

# Работа с базами данных

---

# Введение в БД

---

.NET MAUI

# Введение в БД

---

База данных — это набор сведений об объектах, структурированный определенным образом

# Введение в БД

---

Некоторые типы БД:

□ Реляционные базы данных

□ Key-value базы данных

□ Документно-ориентированные базы данных

□ Графовые базы данных



Нереляционные /  
NoSQL

# Введение в БД

---

## Реляционные БД

Данные формируются в таблицы из строк и столбцов.

В строках приводятся сведения об объектах (значения свойств), а в столбцах — сами свойства объектов (поля).

# Введение в БД

---

## Реляционные БД

Данные формируются в таблицы из строк и столбцов.

В строках приводятся сведения об объектах (значения свойств), а в столбцах — сами свойства объектов (поля).

**Примеры:** MS SQL, MySQL, MariaDB, PostgreSQL, SQLite и др.

# Введение в БД

---

Базы данных «Ключ-значение»

- Данные хранятся в виде словаря, где указателем выступает ключ.
- Хранение и обработка разных по типу и содержанию данных.
- Высокая скорость доступа к данным за счет адресного хранения.
- Легкое масштабирование.

**Ограничения:** отсутствие жесткой типизации и структуризации данных, как результат – валидация данных, а также имена ключей отдаются на откуп разработчику.

**Примеры:** Amazon, DynamoDB, Redis, Riak, LevelDB, различные хранилища на основе NoSQL, например Memcached и др.

# Введение в БД

---

## Документоориентированные БД

Данные хранятся в структурированных форматах – XML, JSON, BSON. Сохраняется адресный доступ к данным по ключу. При этом содержимое документа может иметь различный набор свойств.

- хорошо подходят для быстрой разработки систем и сервисов, работающих с по-разному структурированными данными,
- легко масштабируются и меняют структуру при необходимости.

**Примеры:** MongoDB, RethinkDB, CouchDB, DocumentDB.



# Введение в БД

---

## Графовые базы данных

Предназначены для моделирования сложных отношений с помощью теории графов, где связями выступают ребра графа, а сами объекты – это узлы или вершины.

**Особенности:** высокая производительность, поскольку обход ребер и вершин значительно быстрее анализа множества внешних и внутренних таблиц и их соединения по условию отбора в реляционных БД.

**Примеры:** Neo4J, JanusGraph, Dgraph, OrientDB.

# Реляционные БД

---

# Реляционные БД

Элемент реляционной модели	Форма представления
Отношение	Таблица
Заголовок (схема) отношения	Заголовок таблицы
Кортеж	Строка таблицы
Сущность	Описание свойств объекта
Атрибут	Заголовок столбца таблицы
Домен	Множество допустимых значений атрибута
Значение атрибута	Значение поля в записи
Первичный ключ	Один или несколько атрибутов
<i>Тип данных</i>	<i>Тип значений элементов таблицы</i>


# Реляционные БД

---

**Первичный ключ** - атрибут (множество атрибутов)(*столбца*) отношения (*таблицы*), однозначно идентифицирующий (уникально определяет) каждый из его кортежей (*строки*).

# Реляционные БД

**Синтетический** (суррогатный) первичный ключ (synthetic, surrogate PK) – искусственно добавленное в таблицу поле, единственная задача которого – быть первичным ключом



id	passport	fml_name
1	MP334455	Иванов И.И.
2	MP112233	Иванов И.И.
3	MM998877	Петров П.П.
4	MM991122	Сидоров С.С.
5	MA334455	Сидоров С.С.
6		Какой-то мужик с бодуна.

# Реляционные БД

---

Связь **«1 ко многим»** («многие к 1-му») определяет ситуацию, когда одной записи родительской таблицы соответствует несколько записей дочерней таблицы

Связь **«многие ко многим»** определяет ситуацию, когда любой записи одной таблицы может соответствовать много записей другой таблицы и наоборот.

Связь **«М-М»** в реальных БД реализуется через две связи «1-М».

