

## Раздел 1: Модель, генерация и наполнение базы данных

### 1.1. Общее описание модели

Для решения задач и рассмотрения примеров мы будем использовать три базы данных: «Библиотека», «Большая библиотека» и «Исследование». База данных «Исследование» будет состоять из множества разрозненных таблиц, необходимых для демонстрации особенностей поведения СУБД, и мы будем формировать её постепенно по мере проведения экспериментов. Также в ней будет физически расположена база данных «Большая библиотека».

Модели баз данных «Библиотека» и «Большая библиотека» полностью идентичны (отличаются эти базы данных только количеством записей). Здесь всего семь таблиц:

- **genres** — описывает литературные жанры:
  - **g\_id** — идентификатор жанра (число, первичный ключ);
  - **g\_name** — имя жанра (строка);
- **books** — описывает книги в библиотеке:
  - **b\_id** — идентификатор книги (число, первичный ключ);
  - **b\_name** — название книги (строка);
  - **b\_year** — год издания (число);
  - **b\_quantity** — количество экземпляров книги в библиотеке (число);
- **authors** — описывает авторов книг:
  - **a\_id** — идентификатор автора (число, первичный ключ);
  - **a\_name** — имя автора (строка);
- **subscribers** — описывает читателей (подписчиков) библиотеки:
  - **s\_id** — идентификатор читателя (число, первичный ключ);
  - **s\_name** — имя читателя (строка);
- **subscriptions** — описывает факты выдачи/возврата книг (т.н. «подписки»):
  - **sb\_id** — идентификатор подписки (число, первичный ключ);
  - **sb\_subscriber** — идентификатор читателя (подписчика) (число, внешний ключ);
  - **sb\_book** — идентификатор книги (число, внешний ключ);
  - **sb\_start** — дата выдачи книги (дата);
  - **sb\_finish** — запланированная дата возврата книги (дата);
  - **sb\_is\_active** — признак активности подписки (содержит значение **Y**, если книга ещё на руках у читателя, и **N**, если книга уже возвращена в библиотеку);
- **m2m\_books\_genres** — служебная таблица для организации связи «многие ко многим» между таблицами **books** и **genres**:
  - **b\_id** — идентификатор книги (число, внешний ключ, часть составного первичного ключа);
  - **g\_id** — идентификатор жанра (число, внешний ключ, часть составного первичного ключа);
- **m2m\_books\_authors** — служебная таблица для организации связи «многие ко многим» между таблицами **books** и **authors**:
  - **b\_id** — идентификатор книги (число, внешний ключ, часть составного первичного ключа);
  - **a\_id** — идентификатор автора (число, внешний ключ, часть составного первичного ключа).

## 1.2. Модель для MySQL

Модель базы данных для MySQL представлена на рисунках 1.b и 1.c. Обратите внимание на следующие важные моменты:

- Первичные ключи представлены беззнаковыми целыми числами для расширения максимального диапазона значений.
- Строки представлены типом **varchar** длиной до 150 символов (чтобы гарантированно уложиться в ограничение 767 байт на длину индекса;  $150 \cdot 4 = 600$ , MySQL выравнивает символы UTF-строк на длину в четыре байта на символ при операциях сравнения).
- Поле **sb\_is\_active** представлено характерным для MySQL типом данных **enum** (позволяющим выбрать одно из указанных значений и очень удобным для хранения заранее известного predetermined набора значений — **Y** и **N** в нашем случае).
- Поле **g\_name** сделано уникальным, т.к. существование одноимённых жанров недопустимо.
- Поля **sb\_start** и **sb\_finish** представлены типом **date** (а не более полными, например, **datetime**), т.к. мы храним дату выдачи и возврата книги с точностью до дня.

