Темы урока

[**Базы Данных и Системы Управления Базами Данных**](#_o7g5uzxgtryh) **2**

[Термины “БД” и “СУБД”](#_mqis9wv4n6en) 2

[База Данных (БД)](#_2svv7020bp76) 2

[Система Управления Базами Данных (СУБД)](#_bdbqoicbzs9h) 2

[Типы БД](#_148jex4idhpk) 2

[Классификация по месту постоянного хранения данных](#_fnb5q1ag7pjd) 2

[Классификация по модели данных](#_crkxgygfrgy4) 3

[Классификация по степени распределённости](#_epw973ieezvh) 3

[Наиболее популярные СУБД и их SQL](#_grb4exgi14pe) 3

[SQL — структурированный язык запросов](#_mo0d392cgqii) 4

[Зачем изучать SQL если есть ORM?](#_bbpes7px961b) 4

[Реляционная база данных](#_5enwicggawdo) 4

[СУБД SQL Server 2017 Express Edition](#_5e7mb564ja6d) 5

[Преимущества](#_v077pnz58eet) 5

[Установка инструментов](#_6bnpy38lcfss) 5

[MS SQL Server Express](#_figlll2o7i2) 5

[Microsoft SQL Management Studio](#_9oro9h27gfsk) 5

[Термины](#_w39cai29lils) 5

[**Работа в с базой данных**](#_l41gxiagmwpy) **6**

[Работа в Microsoft Visual Studio](#_orivvrx6t9jq) 6

[Работа в Microsoft SQL Management Studio](#_a5m3c7kpe6kl) 6

[Интерфейс](#_mwgngj5jldk7) 6

[Создание базы данных через UI](#_oluq9ew6pwob) 6

[Создание базы данных с помощью SQL-команд:](#_u8g3r4l6f24y) 6

[Создание новой таблицы](#_ktpm81pwvbuz) 7

[Заведение значений через UI](#_4xrevvgtbw9o) 7

[Генерация запросов](#_om2f1asu83jq) 7

[Создание таблицы](#_r115rylb01s9) 7

[Вставка единичной записи](#_sr6gtxir4quu) 7

[Выборка записей из таблиц](#_i2b4i66odjaf) 7

[Типы данных](#_e5yby8s0lcqn) 7

[Самостоятельная работа](#_wdt04hfv8ji4) 8

[**Домашнее задание**](#_81x86a71ny08) **8**

# 

# Базы Данных и Системы Управления Базами Данных

В самом начале разговора о БД важно провести черту между двумя терминами, часто использующимися рядом:

* База Данных
* Система Управления Базами Данных

Иногда их путают. Иногда, когда говорящие в одном контексте, это выглядит нормально. Иногда это выглядит как в классической ситуации, когда системный блок называют “Процессор”, поэтому давайте разберёмся, что есть что?

## Термины “БД” и “СУБД”

### База Данных (БД)

В википедии можно прочитать универсальное [определение термина База Данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85).

Если его немного упростить, то получится, что БД — это совокупность информационных материалов, систематизированных для удобного поиска и обработки с помощью компьютера.

Если ещё упростить, **БД — это структурно хранящаяся информация / структурно хранящиеся данные**.

### Система Управления Базами Данных (СУБД)

**СУБД — программный комплекс, обеспечивающий управление созданием и использованием баз данных.**

## Типы БД

Есть много критериев, по которым можно классифицировать базы данных, однако, наиболее интересны в практическом смысле 3 критерия:

### Классификация по месту постоянного хранения данных

* ***Оперативная память***, так называемые in-memory хранилища,
* ***Вторичная память***, здесь средой постоянного хранения является периферийная энергонезависимая память (вторичная память) — как правило жёсткий диск.
* ***Третичная память***, здесь мы говорим об оптических носителях и магнитных лентах.

### Классификация по модели данных

* ***Иерархическая*** — данные представляют собой объекты связаны между собой иерархическими связями с единственным предком у потомка.
* ***Сетевая*** — является расширением сетевой, с той разницей, что в иерархических структурах потомок должен иметь в точности одного предка, а в сетевой — у потомка может иметься любое число предков.
* ***Объектная*** — информация представлена в виде объектов, как в объектно-ориентированных языках программирования.
* ***Реляционная*** — данные в базе данных представляют собой набор отношений. Эти отношения отвечают определённым условиям целостности. Данные могут быть нормализованы.  
    