Связь **«1 к 1-му»** определяет ситуацию, когда любой записи одной таблицы может соответствовать ровно одна запись другой таблицы и наоборот.

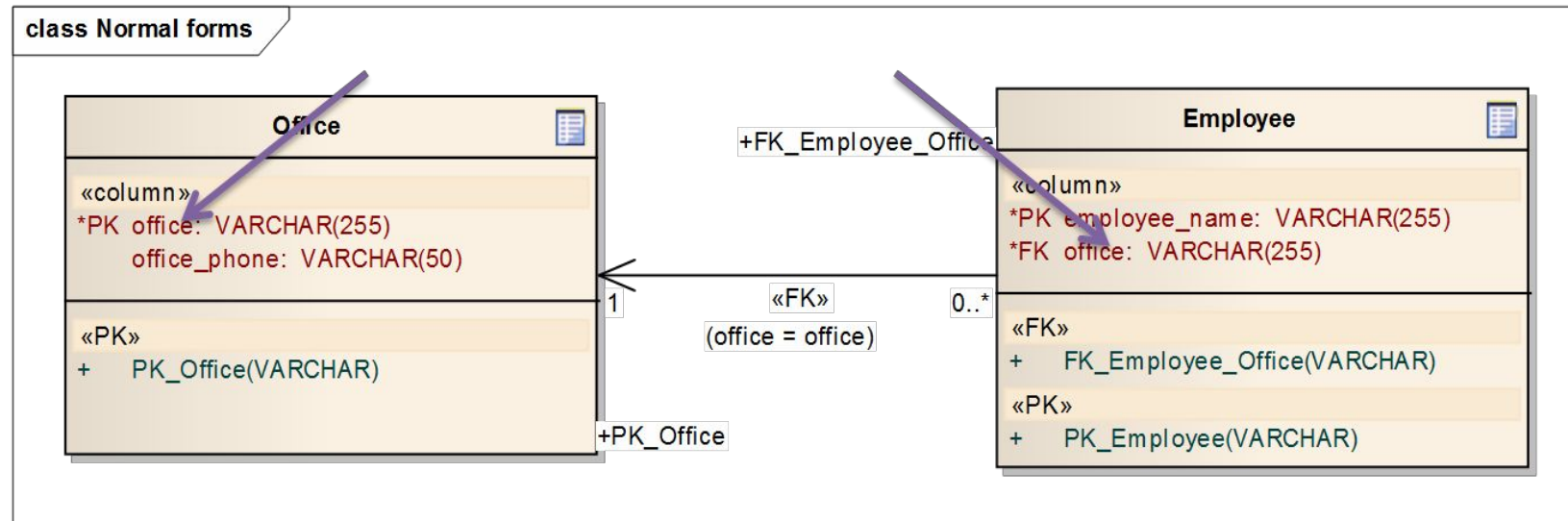
# Реляционные БД

Отношение между классами:



# Реляционные БД

## Отношения между таблицами БД:





# Реляционные БД

---

**Идентифицирующая** связь (identifying relationship) определяет ситуацию, когда запись в дочерней таблице обязана быть связана с записью в родительской таблице (внешний ключ – FK – **не может** принимать значение 0).

**НЕидентифицирующая** связь (nonidentifying relationship) определяет ситуацию, когда запись в дочерней таблице может быть НЕ связана с записью в родительской таблице. (внешний ключ – FK – может принимать значение 0)

# SQL

---

# SQL

---

**SQL** (*Structured Query Language* — ЯЗЫК  
структурированных запросов)

# SQL

---

Язык SQL является декларативным языком. Основное предназначение языка SQL - обеспечение взаимодействия с базами данных, путем формирования различного рода запросов, которые состоят из специальных операторов.

# SQL

Язык SQL основан на запросах (query) – инструкциях, оформленных определённым образом.

Каждая такая инструкция «просит» СУБД выполнить определённое действие, например извлечь данные, создать таблицу и др.