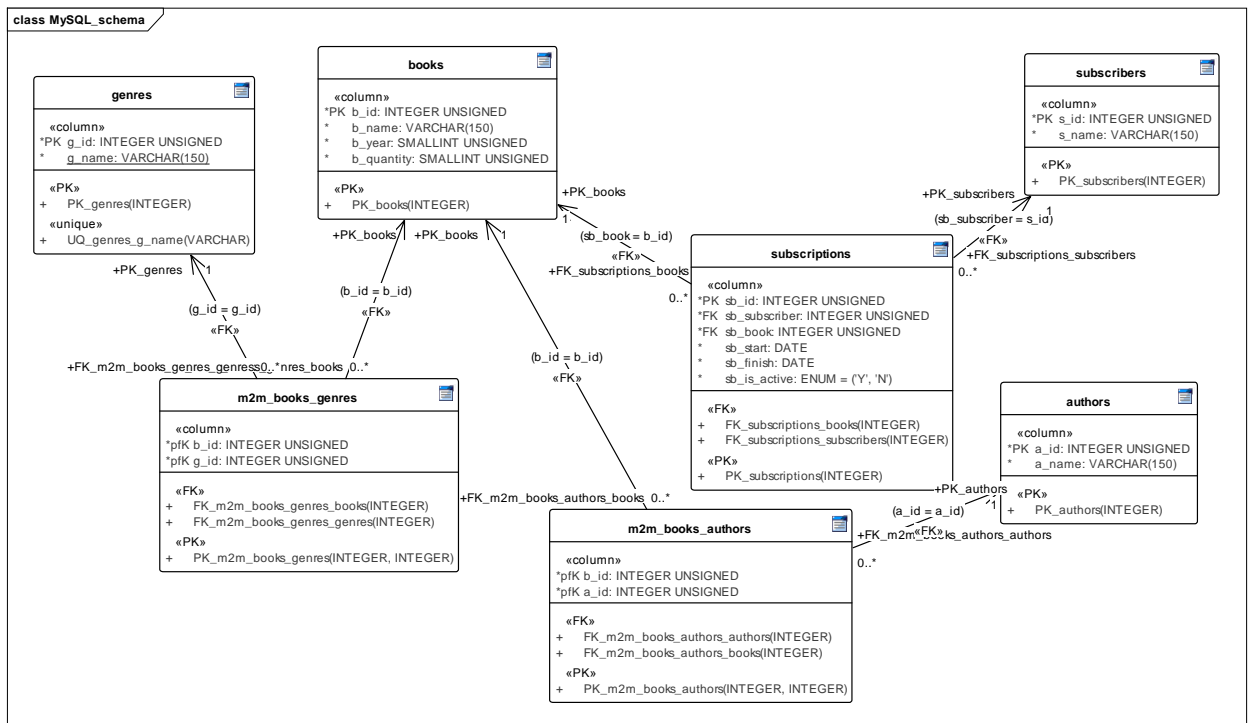


Рисунок 1.b — Модель базы данных для MySQL в Sparx EA

### 1.3. Модель для MS SQL Server

Модель базы данных для MS SQL Server представлена на рисунках 1.d и 1.e. Обратите внимание на следующие важные моменты:

- Первичные ключи представлены знаковыми целыми числами, т.к. в MS SQL Server нет возможности сделать **bigint**, **int** и **smallint** беззнаковыми (а **tinyint**, наоборот, бывает только беззнаковым).
- Строки представлены типом **nvarchar** длиной до 150 символов (как из соображений аналогии с MySQL, так и чтобы гарантированно уложиться в ограничение 900 байт на длину индекса;  $150 \cdot 2 = 300$ , MS SQL Server для хранения и сравнения символов в национальных кодировках использует два байта на символ).
- Поле **sb\_is\_active** представлено типом **char** длиной в один символ (т.к. в MS SQL Server нет типа данных **enum**), а для соблюдения аналогии с MySQL на это поле наложено ограничение **check** со значением **[sb\_is\_active] IN ('Y', 'N')**.
- Как и в MySQL, поля **sb\_start** и **sb\_finish** представлены типом **date** (а не более полными, например, **datetime**), т.к. мы храним дату выдачи и возврата книги с точностью до дня.