  ► ***Про реляционную модель данных написано наиболее загадочно, нужно акцентировать внимание, что здесь мы только определяем словарь, говорим, что объясним их чуть позже, так как именно реляционными базами данных мы и будем заниматься детально!***

### Классификация по степени распределённости

* ***Централизованная***, или ***сосредоточенная*** —, полностью поддерживаемая на одном компьютере.
* ***Распределённая*** — составные части которой размещаются в различных узлах компьютерной сети в соответствии с каким-либо критерием.

## Наиболее популярные СУБД и их SQL

* MS SQL
  + Использует Transact-SQL или T-SQL — процедурное расширение языка SQL, созданное компанией Microsoft (для Microsoft SQL Server) и Sybase (для Sybase ASE).
* Oracle
  + Использует PL/SQL: Procedural Language / Structured Query Language — язык программирования, процедурное расширение языка SQL, разработанное корпорацией Oracle.
* MySQL
  + Использует SQL/PSM: Structured Query Language / Persistent Stored Modules — стандарт, разработанный Американским национальным институтом стандартов (ANSI) в качестве расширения SQL.
* PostgreSQL
  + Использует PL/pgSQL: Procedural Language/PostGres Structured Query Language — процедурное расширение языка SQL, используемое в СУБД PostgreSQL.
* SQLite
  + Использует SQL As Understood By SQLite — собственный синтаксис, поддерживает  
    большинство стандартных команд SQL, что-то отсутствует, что-то добавлено.

## SQL — структурированный язык запросов

Как видно из слайда с популярными СУБД, везде присутствует одна и та же аббревиатура — SQL. При этом почти везде к ней приписаны дополнительные аббревиатуры.

SQL — Structured Query Language, структурированный язык запросов, как правило, использующийся для общения с СУБД. С помощью команд на этом языке можно:

* Управлять практически всеми аспектами работы базы данных,
* Производить тонкую настройку производительности под конкретные задачи.

***Как правило, каждая СУБД использует свой диалект языка SQL. Основные конструкции работают одинаково, однако тонкие настройки запросов, как правило, различны.***

Показываем пример T-SQL-запроса на слайде.

## Зачем изучать SQL если есть ORM?

Рассказываем, что есть ORM - Object-Relational Mapping, объектно-реляционное преобразование. Это технология, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных».

ORM значительно упрощает и ускоряет разработку простых задач, однако для сложных сценариев может строить неоптимальные запросы, что может приводить потерям производительности при обращениях к БД.

► ***Умение анализировать сгенерированные запросы, оптимизировать ихи предлагать собственные более производительные альтернативы — вот чем интересно изучение возможностей языка SQL.***

## Реляционная база данных

Реляционная база данных опирается на два ключевых понятия

* Отношения между данными, описывающими некие сущности, хранящиеся в базе
* Нормализация хранящихся данных

Мы рассмотрим эти понятия сегодня на практике.

## 

## СУБД SQL Server 2017 Express Edition

### Преимущества

* Содержит необходимый для среды разработки набор программных средств для создания полноценной базы данных, ограниченной лишь по размерам и используемым ресурсам компьютера
* “Из коробки” обеспечивает разработчика необходимым набором компонентов для интеграции с программами, использующими базы данных
* Официально разрешена для использования в производственной среде без необходимости дополнительного лицензирования

## Установка инструментов

### MS SQL Server Express

Скачиваем с официального сайта MS SQL Server Express <https://www.microsoft.com/ru-ru/sql-server/sql-server-editions-express> и устанавливаем.

Стараемся не ставить лишнего (см. слайд по настройкам установки)

### Microsoft SQL Management Studio

Скачиваем и устанавливаем MS SQL Server Management Studio:

<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017>

или прямая ссылка на английскую версию [thttps://go.microsoft.com/fwlink/?linkid=2088649&clcid=0x409](https://go.microsoft.com/fwlink/?linkid=2088649&clcid=0x409)

и на русскую <https://go.microsoft.com/fwlink/?linkid=2088649>

## Термины

* Таблица / Table - таблица БД :)
* Поле / Field - столбец таблицы БД
* Запись / Record - строка таблицы БД
* Ячейка / Cell - Пересечение столбца и таблицы

# 

# Работа в с базой данных

## Работа в Microsoft Visual Studio

1. Открываем Microsoft Visual Studio,
2. View > SQL Server Object Explorer > Add SQL Server > Local > XXXSQLEXPRESS,
3. Показываем, что объекты БД также видны и здесь,
4. Закрываем Microsoft Visual Studio.

Это удобно, чтобы посмотреть имена объектов во время написания кода, однако для работы с БД есть специализированные инструменты.

