



# Дополнительная лекция

Знакомство с базами данных





# Оглавление

[Приветствие](#)

[Структура базы данных библиотеки](#)

[Условия задачи](#)

[Что нового?](#)

[Связь по author\\_id](#)

[Книги с несколькими авторами](#)

[Недостающая таблица](#)

[Заключение](#)

[00:01:36]

## Приветствие

**И. Шафигуллин:** всем привет! Мы снова с вами — я и Александр. Как и обещали, разберём ещё один пример, чтобы вы посмотрели на процесс формирования структуры базы данных. Наша задача — сделать структуру базы данных для библиотеки.

[00:02:14]

## Структура базы данных библиотеки

**А. Сагун:** думаю, многие из вас были в библиотеках. Сейчас они не очень актуальны, но студенты обычно сталкиваются с библиотекой вуза или своего города.

Какая структура данных нужна в библиотеке?

- Перечень книг.
- Жанры книг.
- Перечень читателей.
- Данные о том, кто какие книги читал.

Пока мы не будем касаться хранения книг, потому что это довольно сложно. Но, если вам будет интересно, подумайте: как описать структуру данных так, чтобы быстро находить нужную книгу для читателя среди других?

**И. Шафигуллин:** какие есть варианты? Хранить всё по алфавиту? По названию или автору?

**А. Сагун:** да, непонятно. У автора может быть несколько книг, что усложнит структуру. Если хранить по названию, надо решить, какое мы возьмём за основу — оригинальное или переводное? Есть нюансы.

[00:04:52]

## Условия задачи

**А. Сагун:** итак, есть сотрудник, читатель, книга, авторы и жанры.

Мы сделали пример структуры. Он похож на то, что мы видели в примере с арендой автомобилей. Связи уже проставлены.

Книга				Читатель			
Параметр	Тип	Длина	Комментарий	Параметр	Тип	Длина	Комментарий
Book_ID	Int	10	Уникальный идентификатор книги	Reader_ID	Int	10	Уникальный идентификатор Читателя
Name	Text	100	Название книги	First_name	Text	30	Имя читателя
Author_id	Int	10	Автор	Family_name	Text	30	Фамилия
Create_date	TimeStamp	12	Дата создания записи	Middle_name	Text	30	Отчество
Public_date	Date	8	Дата публикации	Birth_date	Date	8	Дата рождения
Style_id	Int	10	Жанр	Passport_ser	Text	10	Серия паспорта
Type	Int	10	Тип	Passport_num	Text	20	Номер паспорта
Pages	Real	10	Кол-во страниц	Sex	Text	1	Пол
Reader_id	Int	10	Уникальный идентификатор Читателя				
Empl_id	Int	10	Уникальный идентификатор Сотрудника				
Жанр				Сотрудник			
Параметр	Тип	Длина	Комментарий	Параметр	Тип	Длина	Комментарий
Style_ID	Int	10	Уникальный идентификатор жанра	Empl_ID	Int	10	Уникальный идентификатор Сотрудника
Name	Text	100	Название жанра	First_name	Text	30	Имя читателя
Author_id	Int	10	Основоположенник	Family_name	Text	30	Фамилия
Create_date	TimeStamp	12	Дата создания записи	Middle_name	Text	30	Отчество
Автор				Birth_date	Date	8	Дата рождения
Параметр	Тип	Длина	Комментарий	Passport_ser	Text	10	Серия паспорта
Author_ID	Int	10	Уникальный идентификатор жанра	Passport_num	Text	20	Номер паспорта
First_name	Text	100	Название жанра	Position	Int	10	Должность
Family_name	Text	100	Основоположенник				
Birth_date	TimeStamp	12	Дата рождения				

Информация о книге, читателе и сотруднике пригодится, если книгу не вернут или вернут в потрепанном виде.

Вам нужно будет доработать структуру базы данных. Файл с таблицами прикреплен к лекции.

**И. Шафигуллин:** кто-то мог вернуть книгу с грязной обложкой или без страниц, и нужно понять, кто это сделал. С нынешним решением найти виновного сложно: придётся дублировать информацию про книги и делать много записей. На прошлом занятии мы выяснили, что это плохой путь.

**А. Сагун:** чтобы решить проблему, вам нужно будет сделать ещё одну таблицу (расположите её посередине листа) и настроить связи. В Excel они легко редактируются, можно их копировать и перетаскивать. Но этим мы займёмся чуть позже.

