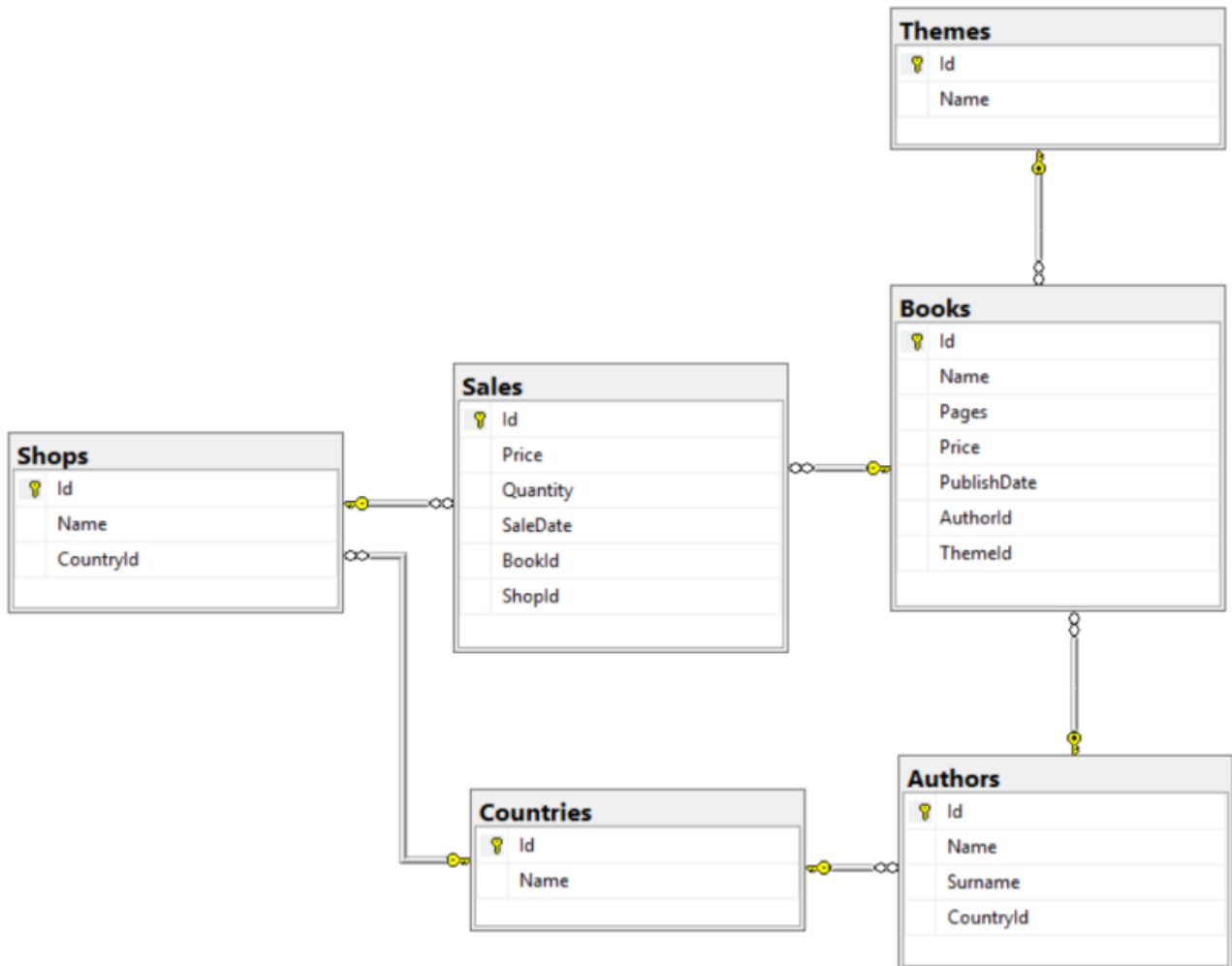




### «Книжный магазин»

## Структура базы данных



База данных Книжный магазин (**BookShop**) содержит информацию о товаре и сделках сети книжных магазинов. Книги, продаваемые в магазинах представлены в виде таблицы Книги (**Books**), в которой собрана основная информация, такая как: название, количество страниц, цена и дата публикации. Также для каждой книги хранится ее автор и тематика. При этом список всех авторов представлен таблицей Авторы (**Authors**), а все доступные тематики содержатся в таблице Тематики (**Themes**). Для хранения всех проведенных в магазинах сделок (продаж книг) используется таблица Продажи (**Sales**), в которой описано что, где, когда, кем и за сколько было продано. Сеть книжных магазинов представлена несколькими филиалами, которые расположены в разных странах. Для хранения данной информации присутствуют таблицы Магазины (**Shops**) и Страны (**Countries**).

