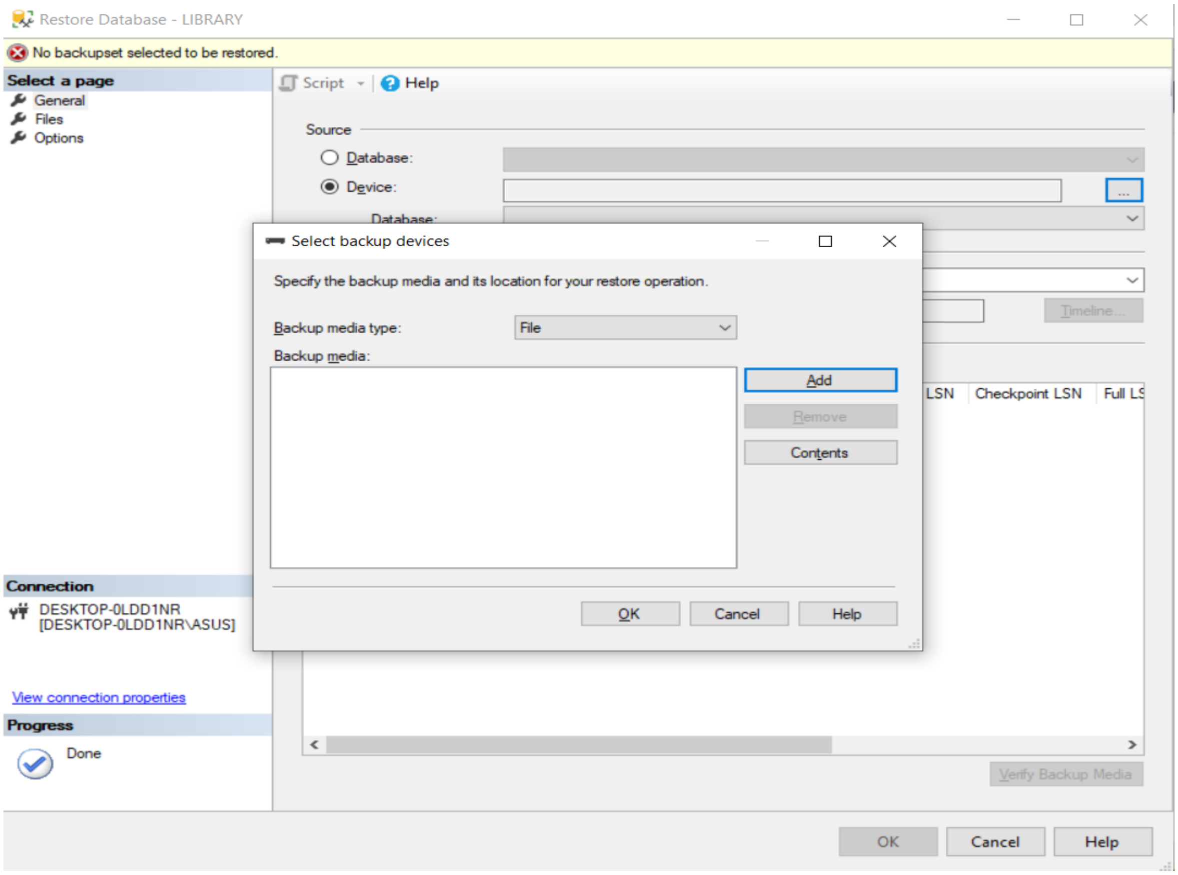
```
DESKTOP-0LDD1NR (SQL Server 13.0.4259.0 - [
  Databases
    System Databases
    Database Snapshots
    ASS
    Bank
    CRUD
    CUS_PROPERTY
    MANAGEMENT
    ReportServer
    ReportServerTempDB
    SALE_MANAGE
    ST_EXAM
    Triggers
    LIBRARY
      Database Diagrams
      Tables
        System Tables
        FileTables
        External Tables
      Views
      External Resources
      Synonyms
      Programmability
      Service Broker
      Storage
      Security
  Security
  Server Objects
  Replication
  PolyBase
  Always On High Availability
  Management
  Integration Services Catalogs
```

+ Step 7: Create databases LIBRARY

+ Step 8: Choose ‘Tasks’ => ‘Restore’ => ‘Database…’



+ Step 9: Choose ‘Decive’ => ‘Add’


Restore Database - LIBRARY

No backupset selected to be restored.

Source
○ Database:
● Device:

Select backup devices

Specify the backup media and its location for your restore operation.

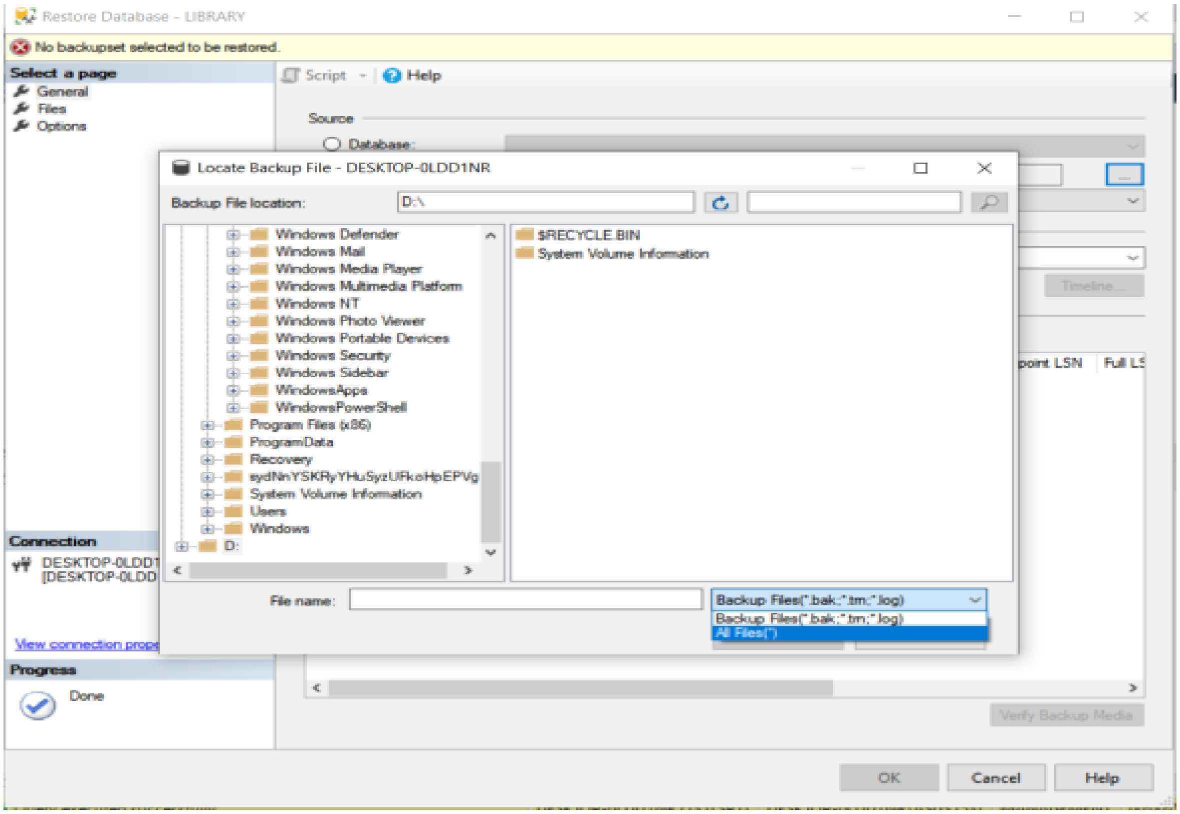
Backup media type: File

Backup media:

