Mini project

Library database.

Step #2: The Description and domains.

This is a Library database for a local library, it has only one location and no shared book with other library. (As we marked **bold** for relation table name)

In our circumstance, anyone recorded in the library can borrow a book as the inventory has enough non-borrowed book, and it will be marked a date for return, once it is pass due, the fine will be charged to the person in his record. All the activities are held by library in social room, the room is unique and no other room are available in this library (means we do not have a room table for reserve or so, other time with no activity, social room will be closed.) All the activity will be scheduled in Event table, all are fixed with specific activity, time and date.

All the book information and type are stored in a **Book** Relation table, included bookID as primary key. Book name (title), type, publish year, ISBN, author and edition are included, also an information about pass due fines refer to each book are recorded, call fine. This table is like a dictionary type for searching book related detail.

Then we have **Inventory** table for all the actual stock in library, means for the same book have 5 copies, they will be held in this table with 5 different itemID as primary key, then having same bookID for foreign key. Also it records the date this book got added to library.

Then a **People** Table contains all the peopleID as primary key, the fname, lname for the full name of people, phone, email and fineAmount for the total overdue fines a waiting to pay.

For **Borrow** table contains all the borrow history, included the itemID and peopleID represent for who borrowed which book in library. Also the borrowDate is part of key, to separate same people borrow same book twice. Then a due date and borrow date are recored. Also an attribute for tracking whether this borrowed book is returned, called status.

All the activity are store in **Event** table, which has a unique eventID as primary key, Event data as Monday to Sunday, time slot as start time and end time, the type are also recorded.

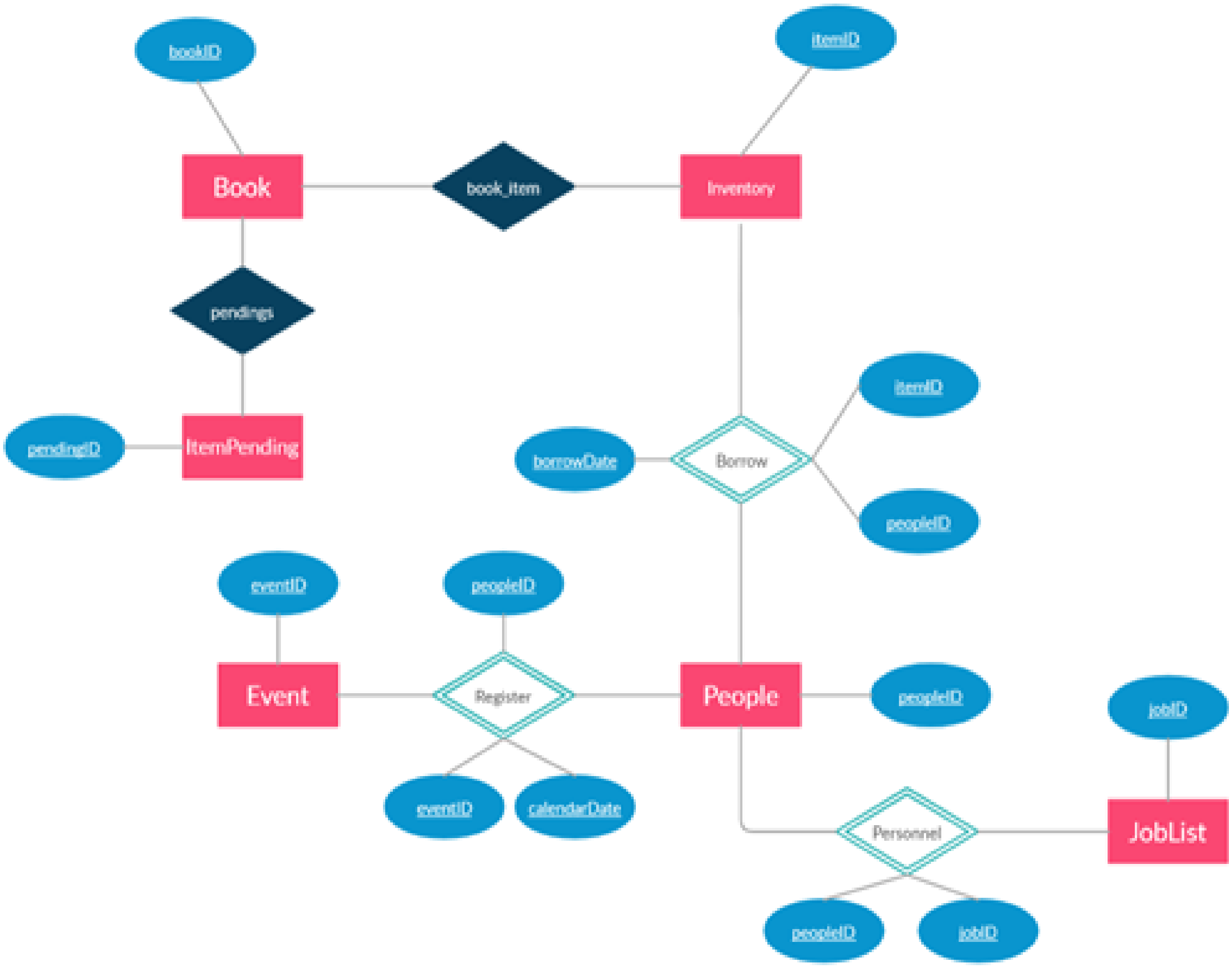
In order to define who want to join an activity on what date, we have a **register** table containing PeopleID, eventID and calendarDate all as primary key to track on people's activity.