## Работа в Microsoft SQL Management Studio

### Интерфейс

* Дерево объектов СУБД слева
* Документы с запросами к БД справа
  + Сверху запрос
  + Снизу результат

### Создание базы данных через UI

В дереве объектов

* Правый щелчок на Databases
* Выбираем пункт New Database…

В диалоговом окне

* Вводим имя БД “Lesson26”
* Смотрим на настройки, но ничего не меняем
* Нажимаем OK

В дереве объектов

* Правый щелчок на Databases
* Выбираем пункт Refresh
* Наблюдаем нашу новую БД Lesson26

Удаляем нашу БД (пункт Delete в контекстном меню нашей базы в дереве объектов)

##### Создание базы данных с помощью SQL-команд:

CREATE DATABASE Banking;

Можно показать файлы на диске (вспоминаем\_путь\_куда\_ставили/Data)

DROP DATABASE Banking;

Можно показать, что файлы пропали.

► ***Обращаем внимание***, что мы написали простые SQL-запросы. Вся работа с СУБД от обслуживания БД до чтения логов может происходить с помощью SQL-запросов. Т.е. запросы бывают не только для работы с данными приложения, но и для настройки хранилища этих данных и сбора статистики по этим данным.

Создаём базу запросом ещё раз. Выделяем слова CREATE DATABASE и нажимаем F1, демонстрируем контекстную справку.

### Создание новой таблицы

*\* по слайду*

### Заведение значений через UI

Контекстное меню таблицы > Edit Top 200 Rows

Вставляем запись

## Генерация запросов

### Создание таблицы

По слайду показать вживую, как генерируется скрипт на создание таблицы.

### Вставка единичной записи

По слайду показать вживую, как генерируется заготовка скрипта на вставку записи в таблицу.

### Выборка записей из таблиц

По слайду показать вживую, как генерируется скрипт на извлечение данных из таблицы.

## Типы данных

Как и в C# у переменных, в SQL Server у каждого столбца (на самом деле, также и у локальной переменной, выражения и параметра) есть определенный тип данных.

Тип данных — атрибут, определяющий, какого рода данные могут храниться в объекте: целые числа, символы, данные денежного типа, метки времени и даты, двоичные строки и так далее.

В SQL Server значительно больше базовых типов данных, чем, например, в C#.

Мы рассмотрим наиболее интересные для нас с точки зрения их соответствия тем или иным типам данных C#

Типы данных в SQL Server объединены в следующие категории:  
(<https://docs.microsoft.com/ru-ru/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-2017>)

|  |  |  |
| --- | --- | --- |
| Точные числа   * bigint * bit * decimal * int * money * numeric * smallint * smallmoney * tinyint   Приблизительные числа   * float * real | Дата и время   * date * datetime2 * datetime * datetimeoffset * smalldatetime * time   Символьные строки   * char * text * varchar   Символьные строки в Юникоде   * nchar * ntext * nvarchar | Двоичные данные   * binary * image * varbinary   Прочие типы данных   * cursor * hierarchyid * sql\_variant * table * rowversion * uniqueidentifier * xml * Типы пространственной геометрии * Типы пространственной географии |

Сопоставления типов данных SQL Server типам данных .NET: <https://docs.microsoft.com/ru-ru/dotnet/framework/data/adonet/sql-server-data-type-mappings>.

# Самостоятельная работа

* Спроектировать и написать SQL-скрипты для создания БД PostOffice из одной таблицы PostalSending, содержащую следующую информацию:
  + ФИО отправителя SenderName
  + ФИО получателя ReceiverName
  + Название документа DocumentTitle
  + Количество страниц NumberOfPages
  + Дата отправки SendingDate
  + Ожидаемая дата доставки ExpectedReceivingDate
* Написать SQL-скрипты для вставки двух записей.
* Написать SQL-скрипты для выборки всех полей всех записей таблицы.
* Написать SQL-скрипты для удаления таблицы и БД.

# 

# Домашнее задание

Предоставить SQL-скрипты, которые

1. Создадут базу данных AirportInfo

2. В ней создадут одну таблицу DepartureBoard (табло вылетов) чтобы в каждом поле хранилась следующая информация (*имена полей и типы данных подбираем самостоятельно*):

* номере рейса
* город и страна вылета и прилёта
* дата и время вылета и прилёта (везде местное)
* время в полёте
* авиакомпания
* модель самолёта

3. Вставят 2 записи в таблицу

4. Вернут все поля всех строк таблицы

5. Удалят базу данных AirportInfo