Все инструкции SQL имеют одинаковую структуру, которая изображена на рисунке:



# SQL

---

## Создание базы данных

```
CREATE DATABASE имя_базы_данных;
```

# SQL

---

## Создание таблицы

```
CREATE TABLE название_таблицы  
(название_столбца1 тип_данных атрибуты_столбца1,  
название_столбца2 тип_данных атрибуты_столбца2,  
.....  
название_столбцаN тип_данных атрибуты_столбцаN,  
атрибуты_уровня_таблицы )
```

# SQL

---

## Примеры запросов:

```
1 SELECT * FROM Products
2 WHERE ProductCount IS NULL;
```

```
1 SELECT * FROM Products
2 WHERE Price > (SELECT AVG(Price) FROM Products)
```

```
1 UPDATE Orders
2 SET Price = (SELECT Price FROM Products WHERE Id=Orders.ProductId) + 3000
3 WHERE Id=1;
```



# SQLite.Net

---

.NET MAUI

# SQLite.Net

---

Ядро базы данных SQLite позволяет приложениям .NET MAUI загружать и сохранять объекты данных в общем коде.

Вы можете интегрировать SQLite.NET в приложения .NET MAUI для хранения и извлечения информации в локальной базе данных.

<https://learn.microsoft.com/en-us/dotnet/maui/data-cloud/database-sqlite>

<https://github.com/praeclarum/sqlite-net>

<https://github.com/praeclarum/sqlite-net/wiki>

# SQLite.Net

---

Требуется пакет NuGet `sqlite-net-pcl`.

Существует ряд пакетов NuGet с похожими именами. Правильный пакет имеет следующие атрибуты:

**ID:** `sqlite-net-pcl`

**Authors:** SQLite-net

**Owners:** praeclarum

**NuGet link:** [sqlite-net-pcl](#)

# SQLite.Net

---

В дополнение к sqlite-net-pcl необходимо временно NgGet пакет, который предоставляет SQLite на каждой платформе:

**ID:** SQLitePCLRaw.bundle\_green

**Version:** 2.1.2

**Authors:** Eric Sink

**Owners:** Eric Sink

**NuGet link:** [SQLitePCLRaw.bundle\\_green](#)

# SQLite.Net

---

Пример настройки подключения к БД:

SQLiteOpenFlags Flags =

// open the database in read/write mode

SQLite.SQLiteOpenFlags.ReadWrite |

// create the database if it doesn't exist

SQLite.SQLiteOpenFlags.Create |

// enable multi-threaded database access

SQLite.SQLiteOpenFlags.SharedCache;

# SQLite.Net

---

Создание подключения в БД:

```
string dbPath = Path.Combine(FileSystem.AppDataDirectory,  
                               databaseFilename);  
_db = new SQLiteConnection(dbPath);
```

# SQLite.Net

Пример классов сущностей:

```
[Table("Courses")]
```

```
internal class Course
```

```
{
```

```
    [PrimaryKey, AutoIncrement]
```

```
    public int CourseId { get; set; }
```

```
    public string Name { get; set; }
```

```
    public DateTime StartDate { get; set; }
```

```
    public int Duration { get; set; }
```

```
}
```

# SQLite.Net

## Пример классов сущностей:

```
[Table("Trainees")]  
internal class Trainee  
{  
    [PrimaryKey, Indexed, AutoIncrement]  
    [Column("Id")]  
    public int TraineeId { get; set; }  
    public string Name { get; set; }  
    public string Photo { get; set; }  
    [Indexed]  
    public int CourseId { get; set; }  
}
```



# SQLite.Net

---

Создание таблиц в БД:

```
_db.CreateTable<Course>();
```

```
_db.CreateTable<Trainee>();
```

# SQLite.Net

---

Добавление данных в БД:

```
var course = new Course()  
{  
    StartDate = DateTime.Now.AddDays(15),  
    Duration = 1,  
    Name = "Введение в запой"  
};  
_db.Insert(course);
```

Id !!!

# SQLite.Net

---

Чтение данных из БД:

```
_db.Table<Trainee>()  
    .Where(t => t.CourseId == id)  
    .ToList();
```

# SQLite.Net