Personnel is belong to People table, but it has jobID foreign key to jobList for all the different jobs in library. This table holds all the current workers information here. (Like shift and the date a personnel join the library)

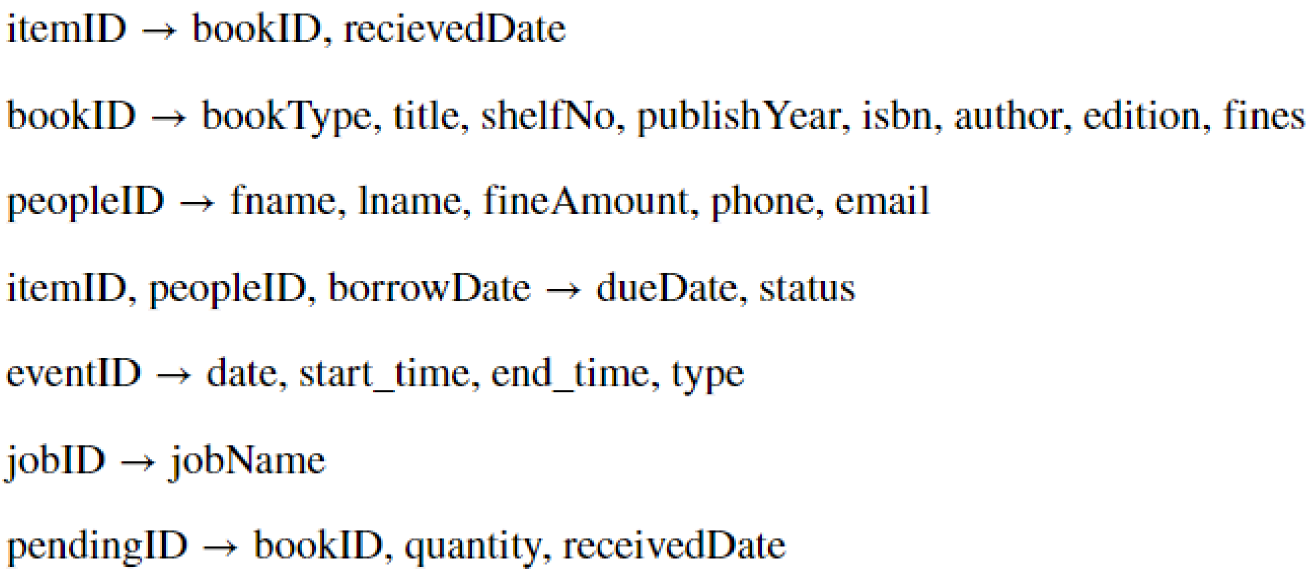
jobList contains all different types of job included volunteer, this contains all different types of job in library, included manager, sales, cleaner, librarian, counter, volunteer and so on. Also we could contain the salary and work hour needed here for each type of job.

itemPending is related to book, which it has pendingID to reference to any bookID, quantity and recievedDate. Quantity and received date can be null since we are not sure when and how much will come to library. Also when inserting this table, we need to check the information in book first, if our current booklist does not have this book (a new book comes in), add the new nook information to **Book,** then add to **itemPending**.

Step #3: The Entity Relationship Model.



Step #4: Functional Dependencies.



itemID → bookID, recievedDate

bookID → bookType, title, shelfNo, publishYear, isbn, author, edition, fines

peopleID → fname, lname, fineAmount, phone, email

itemID, peopleID, borrowDate → dueDate, status

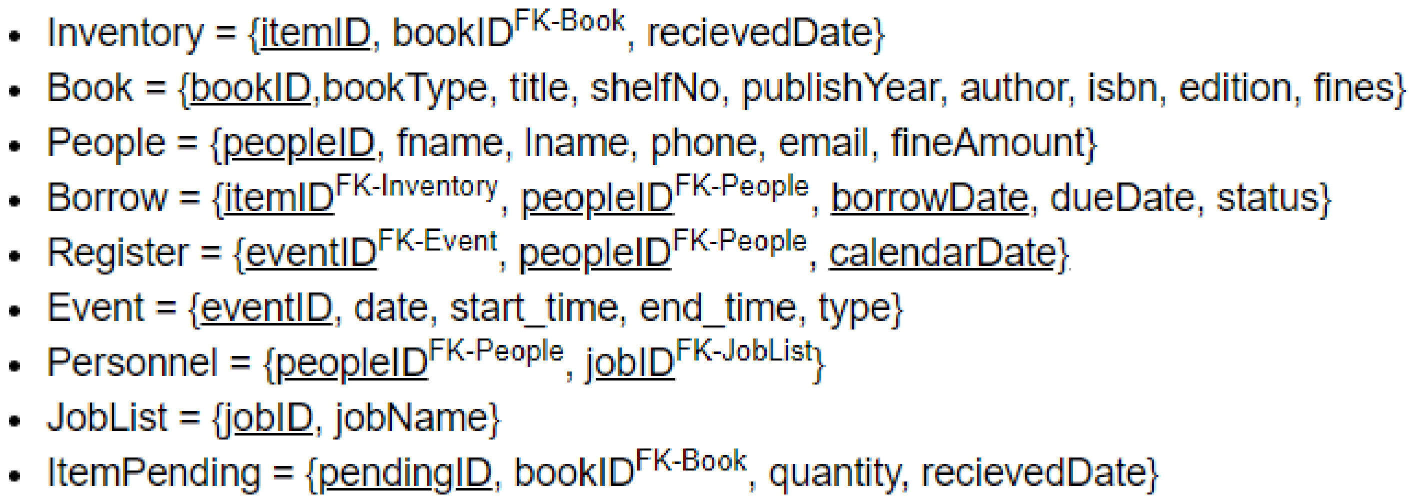
eventID → date, start_time, end_time, type

jobID → jobName

pendingID → bookID, quantity, receivedDate

Step #5: Schema & Database creation.

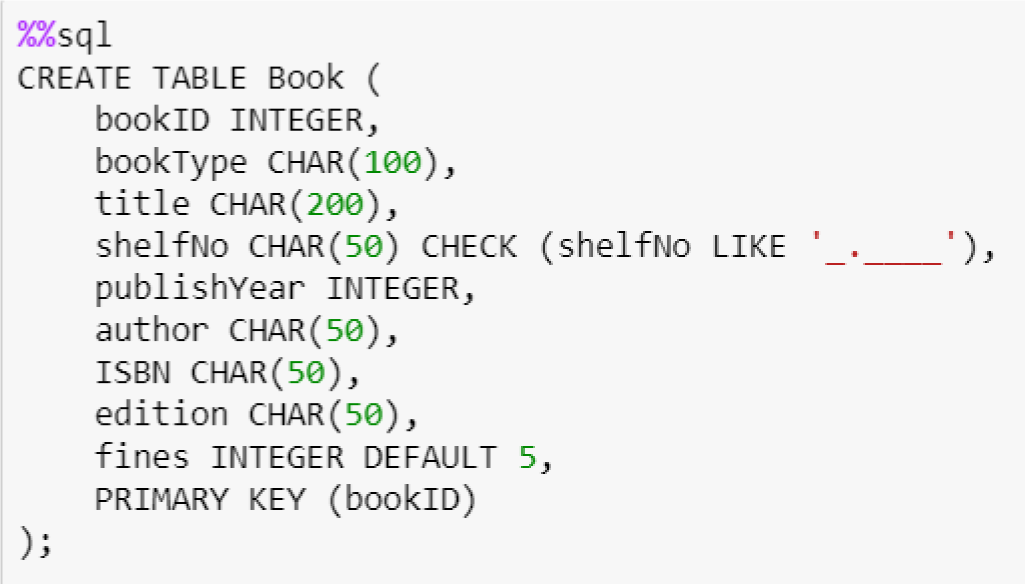
Schema:



- Inventory = {itemID, bookID$^{FK\text{-}Book}$, recievedDate}
- Book = {bookID,bookType, title, shelfNo, publishYear, author, isbn, edition, fines}
- People = {peopleID, fname, lname, phone, email, fineAmount}
- Borrow = {itemID$^{FK\text{-}Inventory}$, peopleID$^{FK\text{-}People}$, borrowDate, dueDate, status}
- Register = {eventID$^{FK\text{-}Event}$, peopleID$^{FK\text{-}People}$, calendarDate}
- Event = {eventID, date, start_time, end_time, type}
- Personnel = {peopleID$^{FK\text{-}People}$, jobID$^{FK\text{-}JobList}$}
- JobList = {jobID, jobName}
- ItemPending = {pendingID, bookID$^{FK\text{-}Book}$, quantity, recievedDate}

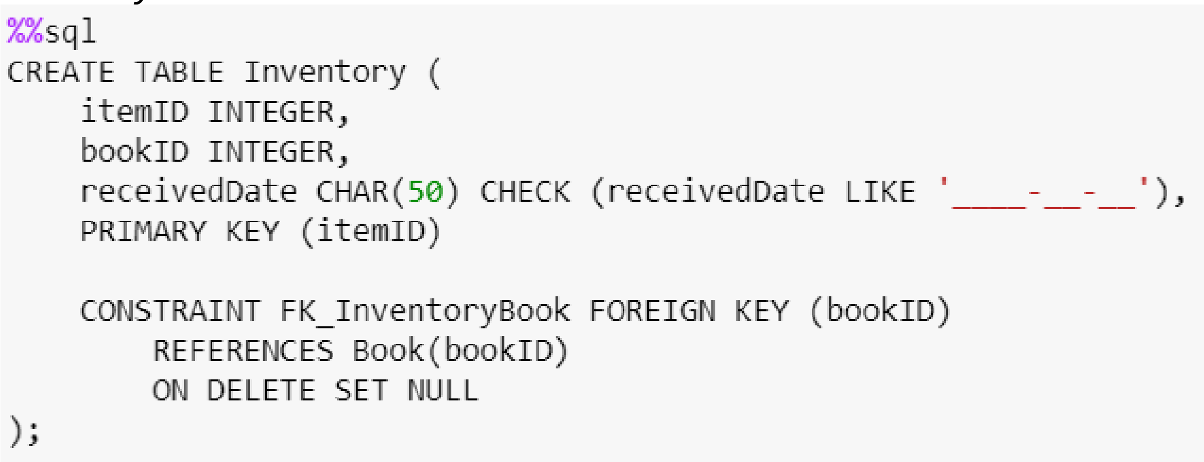
Database creation:

1. *Book Table*



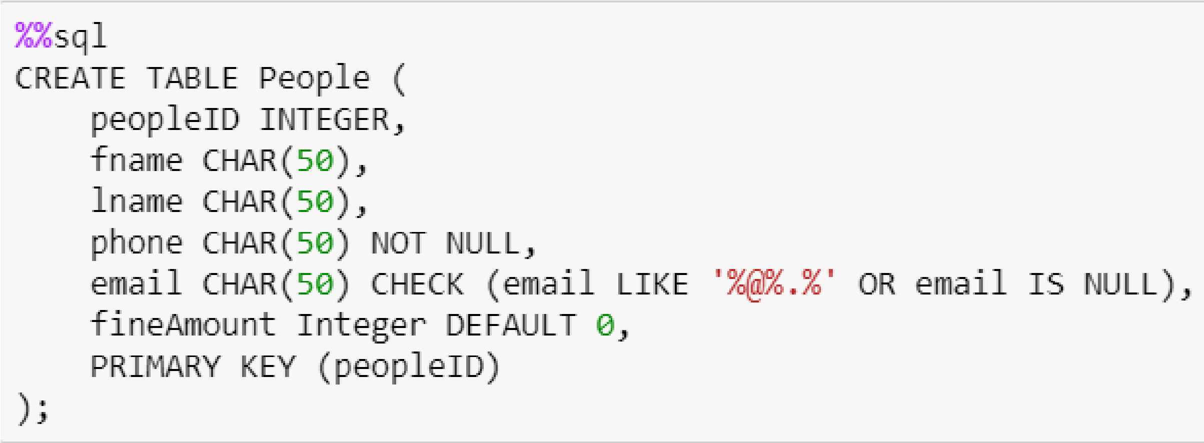
```
%%sql
CREATE TABLE Book (
    bookID INTEGER,
    bookType CHAR(100),
    title CHAR(200),
    shelfNo CHAR(50) CHECK (shelfNo LIKE '_.____'),
    publishYear INTEGER,
    author CHAR(50),
    ISBN CHAR(50),
    edition CHAR(50),
    fines INTEGER DEFAULT 5,
    PRIMARY KEY (bookID)
);
```