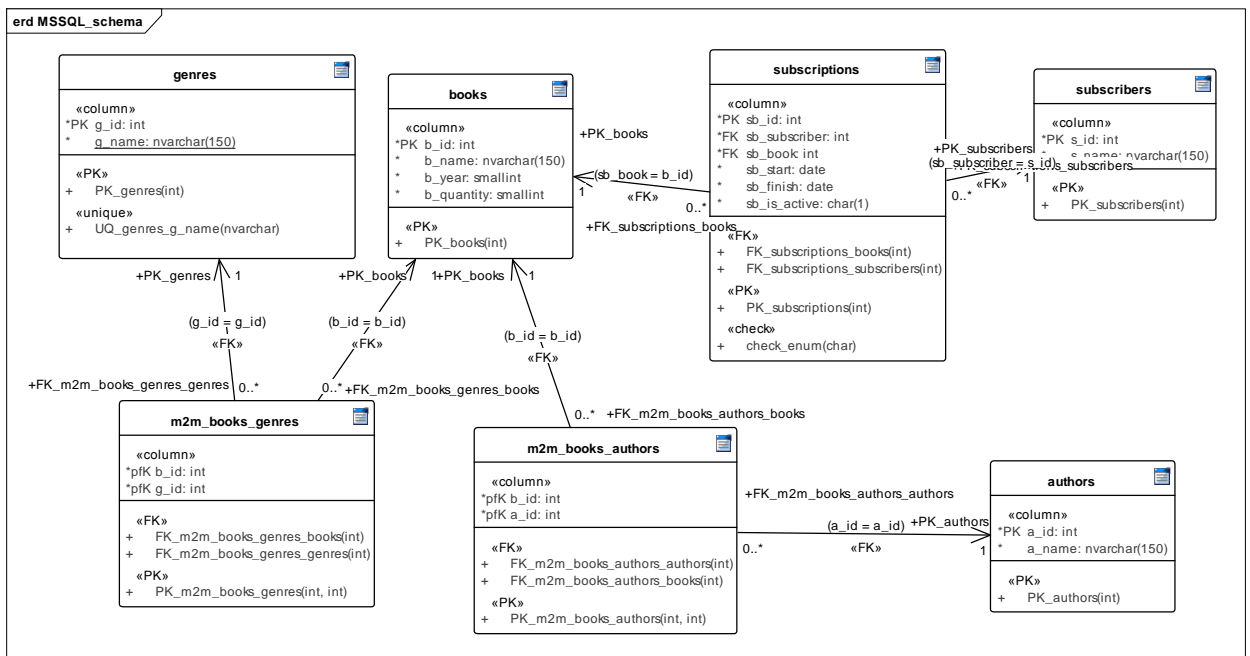


Рисунок 1.d — Модель базы данных для MS SQL Server в Sparx EA

## 1.4. Модель для Oracle

Модель базы данных для Oracle представлена на рисунках 1.f и 1.g. Обратите внимание на следующие важные моменты:

- Целочисленные поля представлены типом **number** с указанием длины — это наиболее простой способ эмуляции целочисленных типов из других СУБД.
- Строки представлены типом **nvarchar2** длиной до 150 символов (как из соображений аналогии с MySQL и MS SQL Server, так и чтобы гарантированно уложиться в ограничение 758-6498 байт на длину индекса;  $150 \cdot 2 = 300$ , Oracle для хранения и сравнения символов в национальных кодировках использует два байта на символ, и максимальная длина индекса может зависеть от разных условий, но минимум — 758 байт).
- Поле **sb\_is\_active** представлено типом **char** длиной в один символ (т.к. в Oracle нет типа данных **enum**), а для соблюдения аналогии с MySQL применено то же решение, что и в случае с MS SQL Server: на это поле наложено ограничение **check** со значением "**sb\_is\_active**" IN ('Y', 'N').
- Для полей **sb\_start** и **sb\_finish** выбран тип **date** как «самый простой» из имеющихся в Oracle типов хранения даты. Да, он всё равно сохраняет часы, минуты и секунды, но мы можем вписать туда нулевые значения.