## 1. Авторы (Authors)

- Идентификатор (**Id**). Уникальный идентификатор автора.

- ▷ Тип данных — **int**.
- ▷ Авто приращение.
- ▷ Не может содержать **null**-значения.
- ▷ Первичный ключ.

- Имя (**Name**). Имя автора.

- ▷ Тип данных — **nvarchar(max)**.
- ▷ Не может содержать **null**-значения.
- ▷ Не может быть пустым.

- Фамилия (**Surname**). Фамилия автора.

- ▷ Тип данных — **nvarchar(max)**.
- ▷ Не может содержать **null**-значения.
- ▷ Не может быть пустым.

- Идентификатор страны (**CountryId**). Страна, из которой родом автор.

- ▷ Тип данных — **int**.
- ▷ Не может содержать **null**-значения.
- ▷ Внешний ключ.

## 2. Книги (Books)

- Идентификатор (**Id**). Уникальный идентификатор книги.

- ▷ Тип данных — **int**.
- ▷ Авто приращение.
- ▷ Не может содержать **null**-значения.
- ▷ Первичный ключ.

■ Название (**Name**). Название книги.

▷ Тип данных — **nvarchar(max)**.

▷ Не может содержать **null**-значения.

▷ Не может быть пустым.

■ Страницы (**Pages**). Количество страниц в книге.

▷ Тип данных — **int**.

▷ Не может содержать **null**-значения.

▷ Не может быть меньше либо равно 0.

■ Цена (**Price**). Цена книги.

▷ Тип данных — **money**.

▷ Не может содержать **null**-значения.

▷ Не может быть меньше 0.

■ Дата публикации (**PublishDate**). Дата публикации книги.

▷ Тип данных — **date**.

▷ Не может содержать **null**-значения.

▷ Не может быть больше текущей даты.

■ Идентификатор автора (**AuthorId**). Автор книги.

▷ Тип данных — **int**.

▷ Не может содержать **null**-значения.

▷ Внешний ключ.

■ Идентификатор тематики (**ThemeId**). Тематика книги.

▷ Тип данных — **int**.

▷ Не может содержать **null**-значения.

▷ Внешний ключ.

## 3. Страны (Countries)

- Идентификатор (**Id**). Уникальный идентификатор страны.

- ▷ Тип данных — **int**.
- ▷ Авто приращение.
- ▷ Не может содержать **null**-значения.
- ▷ Первичный ключ.

- Название (**Name**). Название страны.

- ▷ Тип данных — **nvarchar(50)**.
- ▷ Не может содержать **null**-значения.
- ▷ Не может быть пустым.
- ▷ Должно быть уникальным.

## 4. Продажи (Sales)

- Идентификатор (**Id**). Уникальный идентификатор продажи.

- ▷ Тип данных — **int**.
- ▷ Авто приращение.
- ▷ Не может содержать **null**-значения.
- ▷ Первичный ключ.

- Цена (**Price**). Цена продажи одного экземпляра книги.

- ▷ Тип данных — **money**.
- ▷ Не может содержать **null**-значения.
- ▷ Не может быть меньше 0.

- Количество (**Quantity**). Количество проданных экземпляров книги.

- ▷ Тип данных — **int**.
- ▷ Не может содержать **null**-значения.
- ▷ Не может быть меньше либо равно 0.

- Дата продажи (**SaleDate**). Дата продажи.

- ▷ Тип данных — **date**.

- ▷ Не может содержать **null**-значения.

- ▷ Не может быть больше текущей даты.

- ▷ Значение по умолчанию — текущая дата.

- Идентификатор книги (**BookId**). Проданная книга.

- ▷ Тип данных — **int**.

- ▷ Не может содержать **null**-значения.

- ▷ Внешний ключ.

- Идентификатор магазина (**ShopId**). Магазин, в котором была совершена продажа.

- ▷ Тип данных — **int**.

- ▷ Не может содержать **null**-значения.

- ▷ Внешний ключ.

## 5. Магазины (Shops)

- Идентификатор (**Id**). Уникальный идентификатор магазина.

- ▷ Тип данных — **int**.

- ▷ Авто приращение.

- ▷ Не может содержать **null**-значения.

- ▷ Первичный ключ.

- Название (**Name**). Название магазина.

- ▷ Тип данных — **nvarchar(max)**.

- ▷ Не может содержать **null**-значения.

- ▷ Не может быть пустым.

- Идентификатор страны (**CountryId**). Страна, в которой находится магазин.

- ▷ Тип данных — **int**.

- ▷ Не может содержать **null**-значения.

- ▷ Внешний ключ.

## 6. Тематики (Themes)

- Идентификатор (**Id**). Уникальный идентификатор тематики.

- ▷ Тип данных — **int**.

- ▷ Авто приращение.

- ▷ Не может содержать **null**-значения.

- ▷ Первичный ключ.

- Название (**Name**). Название тематики.

- ▷ Тип данных — **nvarchar(100)**.

- ▷ Не может содержать **null**-значения.

- ▷ Не может быть пустым.

- ▷ Должно быть уникальны

## Запросы (данные вариативны)

1. Показать все книги, количество страниц в которых больше 500, но меньше 650.
2. Показать все книги, в которых первая буква названия либо «А», либо «З».
3. Показать все книги жанра «Детектив», количество проданных книг более 30 экземпляров.
4. Показать все книги, в названии которых есть слово «Microsoft», но нет слова «Windows».
5. Показать все книги (название, тематика, полное имя автора в одной ячейке), цена одной страницы которых меньше 65 копеек.
6. Показать все книги, название которых состоит из 4 слов.
7. Показать информацию о продажах в следующем виде:
  - ▷ Название книги, но, чтобы оно не содержало букву «А».
  - ▷ Тематика, но, чтобы не «Программирование».
  - ▷ Автор, но, чтобы не «Герберт Шилдт».
  - ▷ Цена, но, чтобы в диапазоне от 10 до 20 гривен.
  - ▷ Количество продаж, но не менее 8 книг.
  - ▷ Название магазина, который продал книгу, но он не должен быть в Украине или России.
8. Показать следующую информацию в два столбца (числа в правом столбце приведены в качестве примера):
  - ▷ Количество авторов: 14
  - ▷ Количество книг: 47
  - ▷ Средняя цена продажи: 85.43 грн.
  - ▷ Среднее количество страниц: 650.6.
9. Показать тематики книг и сумму страниц всех книг по каждой из них.
10. Показать количество всех книг и сумму страниц этих книг по каждому из авторов.
11. Показать книгу тематики «Программирование» с наибольшим количеством страниц.
12. Показать среднее количество страниц по каждой тематике, которое не превышает 400.
13. Показать сумму страниц по каждой тематике, учитывая только книги с количеством страниц более 400, и чтобы тематики были «Программирование», «Администрирование» и «Дизайн».
14. Показать информацию о работе магазинов: что, где, кем, когда и в каком количестве было продано.
15. Показать самый прибыльный магазин.

## Туристическое агенство «Вокруг света за 80 дней»

**Иноформация (необходимо продумать состав таблиц и связи, возможно добавить дополнительную инфомацию)**

### ■ Сотрудники турагентства

- ▷ ФИО
- ▷ дата рождения
- ▷ телефон
- ▷ дата приёма на работу
- ▷ должность (дополнительная таблица с должностями)
- ▷ оклад
- ▷ премия

### ■ Сотрудники, ответственные за туры (промежуточная таблица между сотрудниками и турами)

### ■ Туры

- ▷ название тура
- ▷ стоимость
- ▷ дата старта
- ▷ дата окончания
- ▷ транспорт (дополнительная таблица с видами транспорта)
- ▷ страна (дополнительная таблица со странами)
- ▷ город (дополнительная таблица с городами)
- ▷ достопримечательности, входящая в стоимость тура (дополнительная таблица с достопримечательностями)
- ▷ гостиница (дополнительная таблица с гостиницами)
- ▷ поле для связи с промежуточной таблицей между турами и туристами
- ▷ поле для связи с промежуточной таблицей между турами и сотрудниками



■ Туристы (промежуточная таблица между клиентами и турами)