2. *Inventory table*



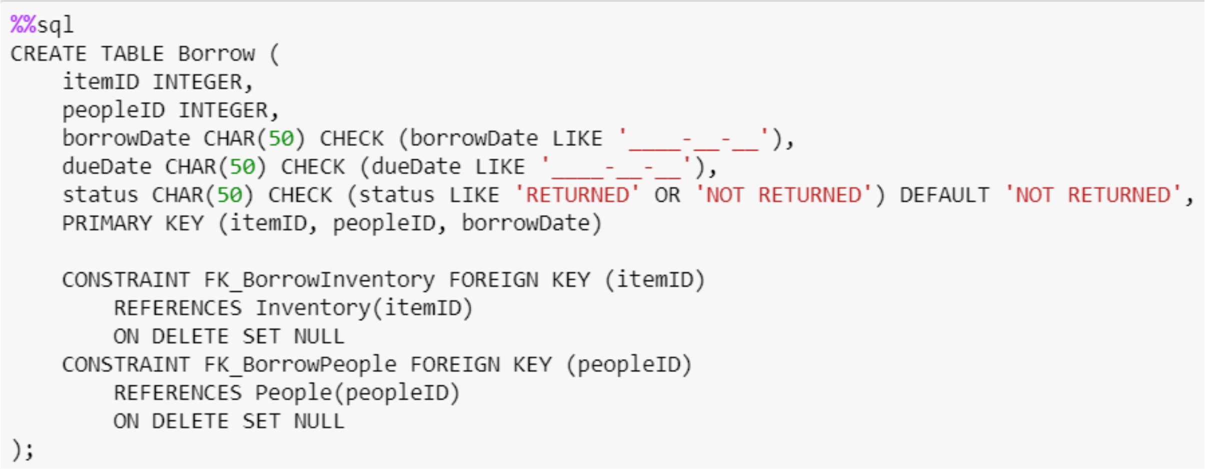
```
%%sql
CREATE TABLE Inventory (
    itemID INTEGER,
    bookID INTEGER,
    receivedDate CHAR(50) CHECK (receivedDate LIKE '____-__-__'),
    PRIMARY KEY (itemID)

    CONSTRAINT FK_InventoryBook FOREIGN KEY (bookID)
        REFERENCES Book(bookID)
        ON DELETE SET NULL
);
```

3. *People table*



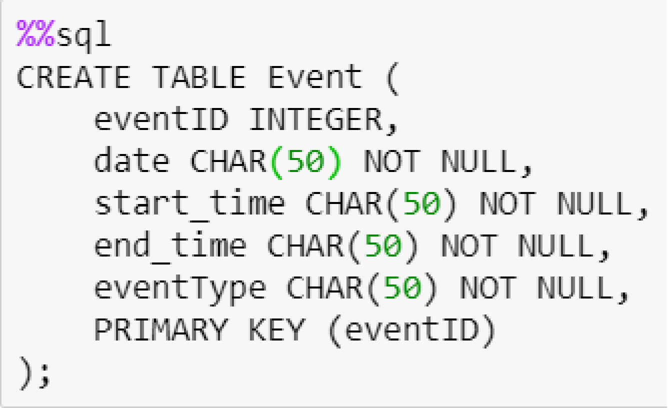
```
%%sql
CREATE TABLE People (
    peopleID INTEGER,
    fname CHAR(50),
    lname CHAR(50),
    phone CHAR(50) NOT NULL,
    email CHAR(50) CHECK (email LIKE '%@%.%' OR email IS NULL),
    fineAmount Integer DEFAULT 0,
    PRIMARY KEY (peopleID)
);
```

4. *Borrow table*



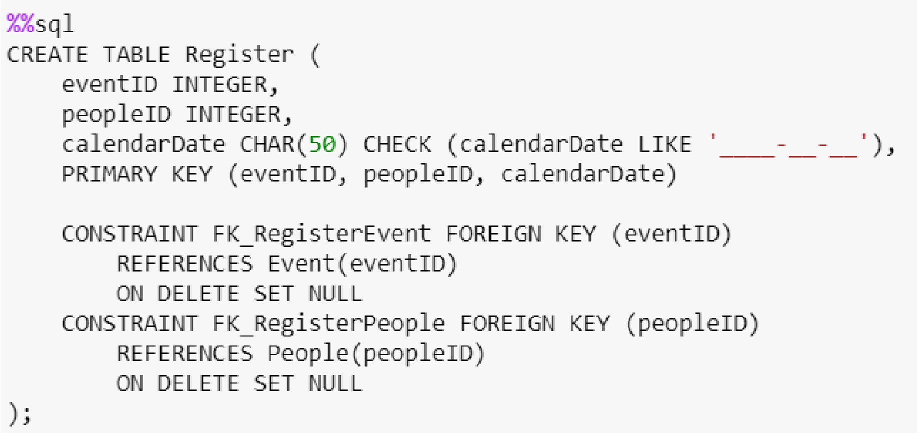
```
%%sql
CREATE TABLE Borrow (
    itemID INTEGER,
    peopleID INTEGER,
    borrowDate CHAR(50) CHECK (borrowDate LIKE '____-__-__'),
    dueDate CHAR(50) CHECK (dueDate LIKE '____-__-__'),
    status CHAR(50) CHECK (status LIKE 'RETURNED' OR 'NOT RETURNED') DEFAULT 'NOT RETURNED',
    PRIMARY KEY (itemID, peopleID, borrowDate)

    CONSTRAINT FK_BorrowInventory FOREIGN KEY (itemID)
        REFERENCES Inventory(itemID)
        ON DELETE SET NULL
    CONSTRAINT FK_BorrowPeople FOREIGN KEY (peopleID)
        REFERENCES People(peopleID)
        ON DELETE SET NULL
);
```

5. *Event table*



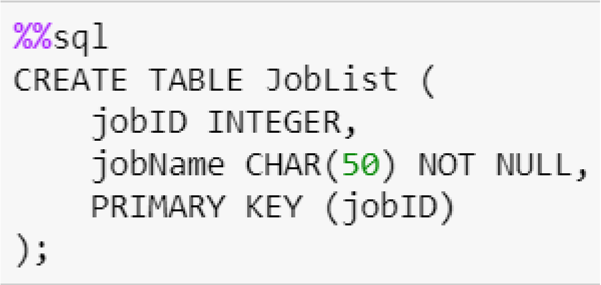
```
%%sql
CREATE TABLE Event (
    eventID INTEGER,
    date CHAR(50) NOT NULL,
    start_time CHAR(50) NOT NULL,
    end_time CHAR(50) NOT NULL,
    eventType CHAR(50) NOT NULL,
    PRIMARY KEY (eventID)
);
```