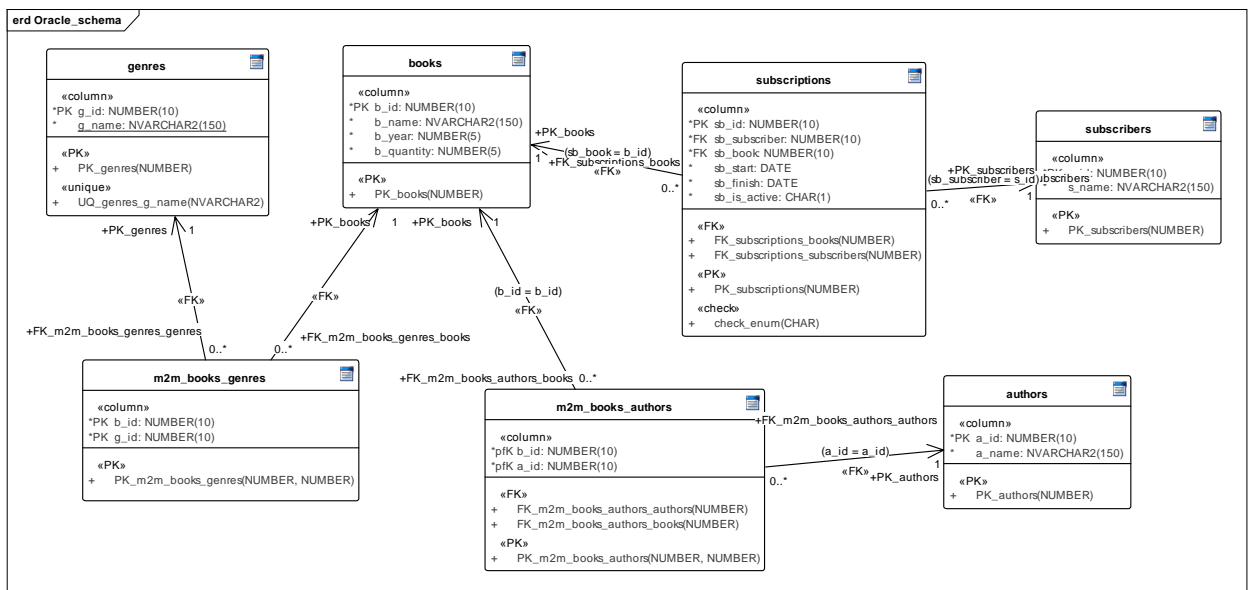


Рисунок 1.f — Модель базы данных для Oracle в Sparx EA

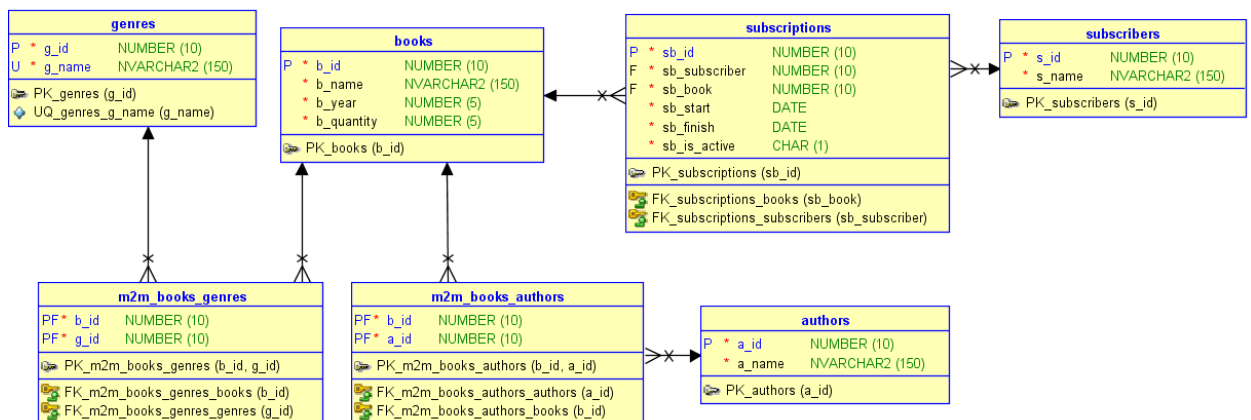


Рисунок 1.g — Модель базы данных для Oracle в Oracle SQL Developer Data Modeler

## 1.5. Генерация и наполнение базы данных

На основе созданных в Sparx Enterprise Architect моделей получим DDL-скрипты<sup>(4)</sup> для каждой СУБД и выполним их, чтобы создать базы данных (обратитесь к документации по Sparx EA за информацией о том, как на основе модели базы данных получить скрипт её генерации).

Наполним имеющиеся базы данных следующими данными.

Таблица **books**:

b_id	b_name	b_year	b_quantity
1	Евгений Онегин	1985	2
2	Сказка о рыбаке и рыбке	1990	3
3	Основание и империя	2000	5
4	Психология программирования	1998	1
5	Язык программирования C++	1996	3
6	Курс теоретической физики	1981	12
7	Искусство программирования	1993	7

Таблица **authors**:

a_id	a_name
1	Д. Кнут
2	А. Азимов
3	Д. Карнеги
4	Л.Д. Ландау
5	Е.М. Лифшиц
6	Б. Страуструп
7	А.С. Пушкин

Таблица **genres**:

g_id	g_name
1	Поэзия
2	Программирование
3	Психология
4	Наука
5	Классика
6	Фантастика

Таблица **subscribers**:

s_id	s_name
1	Иванов И.И.
2	Петров П.П.
3	Сидоров С.С.
4	Сидоров С.С.