Connection
DESKTOP-0LDD1NR
[DESKTOP-0LDD1NR\ASUS]

Progress
Done

+ Step 10: Choose drive data(D:) => All files(*)



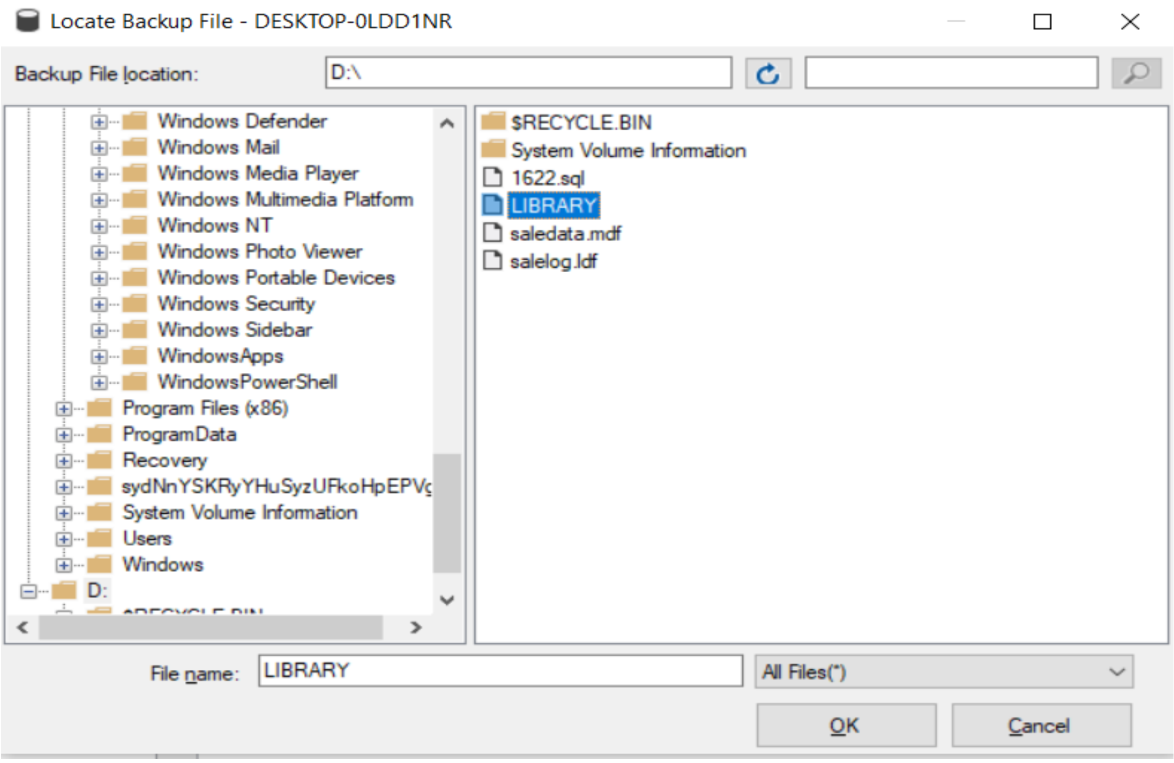
No backupset selected to be restored.

Backup File location: D:\

File name:

Backup Files(*.bak;*.tm;*.log)
Backup Files(*.bak;*.tm;*.log)
All Files(*)

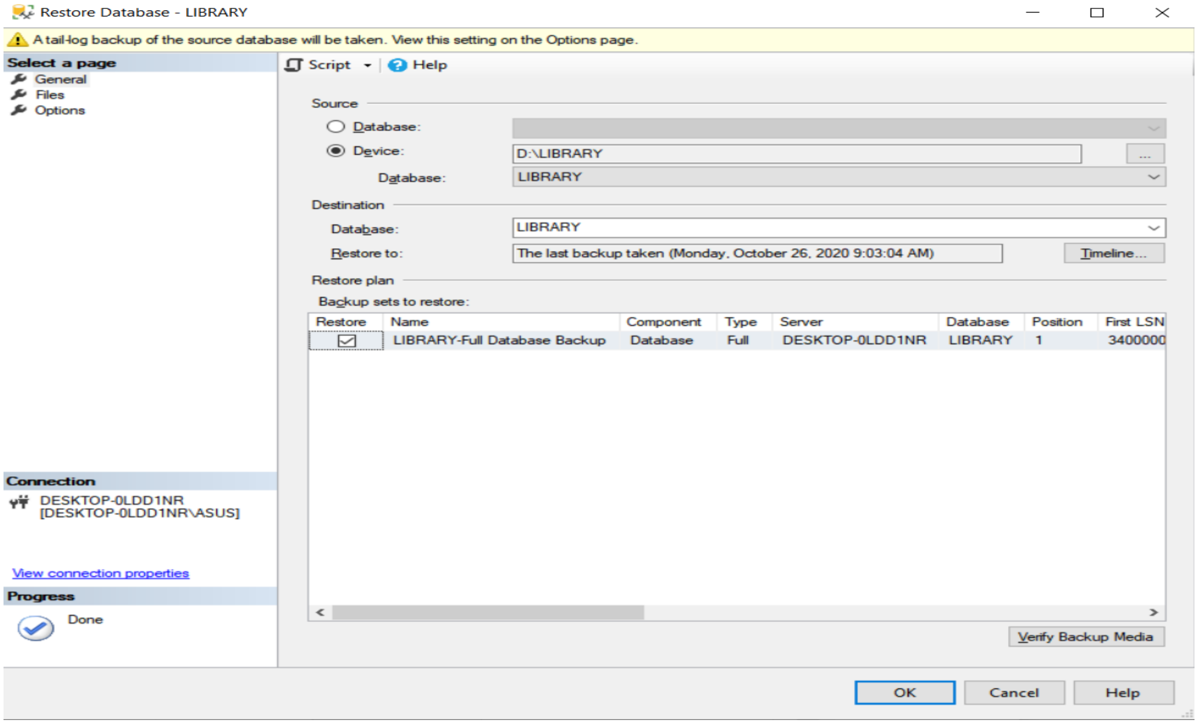
+ Step 11: Choose LIBRARY => OK



Backup File location: D:\

File name: LIBRARY

All Files(*)



Restore Database - LIBRARY

⚠ A tail-log backup of the source database will be taken. View this setting on the Options page.

Select a page: General, Files, Options

Source
○ Database:
● Device: D:\LIBRARY
Database: LIBRARY

Destination
Database: LIBRARY
Restore to: The last backup taken (Monday, October 26, 2020 9:03:04 AM)

Restore plan
Backup sets to restore:

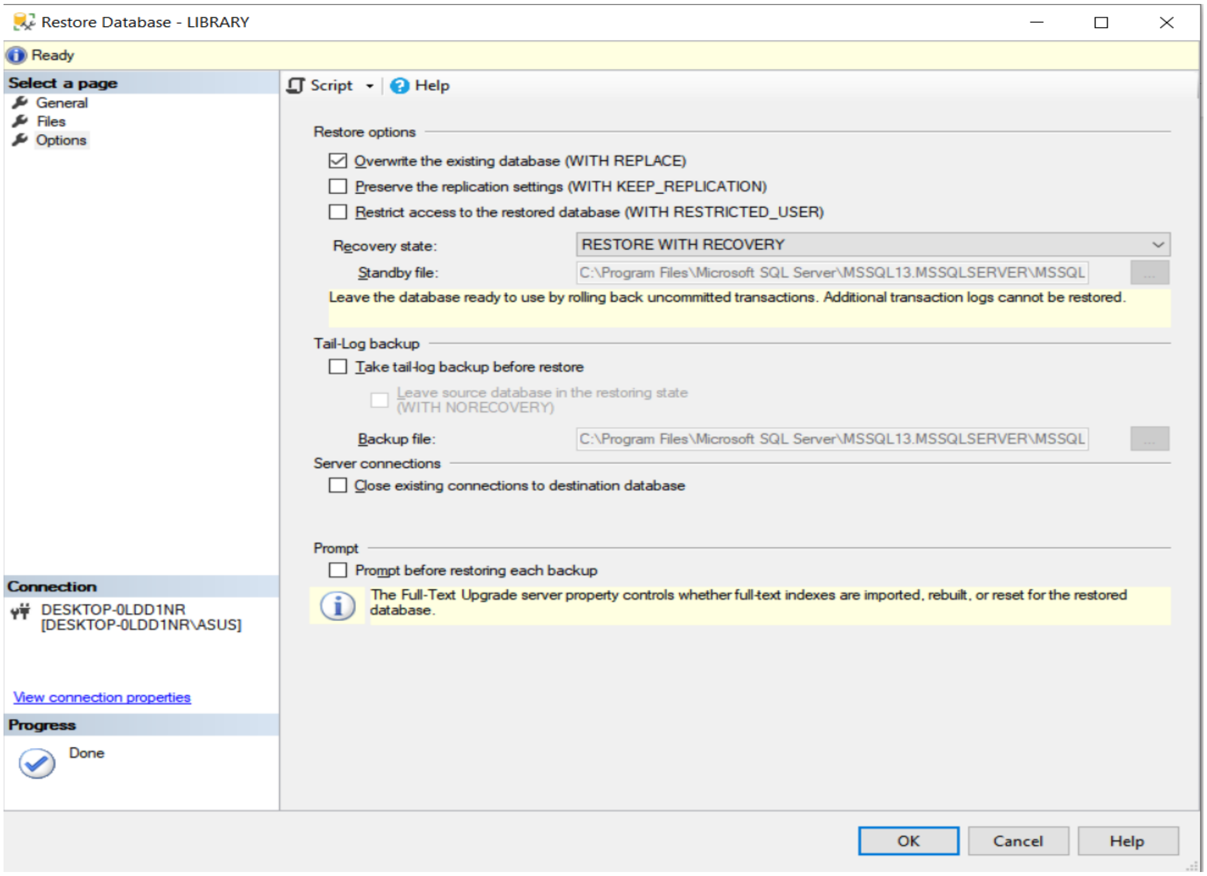
Restore	Name	Component	Type	Server	Database	Position	First LSN
☑	LIBRARY-Full Database Backup	Database	Full	DESKTOP-0LDD1NR	LIBRARY	1	3400000

Connection
DESKTOP-0LDD1NR
[DESKTOP-0LDD1NR\ASUS]

View connection properties

Progress
Done

+ Step 12: Choose 'Options'=> Tick 'Overwrite the existing database (WITH REPLACE)' and Uncheck 'Take tall-log backup before restore'



Restore Database - LIBRARY

Ready

Select a page: General, Files, Options

Restore options
☑ Overwrite the existing database (WITH REPLACE)
☐ Preserve the replication settings (WITH KEEP_REPLICATION)
☐ Restrict access to the restored database (WITH RESTRICTED_USER)

Recovery state: RESTORE WITH RECOVERY
Standby file: C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL
Leave the database ready to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored.

Tail-Log backup
☐ Take tail-log backup before restore
☐ Leave source database in the restoring state (WITH NORECOVERY)
Backup file: C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL

Server connections
☐ Close existing connections to destination database

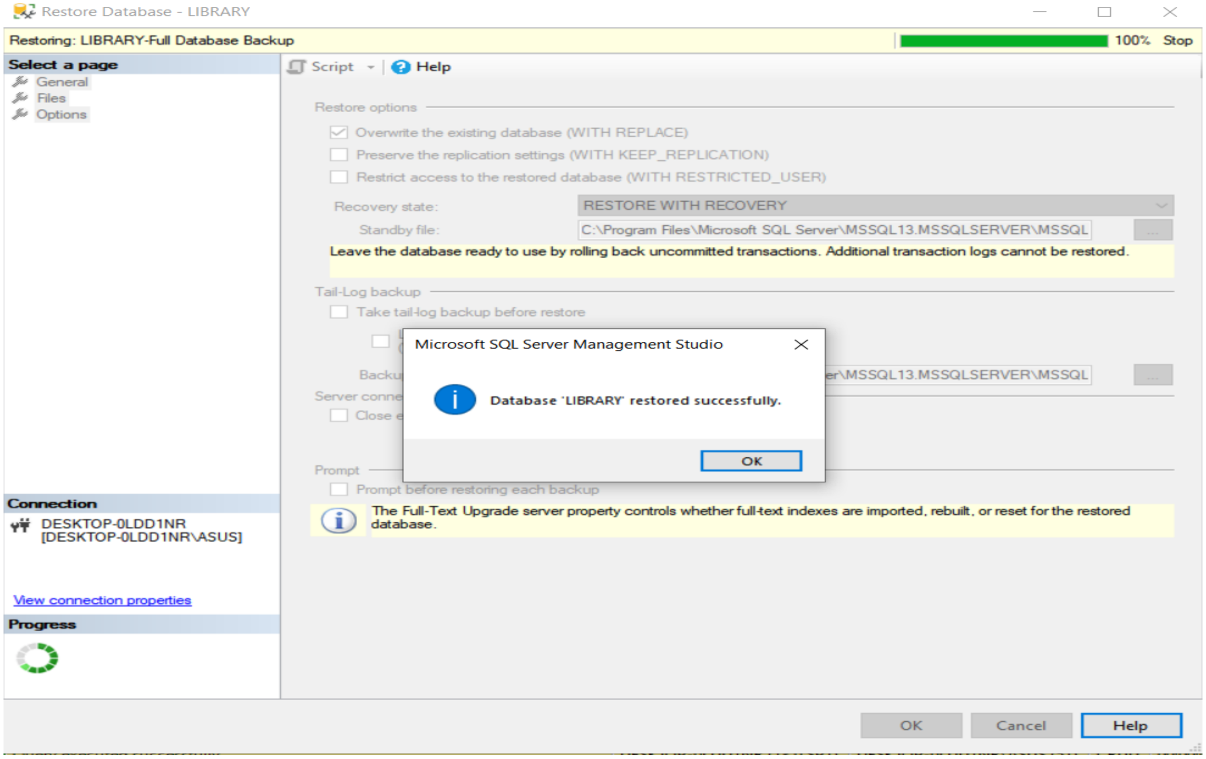
Prompt
☐ Prompt before restoring each backup
The Full-Text Upgrade server property controls whether full-text indexes are imported, rebuilt, or reset for the restored database.

Connection
DESKTOP-0LDD1NR
[DESKTOP-0LDD1NR\ASUS]

View connection properties

Progress
Done

+ Step 13: Choose ‘OK’



Restoring: LIBRARY-Full Database Backup 100%

Restore options

☑ Overwrite the existing database (WITH REPLACE)

□ Preserve the replication settings (WITH KEEP_REPLICATION)

□ Restrict access to the restored database (WITH RESTRICTED_USER)

Recovery state: RESTORE WITH RECOVERY

Standby file: C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL

Leave the database ready to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored.