6. *Register table*



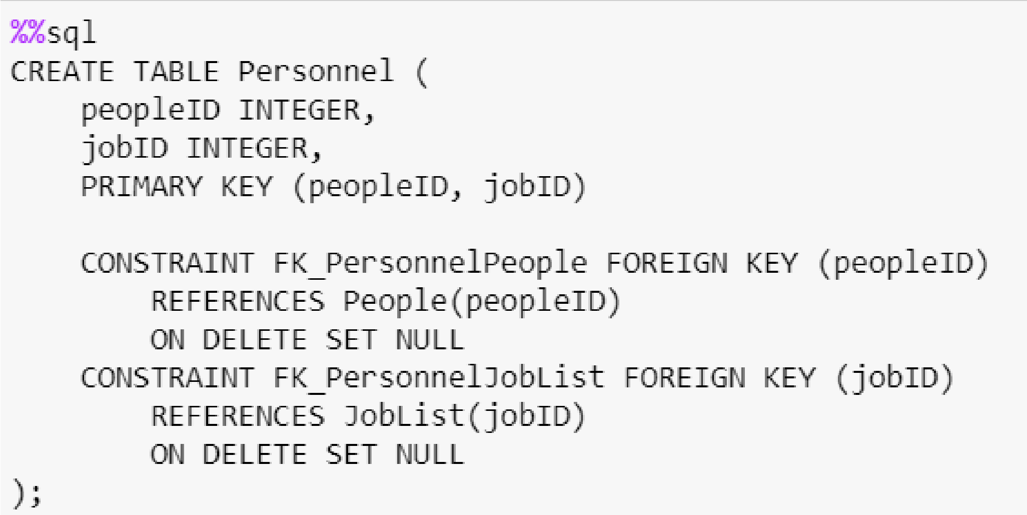
```
%%sql
CREATE TABLE Register (
    eventID INTEGER,
    peopleID INTEGER,
    calendarDate CHAR(50) CHECK (calendarDate LIKE '____-__-__'),
    PRIMARY KEY (eventID, peopleID, calendarDate)

    CONSTRAINT FK_RegisterEvent FOREIGN KEY (eventID)
        REFERENCES Event(eventID)
        ON DELETE SET NULL
    CONSTRAINT FK_RegisterPeople FOREIGN KEY (peopleID)
        REFERENCES People(peopleID)
        ON DELETE SET NULL
);
```

7. *Joblist table*



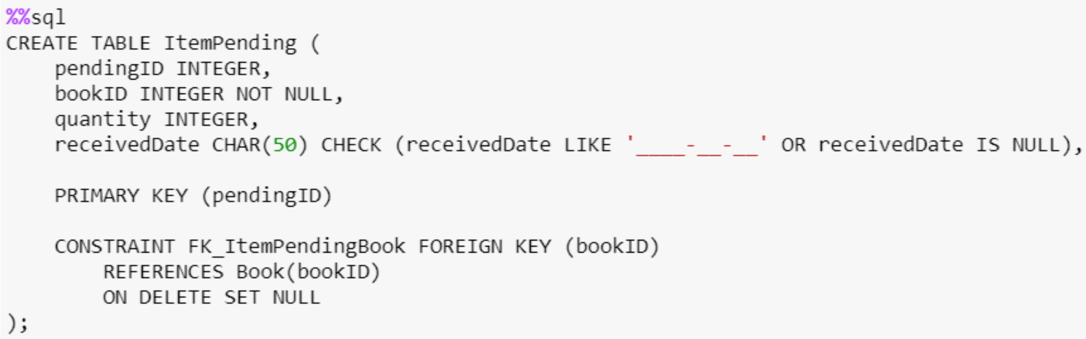
```
%%sql
CREATE TABLE JobList (
    jobID INTEGER,
    jobName CHAR(50) NOT NULL,
    PRIMARY KEY (jobID)
);
```

8. *Personnel table*



```
%%sql
CREATE TABLE Personnel (
    peopleID INTEGER,
    jobID INTEGER,
    PRIMARY KEY (peopleID, jobID)

    CONSTRAINT FK_PersonnelPeople FOREIGN KEY (peopleID)
        REFERENCES People(peopleID)
        ON DELETE SET NULL
    CONSTRAINT FK_PersonnelJobList FOREIGN KEY (jobID)
        REFERENCES JobList(jobID)
        ON DELETE SET NULL
);
```

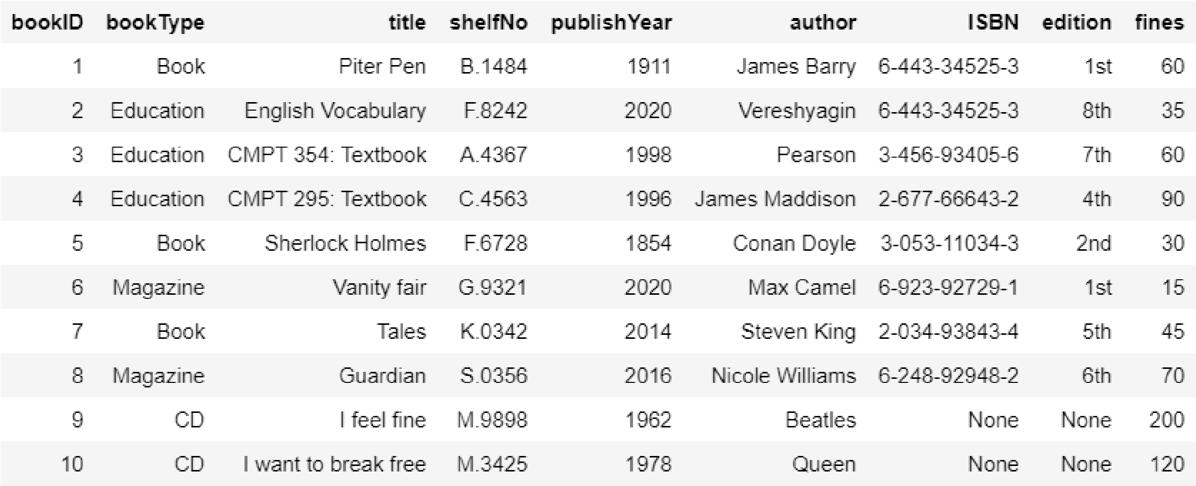
9. *ItemPending table*



```
%%sql
CREATE TABLE ItemPending (
    pendingID INTEGER,
    bookID INTEGER NOT NULL,
    quantity INTEGER,
    receivedDate CHAR(50) CHECK (receivedDate LIKE '____-__-__' OR receivedDate IS NULL),

    PRIMARY KEY (pendingID)

    CONSTRAINT FK_ItemPendingBook FOREIGN KEY (bookID)
        REFERENCES Book(bookID)
        ON DELETE SET NULL
);
```