Присутствие двух читателей с именем «Сидоров С.С.» — не ошибка. Нам понадобится такой вариант полных тёзок для демонстрации нескольких типичных ошибок в запросах.

Таблица **m2m\_books\_authors**:

<b>b_id</b>	<b>a_id</b>
1	7
2	7
3	2
4	3
4	6
5	6
6	5
6	4
7	1

Таблица **m2m\_books\_genres**:

<b>b_id</b>	<b>g_id</b>
1	1
1	5
2	1
2	5
3	6
4	2
4	3
5	2
6	5
7	2
7	5

Таблица **subscriptions**:

<b>sb_id</b>	<b>sb_subscriber</b>	<b>sb_book</b>	<b>sb_start</b>	<b>sb_finish</b>	<b>sb_is_active</b>
100	1	3	2011-01-12	2011-02-12	N
2	1	1	2011-01-12	2011-02-12	N
3	3	3	2012-05-17	2012-07-17	Y
42	1	2	2012-06-11	2012-08-11	N
57	4	5	2012-06-11	2012-08-11	N
61	1	7	2014-08-03	2014-10-03	N
62	3	5	2014-08-03	2014-10-03	Y
86	3	1	2014-08-03	2014-09-03	Y
91	4	1	2015-10-07	2015-03-07	Y
95	1	4	2015-10-07	2015-11-07	N
99	4	4	2015-10-08	2025-11-08	Y

Странные комбинации дат выдачи и возврата книг (2015-10-07 / 2015-03-07 и 2015-10-08 / 2025-11-08) добавлены специально, чтобы продемонстрировать в дальнейшем особенности решения некоторых задач.

Также обратите внимание, что среди читателей, бравших книги, ни разу не встретился «Петров П.П.» (с идентификатором 2), это тоже понадобится нам в будущем.

Неупорядоченность записей по идентификаторам и «пропуски» в нумерации идентификаторов также сделаны осознанно для большей реалистичности.