Tail-Log backup

□ Take tail-log backup before restore

Backu ... er\MSSQL13.MSSQLSERVER\MSSQL

Server conne

□ Close e

Prompt

□ Prompt before restoring each backup

The Full-Text Upgrade server property controls whether full-text indexes are imported, rebuilt, or reset for the restored database.

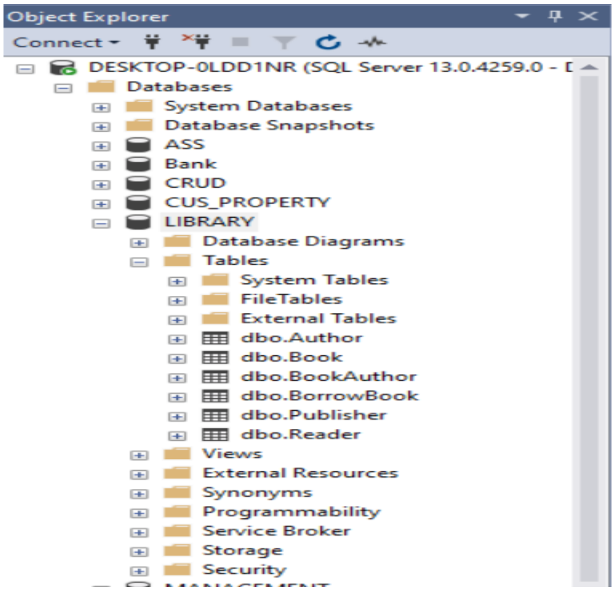
DESKTOP-0LDD1NR
[DESKTOP-0LDD1NR\ASUS]

Microsoft SQL Server Management Studio

Database 'LIBRARY' restored successfully.

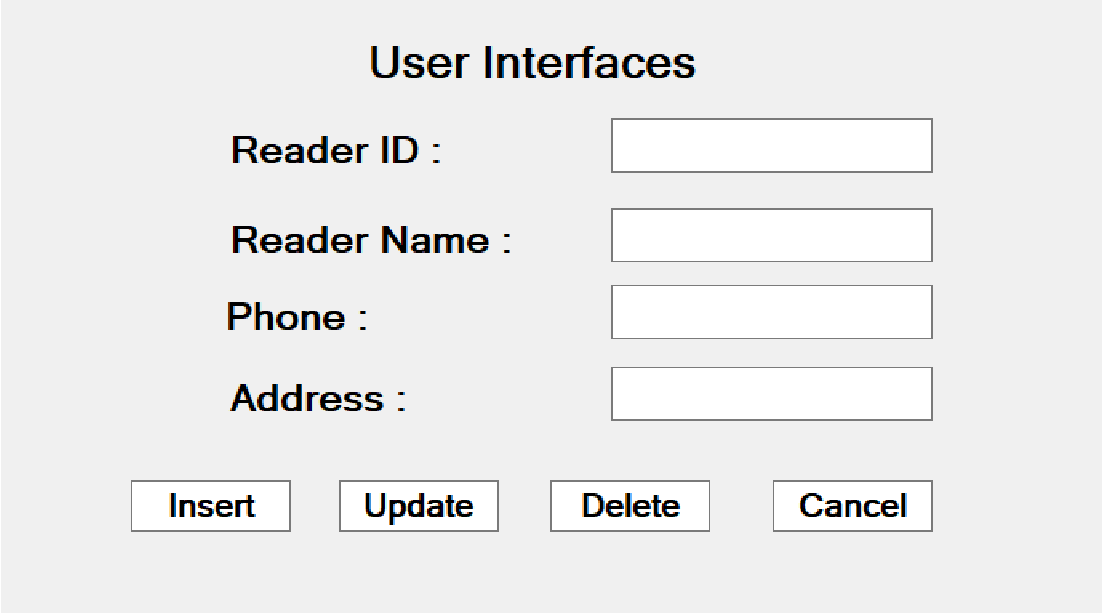
OK

+ Successfully displayed Database LIBRARY



```
⊟ DESKTOP-0LDD1NR (SQL Server 13.0.4259.0 - [
  ⊟ Databases
    ⊞ System Databases
    ⊞ Database Snapshots
    ⊞ ASS
    ⊞ Bank
    ⊞ CRUD
    ⊞ CUS_PROPERTY
    ⊟ LIBRARY
      ⊞ Database Diagrams
      ⊟ Tables
        ⊞ System Tables
        ⊞ FileTables
        ⊞ External Tables
        ⊞ dbo.Author
        ⊞ dbo.Book
        ⊞ dbo.BookAuthor
        ⊞ dbo.BorrowBook
        ⊞ dbo.Publisher
        ⊞ dbo.Reader
      ⊞ Views
      ⊞ External Resources
      ⊞ Synonyms
      ⊞ Programmability
      ⊞ Service Broker
      ⊞ Storage
      ⊞ Security
```

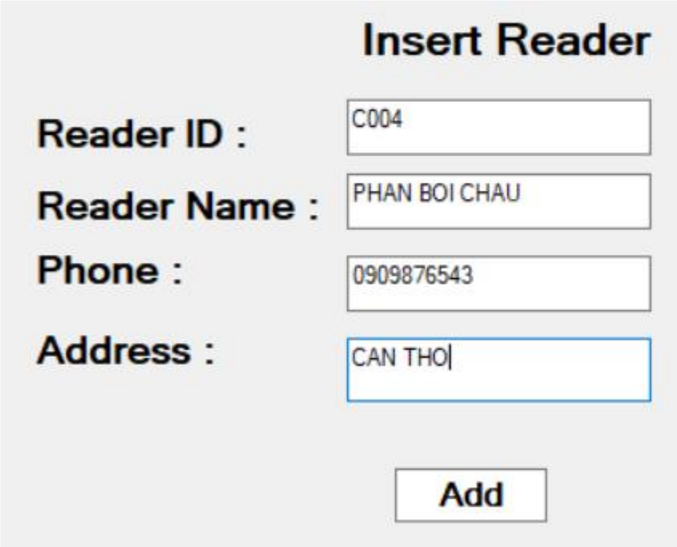
- The user interfaces.



Interfaces add Reader

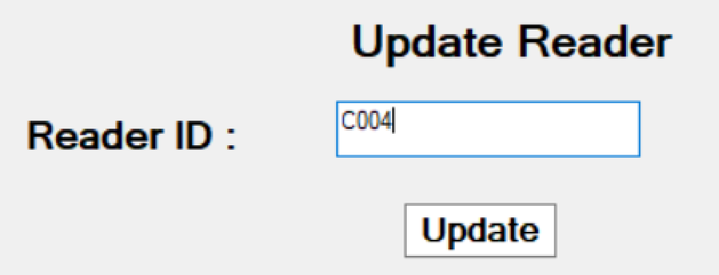
- ReaderID char(10)
- ReaderName varchar(50)
- Phone int
- Address char(10)

+ User enter ReaderID no more than 10 characters, Reader name is limited by 50 characters, User phone number must not enter letters or special characters. Address allows user to enter no more than 10 characters.