■ Клиенты агентства

▷ ФИО

▷ телефон

▷ дата рождения

▷ поле для связи клиента с туром (промежуточная таблица между турами и клиентами)

## Запросы (данные вариативны)

- 1 Отобразить сотрудника, который работает дольше всего
- 2 Отобразить сотрудника, у которого максимальные оклад+премия
- 3 Отобразить самого активного сотрудника (по количеству туристических поездок)
- 4 Отобразить самый прибыльный тур (максимальная стоимость и количество туристов)
- 5 Показать самый непопулярный тур (по минимальному количеству туристов)
- 6 Отобразите информацию о турах, которые стартуют в диапазоне дат
- 7 Отобразить информацию о всех турах указанного способа передвижения
- 8 Отобразите информацию о турах, которые посетят указанную страну
- 9 Отобразите самую популярную туристическую страну (по самому большому количеству туров)
- 10 Отобразить самую популярную достопримечательность (наличие в популярном туре)
- 11 Отобразить информацию о самой популярной гостинице среди туристов (по количеству туристов)
- 12 Показать для конкретного туриста по ФИО список всех его туров
- 13 Отобразить информацию о самом активном туристе (по количеству туров)

# SQL



## Введение в SQL

Создать БД	<code>CREATE DATABASE teststep;</code>
Изменить наименование БД	<code>ALTER DATABASE teststep MODIFY NAME = teststep2;</code>
Удалить БД	<code>DROP DATABASE teststep;</code>
Работать с указанной БД	<code>USE teststep;</code>
Создать таблицу	<code>USE teststep CREATE TABLE students (-- PRIMARY KEY = UNIQUE + NOT NULL id INT NOT NULL IDENTITY(1, 1) PRIMARY KEY, fio NVARCHAR(MAX) NOT NULL CHECK (fio &lt;&gt; N''), groupNumber NVARCHAR(10) UNIQUE, grant MONEY NOT NULL CHECK (grant &gt;= 0.0) DEFAULT (0.0), BirthDate DATE NOT NULL CHECK (BirthDate &lt;= getdate()), StartTime TIME NOT NULL CHECK (StartTime BETWEEN '08:00' AND '18:00') );</code>
Связать таблицы	<code>ALTER TABLE students ADD FOREIGN KEY (GroupId REFERENCES Groups(Id);</code>
Изменить наименование таблицы	<code>USE teststep EXEC sp_rename 'students', 'Students';</code>
Удалить таблицу	<code>USE teststep DROP TABLE students;</code>
Добавить поле в таблицу	<code>USE teststep ALTER TABLE students ADD letter NVARCHAR(5);</code>
Изменить тип поля в таблице	<code>USE teststep ALTER TABLE students ALTER COLUMN letter NVARCHAR(10);</code>
Удалить поле из таблицы	<code>USE teststep ALTER TABLE students DROP COLUMN letter;</code>
Вставить строку с информацией в таблицу	<code>USE teststep INSERT INTO students (id, fio, groupNumber) VALUES (1, N'Иванов Иван Иванович', N'ЕКО- 11021'), (2, N'Петров Петр Петрович', N'ЕКО- 11022'),</code>

# SQL

	(3, N'Сидоров Сидор Сидорович', N'ЕКО-11021');
Выборка данных из таблицы	USE teststep SELECT * FROM students;
Выборка данных из таблицы с псевдонимами	USE teststep SELECT id AS ИД, fio AS ФИО FROM students;
Выборка уникальных данных из таблицы	USE teststep SELECT DISTINCT * FROM students;
Выборка ограниченного количества записей (строк) из таблицы	USE teststep SELECT TOP 2 * FROM students;
Выборка данных из таблицы по условию	USE teststep SELECT * FROM students WHERE id = 1;
Выборка данных из таблицы по составному условию	USE teststep SELECT * FROM students WHERE id > 1 AND id < 3; -- AND OR NOT = != > < <= >=
Выборка данных из таблицы по условию вхождения в множество	USE teststep SELECT * FROM students WHERE id IN (1); -- IN NOT IN
Выборка данных из таблицы по условию вхождения в диапазон	USE teststep SELECT * FROM students WHERE id BETWEEN 1 AND 3;
Выборка данных из таблицы по условию вхождения в фильтр	USE teststep SELECT * FROM students WHERE fio LIKE N'%Иван%'; -- % _
Выборка данных из таблицы по условию вхождения в фильтр регулярного выражения	USE teststep SELECT * FROM students WHERE id LIKE '[1-2]'; -- [abc] [!abc] [начало-конец]
Выборка данных из таблицы с сортировкой	USE teststep SELECT * FROM students ORDER BY id DESC; -- ASC DESC

# SQL

Выборка данных из таблицы с использованием агрегатных функций	<pre>USE teststep SELECT AVG(id) FROM students; -- AVG COUNT SUM MIN MAX</pre>
Выборка данных из таблицы с использованием агрегатных функций для работы с числами	<pre>USE teststep SELECT POWER(id, 2), fio FROM students; -- ROUND FLOOR TRUNCATE POWER SQRT RAND</pre>
Выборка данных из таблицы с использованием агрегатных функций для работы со строками	<pre>USE teststep SELECT SUBSTRING(fio, 1, 7) FROM students; -- CONCAT LENGTH TRIM SUBSTRING REPLACE LOWER UPPER</pre>
Выборка данных из таблицы с использованием агрегатных функций (обязательно) и группировки	<pre>USE teststep SELECT COUNT(id) AS количество, fio FROM students GROUP BY fio;</pre>
Выборка данных из таблицы с использованием агрегатных функций (обязательно) и группировки с условием	<pre>USE teststep SELECT COUNT(id) AS количество, fio FROM students GROUP BY fio HAVING COUNT(id) &lt; 2;</pre>
Обновить (редактировать) данные поля в таблице по условию	<pre>USE teststep UPDATE students SET groupNumber = 'EKO-11023' WHERE id = 3;</pre>
Удалить строку в таблице по условию	<pre>USE teststep DELETE FROM students WHERE id = 3;</pre>
Очистить данные таблицы	<pre>USE teststep TRUNCATE TABLE students;</pre>
Подзапросы	<pre>USE Hospital SELECT Name FROM Wards WHERE Places = (SELECT MIN(Places) FROM Wards);</pre>
Связать данные при помощи идентификаторов в WHERE	<pre>use Hospital select D.Surname, D.Salary+D.Premium as Salary from Doctors as D, Vacations as V where D.Id = V.DoctorId and MONTH(V.StartDate)!=10;</pre>
Связать данные при помощи UNION	<pre>SELECT FirstName + ' ' + LastName AS FullName, BirthDate FROM Students WHERE MONTH(BirthDate) &gt; 5 AND MONTH(BirthDate) &lt; 9</pre>

# SQL

	<pre>UNION SELECT FirstName + ' ' + LastName, BirthDate FROM Teachers WHERE MONTH(BirthDate) &gt; 5 AND MONTH(BirthDate) &lt; 9 ORDER BY BirthDate;</pre>
Связать данные при помощи INNER JOIN ON	<pre>SELECT FirstName, LastName, Name AS Subject, Assesment, GroupName FROM Groups AS G JOIN Students AS S ON G.Id = S.GroupId JOIN Achievements AS A ON S.Id = A.StudentId JOIN Subjects AS Sb ON Sb.Id = A.SubjectId WHERE GroupName LIKE '29%' ORDER BY LastName;</pre>
Связать данные при помощи LEFT OUTER JOIN ON	<pre>SELECT FirstName + ' ' + LastName AS FullName FROM Students AS S LEFT JOIN Achievements AS A ON S.Id = A.StudentId WHERE Assesment IS NULL;</pre>
Связать данные при помощи RIGHT OUTER JOIN ON	<pre>SELECT FirstName + ' ' + LastName AS FullName, Assesment FROM Achievements AS A RIGHT JOIN Students AS S ON S.Id = A.StudentId;</pre>
Связать данные при помощи FULL OUTER JOIN ON	<pre>SELECT FirstName, LastName, GroupName FROM Students AS S FULL JOIN Groups AS G ON G.Id = S.GroupId ORDER BY FirstName;</pre>
Связать данные при помощи комбинирования JOIN ON	<pre>SELECT [Name] AS [Subject], LastName, FirstName FROM TeachersSubjects AS TS RIGHT JOIN Subjects AS S ON S.Id = TS.SubjectId LEFT JOIN Teachers AS T ON T.Id = TS.TeacherId ORDER BY [Name];</pre>