



# Базы Данных

(БД, СУБД, MS SQL Server Express)

Андрей Голяков

# Термины “БД” и “СУБД”

---

База Данных (БД) / Database (DB), — это структурно хранящаяся информация / структурно хранящиеся данные.

Система Управления Базами Данных (СУБД) / Database Management System (DBMS) — программный комплекс, обеспечивающий управление созданием и использованием баз данных.

Т.е. данные хранятся в БД, а вот доступ к ним (для чтения или обновления) обеспечивается с помощью СУБД.



# Типы БД (по месту 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.



# Типы БД (по модели данных)

---

- **Иерархическая** — данные представляют собой объекты связаны между собой иерархическими связями с единственным предком у потомка.
- **Сетевая** — является расширением иерархической, 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.



# Типы БД (по степени распределённости)

---

Классификация по месту постоянного хранения данных

- **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**

- Transact-SQL или **T-SQL** — процедурное расширение языка SQL, созданное компанией Microsoft (для Microsoft SQL Server) и Sybase (для Sybase ASE)

- **Oracle**

- **PL/SQL**: Procedural Language / Structured Query Language — язык программирования, процедурное расширение языка SQL, разработанное корпорацией Oracle

- **MySQL**

- **SQL/PSM**: Structured Query Language / Persistent Stored Modules — стандарт, разработанный Американским национальным институтом стандартов (ANSI) в качестве расширения SQL

- **PostgreSQL**

- **PL/pgSQL**: Procedural Language/PostGres Structured Query Language — процедурное расширение языка SQL, используемое в СУБД PostgreSQL

- **SQLite**

- **SQL As Understood By SQLite** — собственный синтаксис, поддерживает большинство стандартных команд SQL, что-то отсутствует, что-то добавлено



# Структурированный язык запросов

**SQL** — Structured Query Language, структурированный язык запросов, как правило, использующийся для общения с СУБД.

Пример SQL-запроса для создания таблицы из двух полей:

```
CREATE TABLE Customers (  
    Name nvarchar(60),  
    Age int NULL  
);
```



# Зачем изучать SQL если есть ORM?

---

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 к потерям производительности при обращениях к БД.

Умение анализировать сгенерированные запросы, оптимизировать их и 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 MS SQL Server Express и SQL Management Studio

---

URL для скачивания дистрибутива SQL Server 2017 Express Edition:

<https://www.microsoft.com/ru-ru/sql-server/sql-server-editions-express>

URL для скачивания дистрибутива 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

<https://go.microsoft.com/fwlink/?linkid=2088649&clcid=0x409>

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



# Установка SQL Server 2017 Express Edition

SQL Server 2017 Setup

## Feature Selection

Select the Express features to install.

License Terms

Global Rules

Install Setup Files

Install Rules

**Feature Selection**

Feature Rules

Instance Configuration

Server Configuration

Database Engine Configuration

Feature Configuration Rules

Installation Progress

Complete

**Looking for Reporting Services?** [Download it from the web](#)

**Features:**

**Instance Features**

- ☒ Database Engine Services
- ☐ SQL Server Replication
- ☐ Machine Learning Services (In-Database)
  - ☐ R
  - ☐ Python
- ☐ Full-Text and Semantic Extractions for Search
- ☐ PolyBase Query Service for External Data

**Shared Features**

- ☒ Client Tools Connectivity
- ☒ Client Tools Backwards Compatibility
- ☒ Client Tools SDK
- ☒ SQL Client Connectivity SDK
- ☒ LocalDB

**Redistributable Features**

Instance root directory:

Shared feature directory:

Shared feature directory (x86):

**Feature description:**

Includes the Database Engine, the core service for storing, processing and securing data. The Database Engine provides controlled access and rapid transaction processing.

**Prerequisites for selected features:**

**Already installed:**

- Microsoft Visual C++ 2015 Redistributable
- Windows PowerShell 3.0 or higher
- Microsoft .NET Framework 4.6

**Disk Space Requirements**

Drive C: 1494 MB required, 142159 MB available

< Back

Next >

Cancel



# Установка SQL Server Management Studio

The screenshot displays the Microsoft SQL Server Management Studio (SSMS) interface. The title bar indicates the current session is 'SQLQuery1.sql - RUNB-AGO\SQLEXPRESS.master (CNEU\agolyakov (51))\* - Microsoft SQL Server Management Stu...'. The menu bar includes File, Edit, View, Query, Project, Tools, Window, and Help. The toolbar contains icons for opening files, saving, and executing queries. The 'master' database is selected in the database dropdown. The Object Explorer on the left shows the 'RUNB-AGO\SQLEXPRESS (SQL Server)' instance with a tree view of databases (master, model, msdb, tempdb), database snapshots, security, and logins. The main query editor shows the SQL statement: `SELECT * FROM master.INFORMATION_SCHEMA.TABLES`. The Results pane at the bottom displays a table with 6 rows and 4 columns: TABLE\_CATALOG, TABLE\_SCHEMA, TABLE\_NAME, and TABLE\_TYPE. The status bar at the bottom indicates 'Query executed successfully.' and shows the server name 'RUNB-AGO\SQLEXPRESS (14.0 RTM)', the user 'CNEU\agolyakov (51)', the database 'master', and the execution time '00:00:00' with '6 rows' returned.