[00:07:46]

## Что нового?

**А. Сагун:** что мы добавили по сравнению с тем, что было в предыдущем примере? Появилась таблица «Сотрудник» с идентификатором. В примере с арендой было только поле с фамилией.

В описании книги есть название и автор. Для автора мы тоже сделали отдельную таблицу с уникальным идентификатором и набором атрибутов.

Есть дата создания записи — когда книга у нас появилась. Есть дата публикации — когда книга вышла в свет. Есть стиль (жанр). Для него мы сделали отдельную таблицу с идентификатором и атрибутами: названием, основоположником и датой, когда мы завели этот жанр в нашу БД.

Появилось два справочника. Но не хватает как минимум ещё одной связи между таблицами. Итак, первое задание: какой связи между какими двумя таблицами не хватает? По какому полю они должны быть связаны?

[00:11:58]

## Связь по author\_id

**А. Сагун:** думаю, вы заметили, что в таблице «Книги» есть author\_id, связанный с таблицей «Авторы». Но author\_id есть и в таблице «Жанры», а связь между ними не настроена. Её нужно добавить.

**И. Шафигуллин:** подумайте, какой тип связи используется между этими таблицами: один к одному, один ко многим или многие ко многим? Старайтесь по каждой таблице отвечать на этот вопрос.

[00:13:37]

## Книги с несколькими авторами

**А. Сагун:** в нашей базе данных заложено ограничение — у одной книги только один автор. При этом мы говорили, что у книги с несколькими авторами связь многие ко многим. Если нам нужно будет убрать ограничение по одному автору для книги, мы сделаем отдельную сущность, построим структуру, где будет идентификатор книги, идентификатор авторства, и таких записей может быть любое количество.

Например, есть книга «12 стульев». У неё два автора — Ильф и Петров. Соответственно, в таблице «Авторство» будет две записи:

- идентификатор книги (id 1) и идентификатор Ильфа (id 1),
- идентификатор книги (id 1) и идентификатор Петрова (id 2).

Получается, так мы для одной книги обозначим количество авторов.

Есть книги, которые пишутся целым коллективом авторов, например, научные труды. Там структура сложнее.

[00:15:27]

## Недостающая таблица

**А. Сагун:** теперь добавьте в структуру недостающую таблицу, добавьте поля с правильной типизацией и настройте правильные связи, чтобы мы могли решить кейс с испорченными книгами. То есть нашли виновного человека по фамилии, имени и паспорту.

Мы предлагаем такое решение: добавляем таблицу «Журнал читателей».

Параметр	Тип	Длина	Комментарий
Rec_ID	int	10	Уникальный идентификатор записи
Book_id	int	10	Идентификатор книги

Reader_id	int	10	Идентификатор читателя
Empl_id	int	10	Идентификатор сотрудника
Start_date	TimeStamp	12	Дата и время выдачи
End_date	TimeStamp	12	Дата и время сдачи
State	int	10	Состояние

Первые четыре записи — блок идентификаторов: записи в журнале, книги, читателя и сотрудника. По этим идентификаторам мы настроили соответствующие связи. Дальше идёт дата / время выдачи и возвращения, а также состояние книги.

Если одному читателю книга была выдана в хорошем состоянии, а следующий говорит, что она разрисована, нам нужно будет найти предыдущего читателя и узнать у сотрудника, как он принял испорченную книгу обратно. Он идёт в базу, ищет предыдущего читателя и может с ним связаться.

**И. Шафигуллин:** для полей Start\_date и End\_date мы сделали тип данных TimeStamp. Почему?

**А. Сагун:** иногда в библиотеках берут книги не на несколько дней, а на несколько часов для работы в читальном зале. Нужно учитывать выдачу книг в рамках одного дня. Если есть задача фиксировать событие в рамках дня, надо добавлять поле с типом данных TimeStamp, чтобы понимать, в какой день и в какое время произошло событие.

Если вы понимаете, что нужно отслеживать только день, достаточно типа Date. Например, нам не нужно отслеживать время первой публикации книги, достаточно только даты.

**[00:29:39]**

## Заключение

**И. Шафигуллин:** надеюсь, наше решение вам понятно. У вас остаётся задача на вырост — сделать структуру хранения книг. А на этом наш курс заканчивается. Надеюсь, за три месяца вы узнали много нового. Будем рады видеть вас на следующих курсах. До свидания.