Информатика

БД | DB

База данных. Определение

БД?

СУБД?

Модели данных

Данные в БД хранятся и обрабатываются согласно модели данных

- Описание типов и структур данных в БД
- Манипулирование данными
- Поддержка целостности БД

Классификация БД по модели

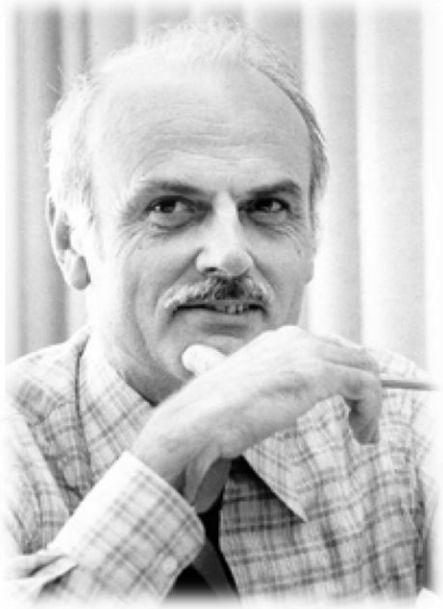
- **Навигационные** (ссылки)
  - Иерархические
  - Сетевые
  - Графовые
- **Реляционные**
- **Объектные** (объектно-ориентированные)
- **Документо-ориентированные**

История

- 1960-е Навигационные СУБД
- 1970-е Реляционные СУБД
- 1980-е SQL, СУБД для настольных ПК
- 1990-е Объектно-ориентированные СУБД
- 2000-е NoSQL и NewSQL

Реляционные БД

- 1969, Эдгар Франк Кодд
- Реляционная модель данных
- БД – связанные таблицы
- Наиболее популярная модель БД



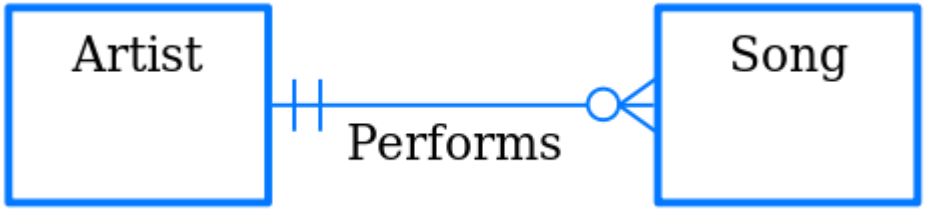
Синонимы

- Таблица, отношение, реляция
- Кортеж, строка, объект, сущность
- Столбец, поле, атрибут

ID	Студент	Группа
123456	Иванов	11-607
543210	Петрова	11-606
123123	Сидоров	11-608
321321	Никитин	11-607
543123	Андреева	11-606

Связи между таблицами

- Виды связей:
  - **Один-к-одному**
    - Студент-зачетка
  - **Один-ко-многим**
    - Группа-студенты
  - **Многие-ко-многим**
    - Предметы-преподаватели



Crow’s Foot notation

Реализация связей

Первичный ключ (Primary Key, PK)

- У каждой таблицы должен быть ***уникальный набор полей для однозначной идентификации записей***
  - Части PK могут быть не уникальны
- Очень часто – целочисленное поле ID
  - Иногда генерируемый автоматически

Найдите все первичные ключи

Classes

Студент	Предмет	Группа	Факультет	Начало
Иванов	Информатика	11-606	ИТИС	8:30
Петрова	Мат. Анализ	11-607	ИТИС	10:10
Сидоров	Мат. Анализ	11-608	ИТИС	10:10
Андреева	Алгебра	11-606	ИТИС	11:50
Ильина	Информатика	11-607	ИТИС	8:30

Реализация связей

Внешний ключ (Foreign Key, FK)

- Набор полей, по которым однозначно идентифицируется ДРУГАЯ строка в ДРУГОЙ таблице
- **Ссылка** на Primary Key в другой таблицы

Пример внешнего ключа

Студент (PK)	Группа (FK)
Иванов	11-606
Петрова	11-607
Сидоров	11-608
Андреева	11-606

Группа (PK)	Куратор
11-605	…
11-606	…
11-607	…
11-608	…

Проблемы БД

Избыточность

Студент	Предмет	Группа	Факультет	Начало
Иванов	Информатика	11-606	ИТИС	8:30
Петрова	Мат. Анализ	11-607	ИТИС	10:10
Сидоров	Мат. Анализ	11-608	ИТИС	10:10
Андреева	Алгебра	11-606	ИТИС	11:50
Ильина	Информатика	11-607	ИТИС	8:30

- Дублируются связи
  - Студент-группа
  - Группа-факультет
- Очевидно, предметы связаны с группами, не нужно их писать для каждого студента

Проблемы БД

- Ссылочная целостность
- Корректность, додстоверность, адекватность, логичность БД
- Если расформировывают группу, куда девать связанных с ней студентов?
  - Оставить? – ссылка на несуществующую группу
  - Удалить? (каскадно) – но они же учатся

Аномалии

- **Модификации** – изменение данных может повлечь просмотр всей таблицы и соответствующее изменение некоторых записей
- **Удаления** – при удалении записи из таблицы может пропасть информация, напрямую не связанная с удаляемой записью
- **Добавления** – невозможность помещения неполной информации, необходимость дополнительного просмотра таблицы

Нормальные формы

Normal Forms

- Для устранения наиболее частых ошибок проектирования БД (аномалий)
- Один из самых популярных вопросов на собеседованиях

Метод нормальных форм

- ***Анализ*** полей отношения, ***декомпозиция*** отношения на несколько взаимосвязанных отношений *на основе процедур нормализации*

1НФ

- **В ячейке одно значение**
  - **Простое, цельное, неделимое**
- Не должно быть повторений строк
- Пример нарушения:
  - *Список значений в ячейке*
    Решение – декомпозиция на несколько записей

1НФ. Пример

Не соответствует:

Студент	Предмет
Иванов	Информатика, АиСД
Петрова	АиСД

Соответствует:

Студент	Предмет
Иванов	Информатика
Иванов	АиСД
Петрова	АиСД

Функциональная зависимость

- B зависит от A (A определяет B)
- A -> B
- Если два объекта совпадают по A, то они совпадают и по B
  - Если у студентов одна группа, то у них один факультет. Значит, есть функциональная зависимость «Группа->Факультет»

2НФ

- **1НФ**
- **Каждый неключевой атрибут *неприводимо* зависит от первичного ключа**

Неприводимость – в составе первичного ключа отсутствует подмножество полей, от которых можно вывести данную функциональную зависимость

  - Не должно быть зависимостей только от части ключа

Пример 2НФ

- Не соответствует:
  - Оценка зависит от обоих полей РК
  - Группа зависит только от студента

Студент	Предмет	Группа	Оценка
Иванов	Информатика	11-606	95
Петрова	Мат. Анализ	11-607	100
Сидоров	Мат. Анализ	11-608	97
Андреева	Алгебра	11-606	100

2НФ. Пример

- Соответствует:

Студент	Предмет	Оценка
Иванов	Информатика	95
Петрова	Мат. Анализ	100
Сидоров	Мат. Анализ	97
Андреева	Алгебра	100

Студент	Группа
Иванов	11-606
Петрова	11-607
Сидоров	11-608
Андреева	11-606

Транзитивность

A < B, B < C – значит, A < C

A = B, B = C – значит A = C

A -> B, B -> C – значит A -> C

-> - импликация

- Пример *нетранзитивного* отношения?

3НФ

- **2НФ**
- **Ни один неключевой атрибут не находится в *транзитивной* функциональной зависимости от первичного ключа**

Проще: Выносить в отдельные таблицы все неключевые поля, содержимое которых может относиться к нескольким записям таблицы

3НФ. Пример

Не соответствует:

Студент	Группа	Факультет
Иванов	11-606	ИТИС
Петрова	11-607	ИТИС
Сидоров	11-608	ИТИС
Юрьев	09-611	ИВМиИТ

3НФ. Пример

Соответствует:

Студент	Группа
Иванов	11-606
Петрова	11-607
Сидоров	11-608
Юрьев	09-611

Группа	Факультет
11-606	ИТИС
11-607	ИТИС
11-608	ИТИС
09-611	ИВМиИТ

НФ Бойса-Кодда

- **Усиленная 3НФ**
  - Не существует функциональной зависимости, в которой левая часть(детерминант) не является потенциальным ключом отношения
  - В случае одного потенциального ключа совпадает с 3НФ
- 3НФ не совсем подходит, если
  - У отношения два или более потенциальных ключа
  - Два и более потенциальных ключа – составные
  - Потенциальные ключи имеют хотя бы один общий атрибут

3НФБК. Пример

- Не соответствует:
  ключи {1,2},{1,3},{4,2},{4,3}
  есть функциональная зависимость Формат->Экзамен

Экзамен	Время начала	Время окончания	Формат
Информатика	09:30	10:30	Информатика. Полуавтомат
Информатика	11:00	12:30	Информатика. Стандарт
Информатика	13:00	14:30	Информатика. Стандарт
Мат. анализ	10:00	11:00	Мат. Анализ. Полуавтомат
Мат. анализ	11:30	12:30	Мат. Анализ. Полуавтомат
Мат. анализ	13:00	14:30	Мат. Анализ. Стандарт

ЗНФБК. Пример

Экзамен	Формат	Полуавтомат
Информатика	Информатика. Полуавтомат	Да
Информатика	Информатика. Стандарт	Нет
Информатика	Информатика. Стандарт	Нет
Мат. анализ	Мат. Анализ. Полуавтомат	Да
Мат. анализ	Мат. Анализ. Полуавтомат	Да
Мат. анализ	Мат. