Books

SQL, One-To-Many Joins

Overview: In this scenario there are two tables. One for library books and one for members who can check out the books. Each member can check out multiple books at once, but each book can only be checked out by one member at a time. Create the tables and fill them with data.

Tables:

book
id: INT <PK> title: NVARCHAR 60 author: NVARCHAR 60 checked_out_by_id: INT <FK>

member
id: INT <PK> name: NVARCHAR 40 card_number: CHAR 10

Execute SQL queries for the following. Keep the queries in a text file if you like.

1. Create both tables.
2. Add the following rows to the **member** table.
    a. Name: “Annabelle Aster”, Card Number: “772-93-110”
    b. Name: “Boris Berceli”, Card Number: “000-00-000”
    c. Name: “Carter Corbin”, Card Number: “282-09-382”
3. Add the following rows to the **book** table.
    a. Title: “In Search of Lost Time”, Author: “Marcel Proust”, Checked out by: Annabelle.
    b. Title: “Ulysses”, Author: “James Joyce”, Checked out by: Annabelle.
    c. Title: “Don Quixote”, Author: “Miguel de Cervantes”, Checked out by: Carter.
    d. Title: “Moby Dick”, Author: “Herman Melville”, Not checked out (null).
4. Update the member with the id of 2 to have a card number of “357-15-964”.
5. Remove the member with the id of 2.
6. Select all members that have the card number 772-93-110.
7. Select all books sorted by title
8. Annabelle checked out another book... Update Moby Dick to be checked out by Annabelle.
9. Annabelle turned in a book... Update Ulysses to be checked out by no one (null).
10. Write a JOIN that lists all the books and who they’re checked out to. Include the books that are not checked out.
11. Write a JOIN that lists the titles of all the books that are checked out to someone named Annabelle Aster (don't just use the ID)

12. Write a JOIN that lists the name and card number of all members who have checked out books by the author Herman Melville.

Check Your Solutions

1.
```
CREATE TABLE book (
  id INT IDENTITY(1,1) PRIMARY KEY,
  title NVARCHAR(60),
  author NVARCHAR(60),
  checked_out_by_id INT);
CREATE TABLE member (
  id INT IDENTITY(1,1) PRIMARY KEY,
  name NVARCHAR(40),
  card_number CHAR(10));
```
2.
```
INSERT INTO member (name, card_number) VALUES ('Annabelle Aster',
'772-93-110');
INSERT INTO member (name, card_number) VALUES ('Boris Berceli',
'000-00-000');
INSERT INTO member (name, card_number) VALUES ('Carter Corbin',
'282-09-382');
```
3.
```
INSERT INTO book (title, author, checked_out_by_id) VALUES ('In Search
of Lost Time', 'Marcel Proust', 1);
INSERT INTO book (title, author, checked_out_by_id) VALUES ('Ulysses',
'James Joyce', 1);
INSERT INTO book (title, author, checked_out_by_id) VALUES ('Don
Quixote', 'Miguel de Cervantes', 3);
INSERT INTO book (title, author) VALUES ('Moby Dick', 'Herman
Melville');
```
4.
```
UPDATE member SET card_number = '357-15-964' WHERE id = 2;
```
5.
```
DELETE FROM member WHERE id = 2;
```
6.
```
SELECT * FROM member WHERE card_number = '772-93-110';
```
7.
```
SELECT * FROM book ORDER BY title;
```
8.
```
UPDATE book SET checked_out_by_id = 1 WHERE id = 4;
 - or -
UPDATE book SET checked_out_by_id = 1 WHERE title = 'Moby Dick';
```
9.
```
UPDATE book SET checked_out_by_id = NULL WHERE id = 2;
 - or -
UPDATE book SET checked_out_by_id = NULL WHERE title = 'Ulysses';
```
10.
```
SELECT * FROM book LEFT JOIN member ON book.checked_out_by_id =
member.id;
```
11.
```
SELECT book.title FROM book JOIN member ON book.checked_out_by_id =
member.id WHERE member.name = 'Annabelle Aster';
```
12.
```
SELECT member.name, member.card_number FROM book JOIN member ON
book.checked_out_by_id = member.id WHERE book.author = 'Herman
Melville';
```