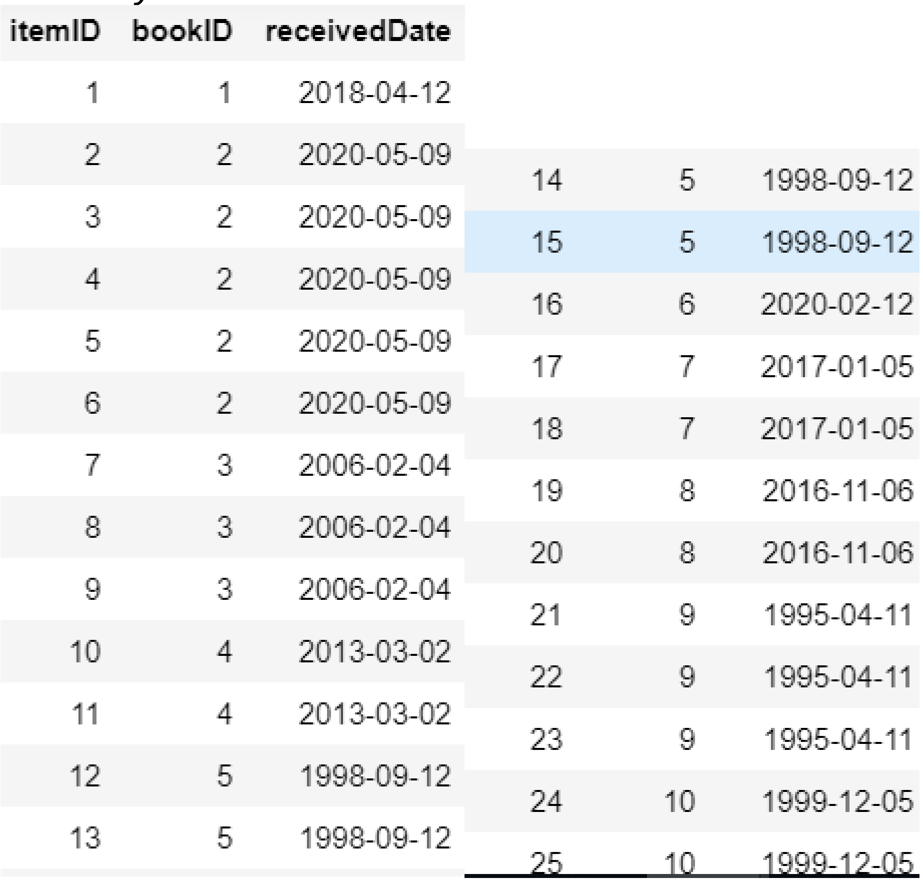
Step #6: The datasets:

1. *Book*



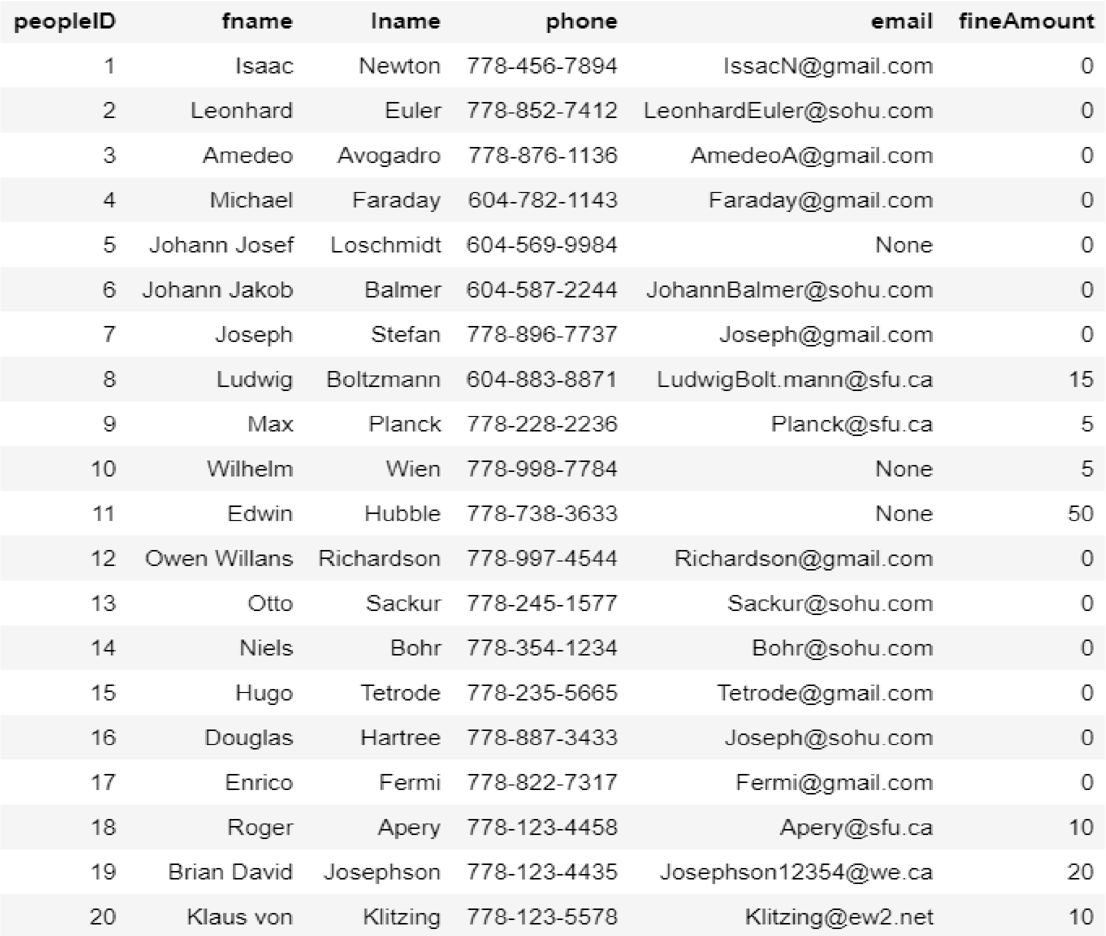
bookID	bookType	title	shelfNo	publishYear	author	ISBN	edition	fines
1	Book	Piter Pen	B.1484	1911	James Barry	6-443-34525-3	1st	60
2	Education	English Vocabulary	F.8242	2020	Vereshyagin	6-443-34525-3	8th	35
3	Education	CMPT 354: Textbook	A.4367	1998	Pearson	3-456-93405-6	7th	60
4	Education	CMPT 295: Textbook	C.4563	1996	James Maddison	2-677-66643-2	4th	90
5	Book	Sherlock Holmes	F.6728	1854	Conan Doyle	3-053-11034-3	2nd	30
6	Magazine	Vanity fair	G.9321	2020	Max Camel	6-923-92729-1	1st	15
7	Book	Tales	K.0342	2014	Steven King	2-034-93843-4	5th	45
8	Magazine	Guardian	S.0356	2016	Nicole Williams	6-248-92948-2	6th	70
9	CD	I feel fine	M.9898	1962	Beatles	None	None	200
10	CD	I want to break free	M.3425	1978	Queen	None	None	120