Reader ID : C004
Reader Name : PHAN BOI CHAU
Phone : 0909876543
Address : CAN THO

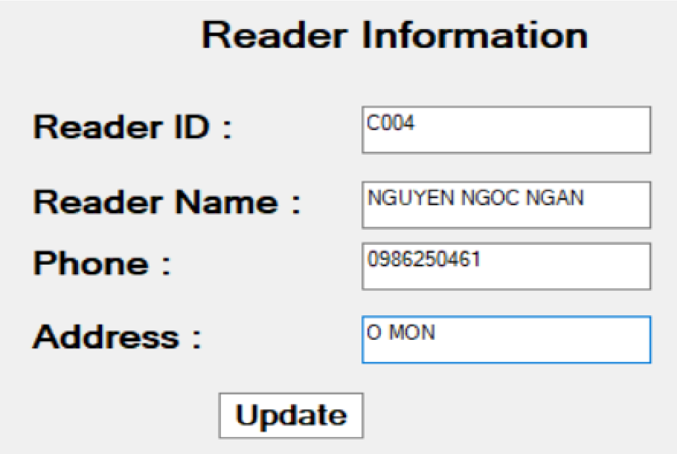
+ Update Reader



Update Reader

Reader ID : C004

+ User enters Reader ID to update



Reader Information

Reader ID : C004

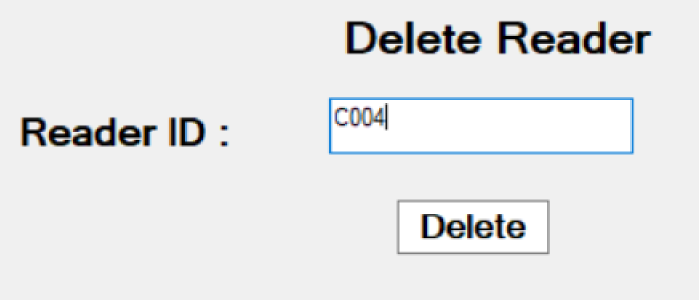
Reader Name : NGUYEN NGOC NGAN

Phone : 0986250461

Address : O MON

+ The user enters Reader information need update into the textbox and clicks 'update'. Update information requires the user to enter valid information, reader ID and without special characters.

+ Delete Reader



Delete Reader

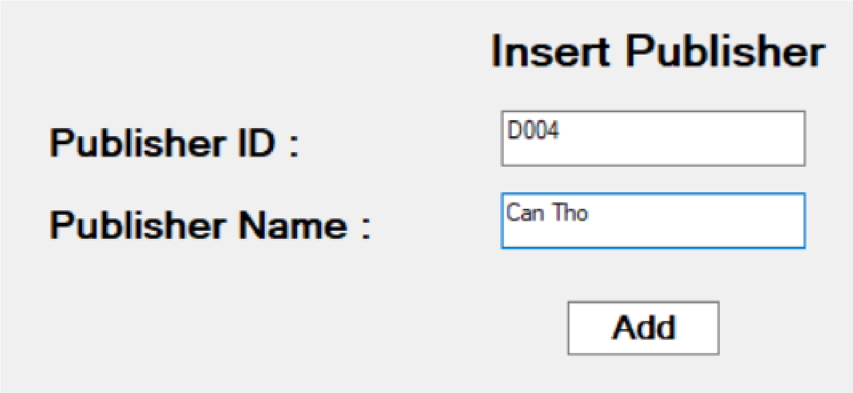
Reader ID : C004

+ The user enters ReaderID need delete into the textbox and clicks 'Delete'.

◆ Interfaces add Publisher.

- PublisherID varchar(20)
- PublisherName varchar(20)

+ User enter PublisherID no more than 20 characters, Publisher name is limited by 20 characters.

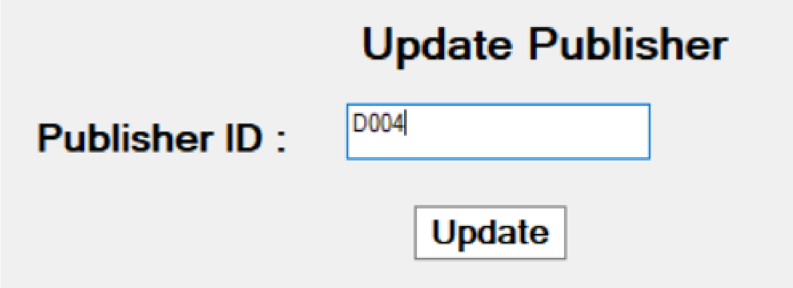


Insert Publisher

Publisher ID : D004

Publisher Name : Can Tho

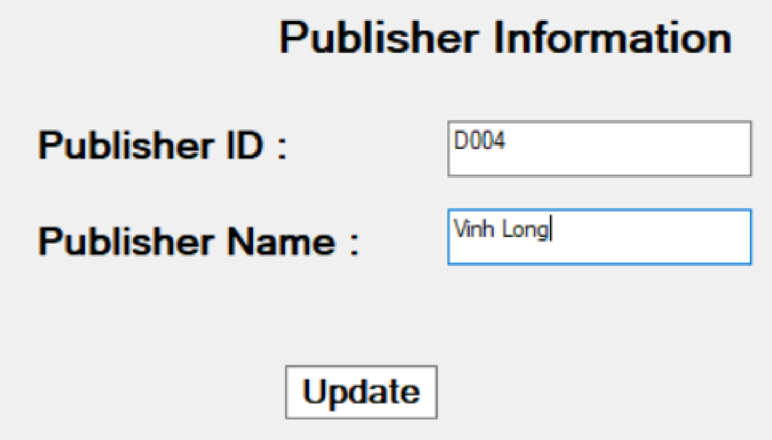
+ Interfaces Update Publisher.



Update Publisher

Publisher ID : D004

+ User enters Publisher ID to update



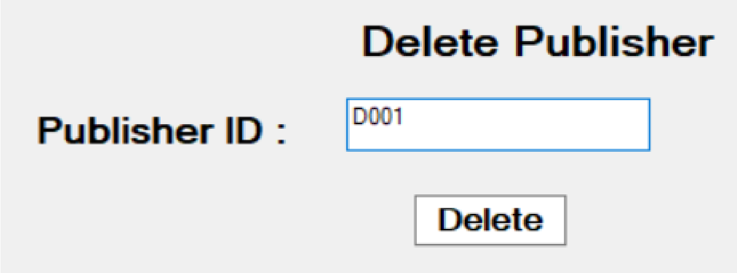
Publisher Information

Publisher ID : D004

Publisher Name : Vinh Long

The user enters Publischer information need update into the textbox and clicks 'update'. Update information requires the user to enter valid information, Publisher ID and without special characters.

+ Interfaces Delete Reader

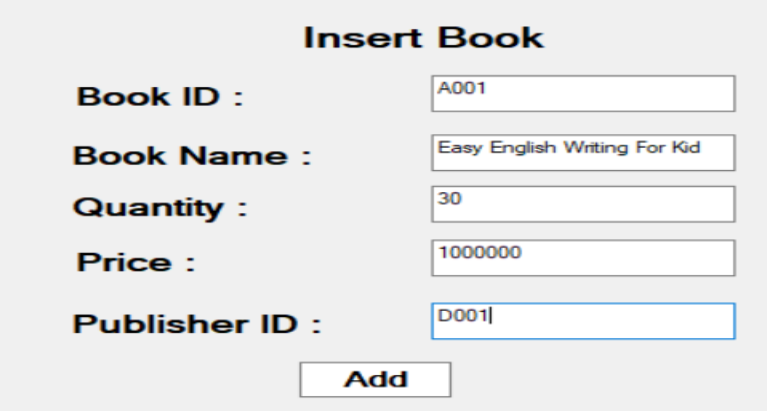


Delete Publisher

Publisher ID : D001

The user enters Publischer information need Delete into the textbox and clicks 'Delete'. Delete information requires the user into enter valid information, Publisher ID and without special characters.

◆ Interfaces add Book



Insert Book

Book ID : A001

Book Name : Easy English Writing For Kid

Quantity : 30

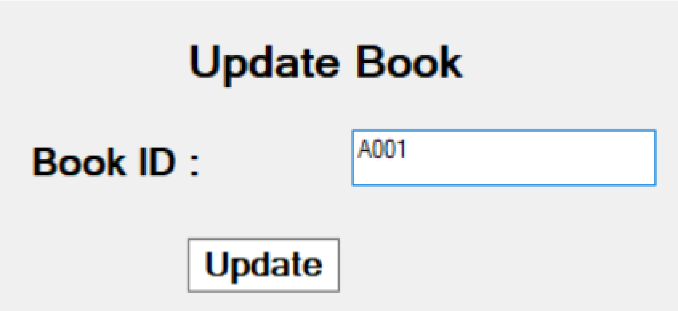
Price : 1000000

Publisher ID : D001

- BookID char(10)
- BookName varchar(50)
- Quantity int
- Price int

+ User enter BookID no more than 10 char, Book name is limited by 50 char, Quantity number must not enter letters or special characters, and price number must not enter letters or special characters.

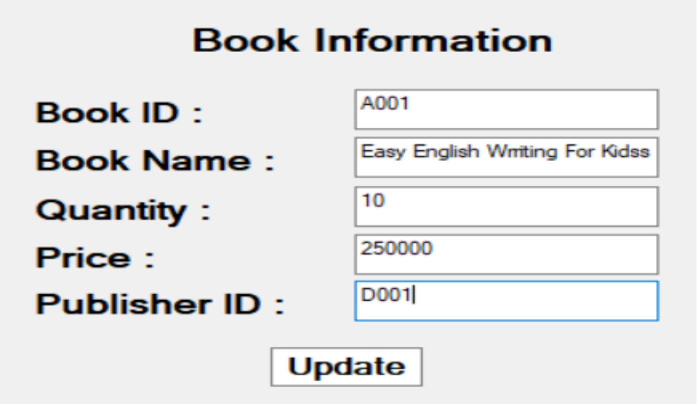
+ Interfaces Update Book.



Update Book

Book ID : A001

+ User enters Book ID to update.



Book Information

Book ID : A001

Book Name : Easy English Wrriting For Kidss

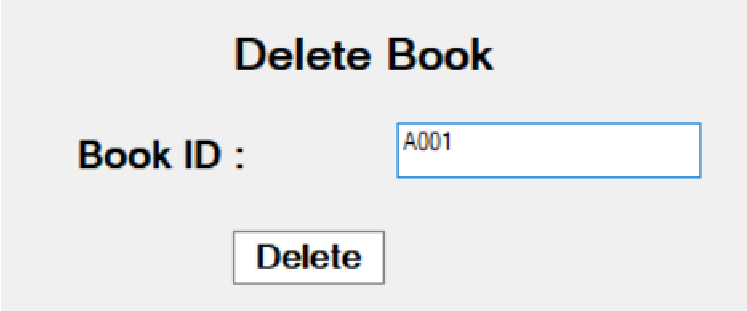
Quantity : 10

Price : 250000

Publisher ID : D001

The user enters Book information need update into the textbox and clicks 'update'. Update information requires the user to enter valid information, Book ID and Book name, quantity, price, and without special characters.

+ Interfaces Delete Book



Delete Book

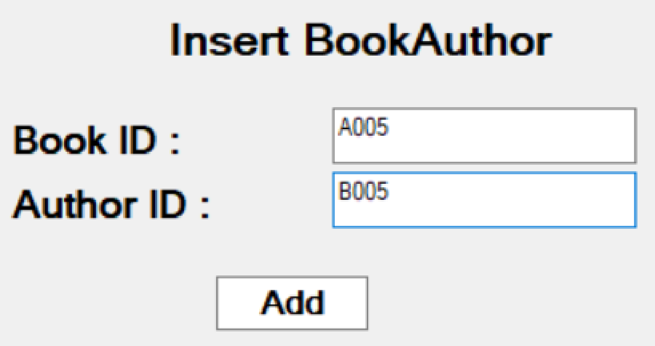
Book ID : A001

The user enters Book ID need Delete into the textbox and clicks 'Delete'. Delete information requires the user into Book ID and without special characters.

Interfaces add BookAuthor.

- BookID char(10)
- AuthorID varchar(20)

+ User enter BookID no more than 10 char, AuthorID is limited by 20 char.

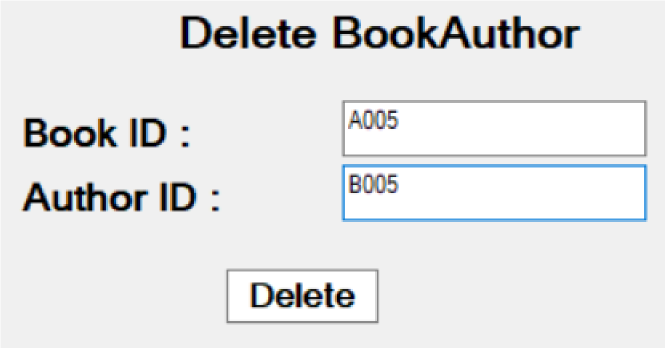


Insert BookAuthor

Book ID : A005

Author ID : B005

+ Interfaces delete BookAuthor



Delete BookAuthor

Book ID : A005

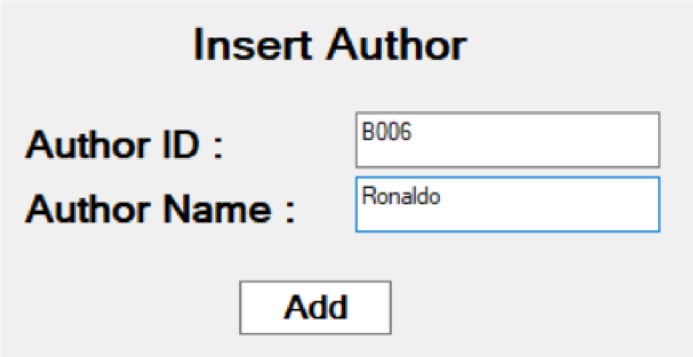
Author ID : B005

The user enters Book ID and AuthorID need Delete into the textbox and clicks 'Delete'. Delete information requires the user into Book ID and without special characters.

Interfaces add Author

- AuthorID varchar(20
- AuthorName varchar(20)

+ User enter AuthorID no more than 10 characters, AuthorName is limited by 20 characters.

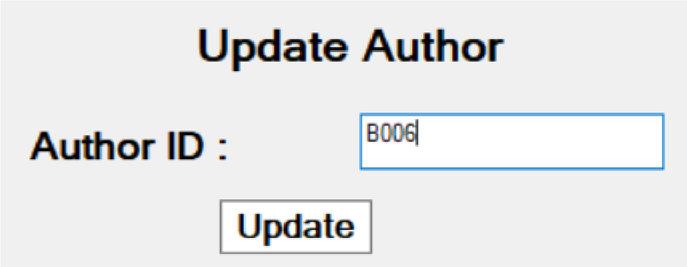


Insert Author

Author ID : B006

Author Name : Ronaldo

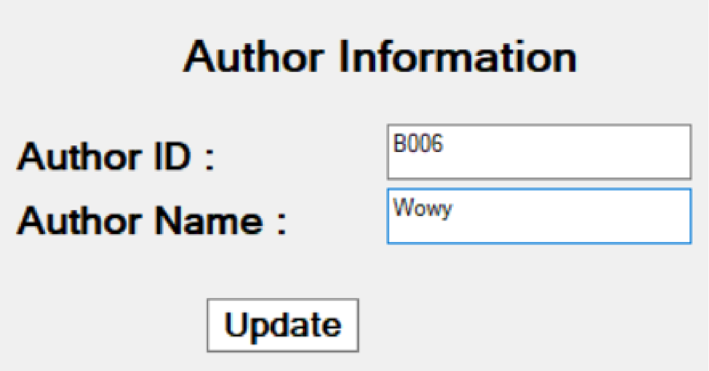
+ Interfaces update Author



Update Author

Author ID : B006

+ User enters Author ID to update.



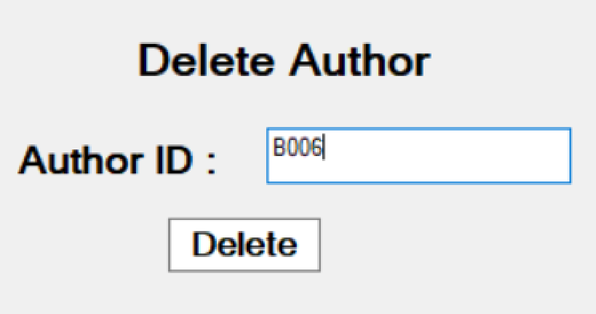
Author Information

Author ID : B006

Author Name : Wowy

The user enters Author information need update into the textbox and clicks 'update'. Update information requires the user to enter valid information, Author ID and Author name, and without special characters.

+ Interfaces Delete Author



Delete Author

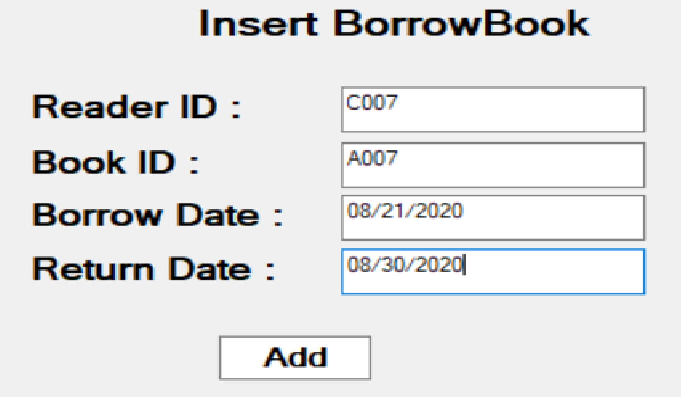
Author ID : B006

The user enters AuthorID need Delete into the textbox and clicks 'Delete'. Delete information requires the user into Author ID and without special characters.

Interfaces add BorrowBook

- ReaderID char(10)
- BookID char(10)
- BorrowDate datetime
- ReturnDate datetime

+ User enter BookID, and readerID no more than 10 characters, BorrowDate and ReturnDate number must not enter letters or special characters.



Insert BorrowBook

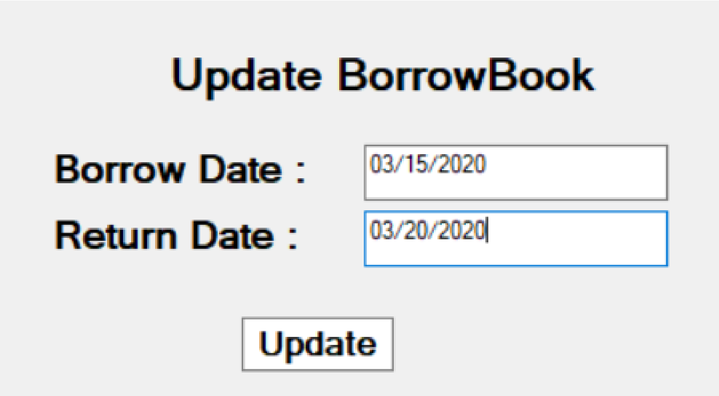
Reader ID : C007

Book ID : A007

Borrow Date : 08/21/2020

Return Date : 08/30/2020

+ Interfaces update BorrowBook



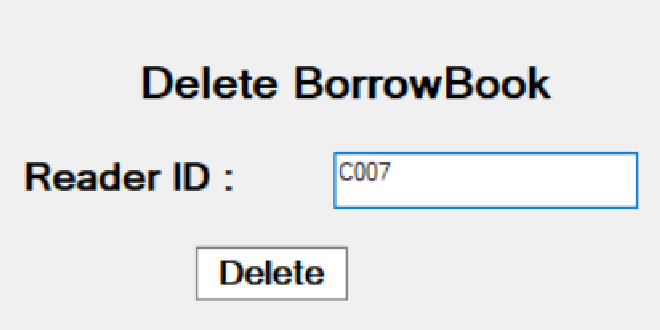
Update BorrowBook

Borrow Date : 03/15/2020

Return Date : 03/20/2020

The user enters Borrow Date, Return Date information need update into the textbox and clicks 'update'. Update information requires the user to enter valid information, BorrowDate and ReturnDate, and without special characters.

+ Interfaces Delete BorrowBook



Delete BorrowBook

Reader ID : C007

The user enters ReaderID need Delete into the textbox and clicks 'Delete'. Delete information requires the user into reader ID and without special characters.