SQLQuery1.sql - RUNB-AGO\SQLEXPRESS.master (CNEU\agolyakov (51))\* - Microsoft SQL Server Management Stu... Quick Launch (Ctrl+Q)

File Edit View Query Project Tools Window Help

New Query MDX DMX XMLA DAT

master Execute Find

Object Explorer

Connect

RUNB-AGO\SQLEXPRESS (SQL Server)

- Databases
  - System Databases
    - master
    - model
    - msdb
    - tempdb
  - Database Snapshots
- Security
- Logins
  - ##MS\_PolicyEventProcessingLc
  - ##MS\_PolicyTsqlExecutionLogii
  - BUILTIN\Users

SQLQuery1.sql - RU...EU\agolyakov (51))\*

```
SELECT * FROM master.INFORMATION_SCHEMA.TABLES
```

100 %

Results Messages

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
1	master	dbo	spt_fallback_db	BASE TABLE
2	master	dbo	spt_fallback_dev	BASE TABLE
3	master	dbo	spt_fallback_usg	BASE TABLE
4	master	dbo	spt_values	VIEW
5	master	dbo	spt_monitor	BASE TABLE
6	master	dbo	MSreplication_options	BASE TABLE

Query executed successfully. RUNB-AGO\SQLEXPRESS (14.0 RTM) CNEU\agolyakov (51) master 00:00:00 6 rows

Ready Ln 1 Col 47 Ch 47 INS

# Создание базы данных

---

## 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 ОК

## В дереве объектов

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



# Термины

---

**Таблица** / Table - таблица БД :)

**Поле** / Field - столбец таблицы БД

**Запись** / Record - строка таблицы БД

**Ячейка** / Cell - Пересечение столбца и таблицы



# Создание новой базы данных Banking

---

## В дереве объектов

- Правый щелчок на Databases
- Выбираем пункт New Database...

## В диалоговом окне

- Вводим имя БД **Banking**
- Смотрим на настройки, но ничего не меняем
- Нажимаем ОК

## В дереве объектов

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





# Создание новой таблицы CreditCard

## В дереве объектов

- Раскрываем объекты базы данных Banking
- В контекстном меню узла Tables выбираем **New > Table**

## В диалоговом окне

- Вводим следующие значения:

<u>Column Name</u>	<u>Data Type</u>	<u>Allow Nulls</u>
CardNumber	char(20)	<input type="checkbox"/>
HolderName	varchar(50)	<input checked="" type="checkbox"/>
ExpirationMonth	tinyint	<input type="checkbox"/>
ExpirationYear	tinyint	<input type="checkbox"/>
SecurityCode	smallint	<input type="checkbox"/>

- Сохранить > **CreditCard** > OK



# Ручной ввод данных в таблицы

---

Иногда для скорости удобно ввести или исправить данные в таблице вручную “как в Excel’e”:

1. Контекстное меню таблицы
2. **Edit Top 200 Rows**



# Генерация кода создания таблицы

---

Почти все операции, для которых нам предоставлен интерфейс SQL Server Management Studio, можно сделать через SQL-запрос.

Например, можно получить данные для построения дерева объектов слева.

И конечно же, можно получить запрос создания нашей таблицы:

1. Контекстное меню таблицы
2. **Script Table as > Create to > New Query Editor Window**

```
CREATE TABLE [dbo].[CreditCard](  
    [CardNumber] [char](20) NOT NULL,  
    [HolderName] [char](50) NULL,  
    [ExpirationMonth] [tinyint] NOT NULL,  
    [ExpirationYear] [tinyint] NOT NULL,  
    [SecurityCode] [smallint] NOT NULL  
    ) ON [PRIMARY]  
GO
```

# Запросы на вставку, выборку и удаление

-- Вставка записи в таблицу

```
INSERT INTO [dbo].[CreditCard] (  
    [CardNumber],  
    [HolderName],  
    [ExpirationMonth],  
    [ExpirationYear],  
    [SecurityCode])  
VALUES ('01234567890123456789', 'ANDREY GOLYAKOV', 6, 23, 532)
```

-- Выборка всех полей (\*) всех записей из таблицы

```
SELECT * FROM [dbo].[CreditCard]
```

-- Удаление таблицы

```
DROP TABLE [dbo].[CreditCard]
```

-- Удаление базы данных

```
DROP DATABASE Banking
```



# Типы данных SQL Server 2017

---

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 SQL Server 2017

---

Типы данных в SQL Server объединены в следующие категории:

- Точные числа
- Приблизительные числа
- Дата и время
- Символьные строки
- Символьные строки в Юникоде
- Двоичные данные
- Прочие типы данных

Сопоставления типов данных SQL Server типам данных .NET



# Некоторые соответствия (между типами SQL и C#)

## Точные числа

- bigint Int64
- bit Boolean
- decimal
- int Int32
- money
- numeric
- smallint Int16
- smallmoney
- tinyint Byte

## Приблизительные числа

- float Single
- real Double

## Дата и время

- date DateTime
- datetime2 DateTime
- datetime DateTime
- datetimeoffset DateTimeOffset
- smalldatetime DateTime
- time TimeSpan

## Символьные строки

- char String
- text String
- varchar String

## Символьные строки в Юникоде

- nchar String
- ntext String
- nvarchar String

## Двоичные данные

- binary Byte[]
- image Byte[]
- varbinary Byte[]

## Прочие типы данных

- cursor
- hierarchyid
- sql\_variant
- table
- rowversion
- uniqueidentifier Guid
- xml
- Типы пространственной геометрии
- Типы пространственной географии

# 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.





# Домашняя работа

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



# Спасибо за внимание.