2. *Inventory*



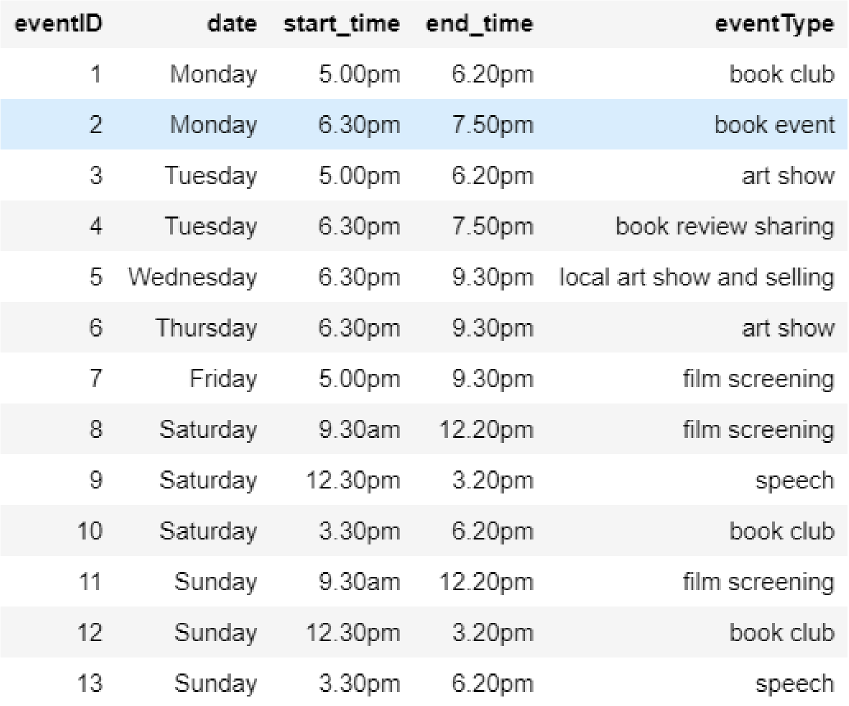
itemID	bookID	receivedDate
1	1	2018-04-12
2	2	2020-05-09
3	2	2020-05-09
4	2	2020-05-09
5	2	2020-05-09
6	2	2020-05-09
7	3	2006-02-04
8	3	2006-02-04
9	3	2006-02-04
10	4	2013-03-02
11	4	2013-03-02
12	5	1998-09-12
13	5	1998-09-12
14	5	1998-09-12
15	5	1998-09-12
16	6	2020-02-12
17	7	2017-01-05
18	7	2017-01-05
19	8	2016-11-06
20	8	2016-11-06
21	9	1995-04-11
22	9	1995-04-11
23	9	1995-04-11
24	10	1999-12-05
25	10	1999-12-05

3. *People*



peopleID	fname	lname	phone	email	fineAmount
1	Isaac	Newton	778-456-7894	IssacN@gmail.com	0
2	Leonhard	Euler	778-852-7412	LeonhardEuler@sohu.com	0
3	Amedeo	Avogadro	778-876-1136	AmedeoA@gmail.com	0
4	Michael	Faraday	604-782-1143	Faraday@gmail.com	0
5	Johann Josef	Loschmidt	604-569-9984	None	0
6	Johann Jakob	Balmer	604-587-2244	JohannBalmer@sohu.com	0
7	Joseph	Stefan	778-896-7737	Joseph@gmail.com	0
8	Ludwig	Boltzmann	604-883-8871	LudwigBolt.mann@sfu.ca	15
9	Max	Planck	778-228-2236	Planck@sfu.ca	5
10	Wilhelm	Wien	778-998-7784	None	5
11	Edwin	Hubble	778-738-3633	None	50
12	Owen Willans	Richardson	778-997-4544	Richardson@gmail.com	0
13	Otto	Sackur	778-245-1577	Sackur@sohu.com	0
14	Niels	Bohr	778-354-1234	Bohr@sohu.com	0
15	Hugo	Tetrode	778-235-5665	Tetrode@gmail.com	0
16	Douglas	Hartree	778-887-3433	Joseph@sohu.com	0
17	Enrico	Fermi	778-822-7317	Fermi@gmail.com	0
18	Roger	Apery	778-123-4458	Apery@sfu.ca	10
19	Brian David	Josephson	778-123-4435	Josephson12354@we.ca	20
20	Klaus von	Klitzing	778-123-5578	Klitzing@ew2.net	10

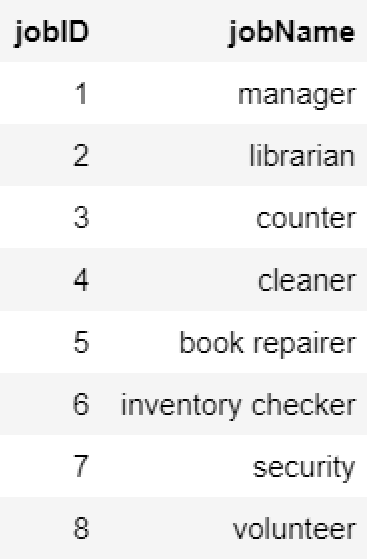
4. *Event*



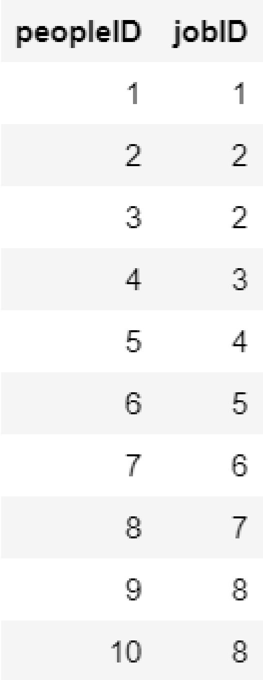
eventID	date	start_time	end_time	eventType
1	Monday	5.00pm	6.20pm	book club
2	Monday	6.30pm	7.50pm	book event
3	Tuesday	5.00pm	6.20pm	art show
4	Tuesday	6.30pm	7.50pm	book review sharing
5	Wednesday	6.30pm	9.30pm	local art show and selling
6	Thursday	6.30pm	9.30pm	art show
7	Friday	5.00pm	9.30pm	film screening
8	Saturday	9.30am	12.20pm	film screening
9	Saturday	12.30pm	3.20pm	speech
10	Saturday	3.30pm	6.20pm	book club
11	Sunday	9.30am	12.20pm	film screening
12	Sunday	12.30pm	3.20pm	book club
13	Sunday	3.30pm	6.20pm	speech

5. *JobList*

6. *Personnel*

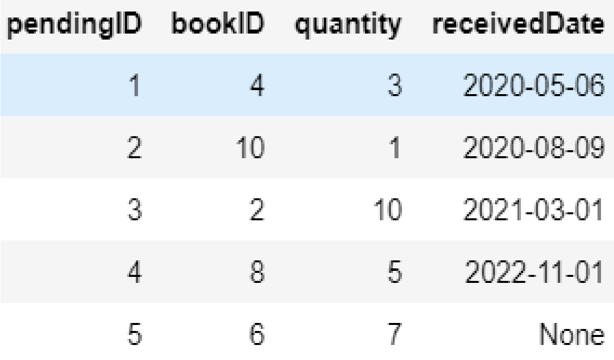


jobID	jobName
1	manager
2	librarian
3	counter
4	cleaner
5	book repairer
6	inventory checker
7	security
8	volunteer



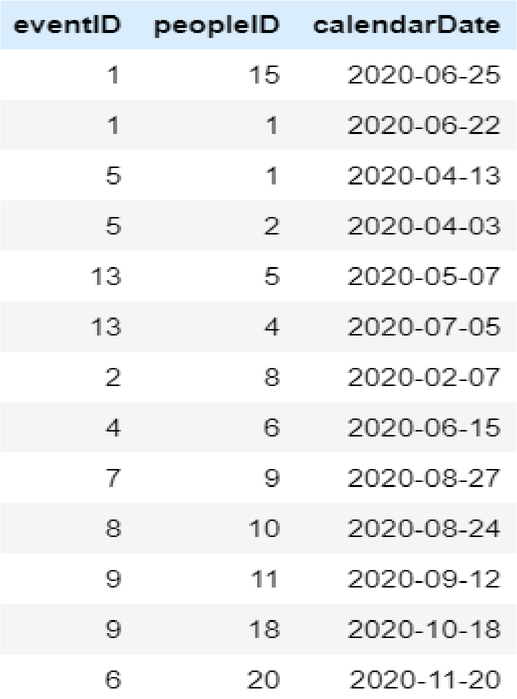
peopleID	jobID
1	1
2	2
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	8

7. ItemPending



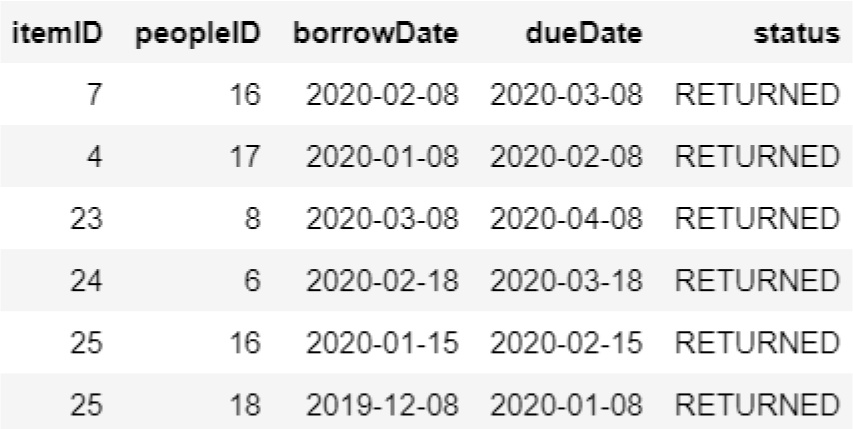
pendingID	bookID	quantity	receivedDate
1	4	3	2020-05-06
2	10	1	2020-08-09
3	2	10	2021-03-01
4	8	5	2022-11-01
5	6	7	None

8. Register



eventID	peopleID	calendarDate
1	15	2020-06-25
1	1	2020-06-22
5	1	2020-04-13
5	2	2020-04-03
13	5	2020-05-07
13	4	2020-07-05
2	8	2020-02-07
4	6	2020-06-15
7	9	2020-08-27
8	10	2020-08-24
9	11	2020-09-12
9	18	2020-10-18
6	20	2020-11-20

9. Borrow



itemID	peopleID	borrowDate	dueDate	status
7	16	2020-02-08	2020-03-08	RETURNED
4	17	2020-01-08	2020-02-08	RETURNED
23	8	2020-03-08	2020-04-08	RETURNED
24	6	2020-02-18	2020-03-18	RETURNED
25	16	2020-01-15	2020-02-15	RETURNED
25	18	2019-12-08	2020-01-08	RETURNED