APENDIX:

```
create database LIBRARY
use LIBRARY
create table Reader
(
	ReaderID char(10) not null primary key,
	ReaderName varchar(50) not null,
	Phone int,
	Address char(10) not null
)
INSERT INTO Reader VALUES ('C001','VO NHUT HUY','0947605978','VINH LONG')
INSERT INTO Reader VALUES ('C002','NGUYEN MINH TU','0972078173','DONG THAP')
INSERT INTO Reader VALUES ('C003','PHAM CAO NGUYEN','0981424285','VINH LONG')
select * from Reader
create table Publisher
(
	PublisherID varchar(20) not null primary key,
        PublisherName varchar(20) not null
)
INSERT INTO Publisher VALUES ('D001','HCM')
INSERT INTO Publisher VALUES ('D002','HN')
INSERT INTO Publisher VALUES ('D003','HCM')
select * from Publisher

create table Book
(
        BookID char(10) not null primary key,
        BookName varchar(50) not null,
        Quantity int not null,
        Price int,
        PublisherID varchar(20) not null foreign key references Publisher(PublisherID)
)
INSERT INTO Book VALUES ('A001','The Magic of Thinking BIG',50,100000,'D001')
INSERT INTO Book VALUES ('A002','Secrets of Millionaire Mind',60,500000,'D002')
INSERT INTO Book VALUES ('A003','The Richest Man in Babylon',40,900000,'D003')
select * from Book
create table BorrowBook
(
        ReaderID char(10) foreign key(ReaderID) references Reader(ReaderID),
        BookID char(10) foreign key(BookID) references Book(BookID),
        BorrowDate datetime,
        ReturnDate datetime,
        primary key(ReaderID, BookID, BorrowDate)
)
INSERT INTO BorrowBook VALUES ('C001','A001','01/11/2020','01/18/2020')
INSERT INTO BorrowBook VALUES ('C002','A002','09/10/2020','09/17/2020')
INSERT INTO BorrowBook VALUES ('C003','A003','11/01/2020','11/08/2020')
select * from BorrowBook
create table Author
(
        AuthorID varchar(20) not null primary key,
        AuthorName varchar(20) not null,
)
INSERT INTO Author VALUES ('B001','David J. Schwartz')
INSERT INTO Author VALUES ('B002','T. Harv Eker')
INSERT INTO Author VALUES ('B003','George Samuel Clason')
select * from Author
create table BookAuthor
(
        BookID char(10) not null foreign key references Book(BookID),
        AuthorID varchar(20) not null foreign key references Author(AuthorID),
        primary key(BookID, AuthorID)
)
INSERT INTO BookAuthor VALUES ('A001','B001')
INSERT INTO BookAuthor VALUES ('A002','B002')
INSERT INTO BookAuthor VALUES ('A003','B003')
select * from BookAuthor
```

```
------------------------------------------------select---------------------
-- Display book borrow in November.
select BookID, BorrowDate from [dbo].[BorrowBook] where month(BorrowDate)=11
-- Display book return reader in September
select ReaderID, ReturnDate from [dbo].[BorrowBook] where month(ReturnDate)=09
-- Displays book with a quantity greater than 10
select * from [dbo].[Book]
where Quantity>10
-- Displays book with price greater than 600000
select BookID,Price as 'total Price Book'
from [dbo].[Book]
where Price>600000
-- Display the sum of all book with quantity above 40.
select BookID, sum(Quantity) as 'total of all book'
from [dbo].[Book]
group by BookID
having sum(Quantity)>40
-----------------------------------------------------------------INSERT--------------------------
--INSERT Reader
insert into [dbo].[Reader] (ReaderID,ReaderName,Phone,Address)
values('C004','PHAN BOI CHAU','0909876543','CAN THO')
select * from Reader
--INSERT Publisher
insert into [dbo].[Publisher] (PublisherID,PublisherName)
values('D006','Can Tho')
select * from Publisher
--INSERT Book
insert into [dbo].[Book] (BookID,BookName, Quantity, Price, PublisherID)
values('A001','Easy English Writing For Kid',30,1000000,'D001')
select * from [dbo].[Book]
--INSERT [dbo].[BookAuthor]
insert into [dbo].[Publisher] (PublisherID,PublisherName)
values('D006','Da Nang')
insert into [dbo].[Book] (BookID,BookName, Quantity, Price, PublisherID)
values('A005','Summit 2','80',2000000,'D005')
insert into [dbo].[Author] (AuthorID, AuthorName)
values ('B005', 'Messi')
insert into [dbo].[BookAuthor] (BookID, AuthorID)
values('A005','B005')
select * from [dbo].[BookAuthor]
--INSERT [dbo].[Author]
insert into [dbo].[Author] (AuthorID, AuthorName)
values ('B006', 'Ronaldo')
select * from [dbo].[Author]
--INSERT [dbo].[BorrowBook]
insert into [dbo].[Reader] (ReaderID,ReaderName,Phone,Address)
values('C007','Bin','0896693099','HA NOI')
insert into [dbo].[Publisher] (PublisherID,PublisherName)
values('D007','HCM')
insert into [dbo].[Book] (BookID,BookName, Quantity, Price, PublisherID)
```

```
values('A007','RARY ZUKAV THE SEAT OF THE SOUL',85,1500000,'D007')
Insert into [dbo].[BorrowBook] (ReaderID, BookID, BorrowDate, ReturnDate)
values ('C007', 'A007', '08/21/2020', '08/30/2020')
select * from [dbo].[BorrowBook]
---------------------------------------------------------------UPDATE---------------------------
--UPDATE Reader
update Reader set ReaderName = 'NGUYEN NGOC NGAN', Address = 'O MON', Phone = 0986250461
where ReaderID='C004'
select * from Reader
--UPDATE [dbo].[Publisher]
update [dbo].[Publisher] set PublisherID='D004'
where PublisherName ='Vinh Long'
select * from [dbo].[Publisher]
--UPDATE [dbo].[Book]
update [dbo].[Book] set BookID='A001',BookName='Easy English Writing For
Kidss',Quantity=10,Price=250000
where PublisherID = 'D001'
select * from [dbo].[Book]
update [dbo].[Book] set BookID = 'A004', Quantity = 10, Price = 200000
where BookName = 'Easy English Writing For Kid'
select * from [dbo].[Book]
--UPDATE [dbo].[Author]
update [dbo].[Author] set AuthorName ='Wowy'
where AuthorID='B006'
select * from [dbo].[Author]
--UPDATE [dbo].[BorrowBook]
Update [dbo].[BorrowBook] set ReaderID = 'C007', BorrowDate = '03/15/2020', ReturnDate =
'03/20/2020'
where BookID = 'A007'
select * from [dbo].[BorrowBook]
--------------------------------------------------------------DELETE---------------------------
-- DELETE [dbo].[Reader]
DELETE from [dbo].[Reader] where ReaderID = 'C004'
select * from Reader
-- DELETE [dbo].[Publisher]
DELETE from [dbo].[BorrowBook] where BookID ='A001'
DELETE from [dbo].[BookAuthor] where BookID='A001'
DELETE from [dbo].[Book] where BookID='A001'
DELETE FROM [dbo].[Publisher] WHERE PublisherID = 'D007'
select * from [dbo].[Publisher]
-- DELETE [dbo].[Book]
DELETE from [dbo].[Book] where BookID = 'A007'
select * from [dbo].[Book]
-- DELETE [dbo].[BookAuthor]
DELETE from [dbo].[BookAuthor] where BookID='A005' and AuthorID ='B005'
select * from [dbo].[BookAuthor]
-- DELETE [dbo].[Author]
Delete from [dbo].[Author] where AuthorID = 'B006'
select * from [dbo].[Author]
-- DELETE [dbo].[BorrowBook]
```

```
Delete from [dbo].[BorrowBook] where ReaderID = 'C007'
select * from [dbo].[BorrowBook]
select * from [dbo].[Book]
---------------------------------Triggers-----------------------------
go
--------------------------------------------------------insert--------------------------------
--Verify that book number will be saved in the Book table when the reader enters valid 
information(<100)
--Verify that the book number will not be saved in the book table when the reader inserts invalid 
information(>100)
create trigger insert_triggers_Bookaa
on Book
for insert
as
	if (select Quantity from inserted) >100
begin
	print ('can not quantity greater than 100')
	Rollback transaction
end
--test
insert into [dbo].[Publisher] values ('D007', 'HN')
insert into [dbo].[Book] values('A007','summit',200,100000,'D007')

insert into [dbo].[Publisher] values ('D007', 'HN')
insert into [dbo].[Book] values('A007','summit',50,100000,'D007')
select * from [dbo].[Book]
go
---------------------------------------------------------update-------------------------------
--Verify that the book information is saved in the book table when the reader updates valid 
information
--Verify that the book information is not saved to the book table when the reader updates the 
invalid information
Create trigger Update_triggers_Bookss
on Book
for Update
as
if ( select price from inserted) > (select price from deleted)
begin
	print('Update failed')
	Rollback transaction
end
--test
update [dbo].[Book]
set price = 200000
where BookName = 'The Magic of Thinking BIG'
update [dbo].[Book]
set price = 50000
where BookName = 'The Magic of Thinking BIG'
select * from [dbo].[Book]
go
```

```
----------------------------------------------------------Delete----------------------------
--Verify that the reader wants to delete the book information by entering a valid BookID code
Create trigger Delete_triggers_Bookdd
on [dbo].[Book]
INSTEAD OF DELETE
as
   begin
      delete from [dbo].[BorrowBook] where BookID in (select BookID from deleted)
         delete from [dbo].[BookAuthor] where BookID in (select BookID from deleted)
         delete from [dbo].[Book] where BookID in (select BookId from deleted)
   end
--test
Delete From [dbo].[Book] where BookID='A001'
drop trigger casc_delete
select * from [dbo].[Book]
select * from[dbo].[BookAuthor]
select * from [dbo].[BorrowBook]
select * from[dbo].[Publisher]
select * from[dbo].[Reader]
```

References

Ponniah, P., 2003. *Database design and development* (pp. 274-298). Ed. Wiley Interscience. New York.