

Упражнение 1.

Создайте таблицы с указанными столбцами и заполните их произвольными данными.

1) *Books (id, author, title, publish_year)*

```
CREATE TABLE books(  
  id INT,  
  author TEXT,  
  title TEXT,  
  publish_year INT  
);
```

Заполним созданную таблицу значениями:

```
INSERT INTO books  
VALUES  
  (1, 'Булгаков', 'Мастер и Маргарита', 1980),  
  (2, 'Достоевский', 'Преступление и наказание', 2000),  
  (3, 'Роулинг', 'Гарри Поттер и философский камень', 2011),  
  (4, 'Уальд', 'Портрет Дориана Грея', 2002),  
  (5, 'Носов', 'Приключения Незнайки', 2005),  
  (6, 'Толкин', 'Властелин колец', 2011),  
  (7, 'Толстой', 'Война и мир', 1999),  
  (8, 'Роулинг', 'Гарри Поттер и тайная комната', 2012),  
  (9, 'Роулинг', 'Гарри Поттер и узник Азкабана', 2012);
```

2) *Readers (id, name)*

```
CREATE TABLE readers(  
  id INT,  
  name TEXT  
);
```

Заполним созданную таблицу значениями:

```
INSERT INTO readers  
VALUES  
  (1, 'Сидоров'),  
  (2, 'Иванов'),  
  (3, 'Петров'),  
  (4, 'Кузнецов'),  
  (5, 'Колокольчиков');
```

3) *Records (reader_id, book_id, taking_date, returning_date)*

```
CREATE TABLE records(  
    reader_id INT,  
    book_id INT,  
    taking_date TEXT,  
    returning_date TEXT,  
    FOREIGN KEY (reader_id) REFERENCES readers (id),  
    FOREIGN KEY (book_id) REFERENCES books (id)  
);
```

Заполним созданную таблицу значениями:

```
INSERT INTO records  
VALUES  
    (5, 1, '2019-07-02', '2019-07-17'),  
    (2, 2, '2019-03-10', '2019-03-29'),  
    (3, 1, '2020-01-02', '2020-02-02'),  
    (1, 7, '2019-06-23', '2019-06-30'),  
    (5, 7, '2020-02-02', '2020-03-30'),  
    (2, 1, '2020-03-15', '2020-03-24');
```

Упражнение 2.

Постройте select запросы:

1) Запрос возвращает id и названия книг, находящихся в данный момент на руках у читателей.

```
SELECT DISTINCT book_id, title  
FROM records  
JOIN books ON records.book_id = books.id  
WHERE records.returning_date > date('now')
```

2) Запрос возвращает имена читателей и названия книг, которые они когда либо брали.

```
SELECT name, title  
FROM records  
JOIN readers ON records.reader_id = readers.id  
JOIN books ON records.book_id = books.id;
```

3) Запрос возвращает количество книг для каждого автора.

```
SELECT author, COUNT(title)  
FROM books  
GROUP BY author;
```

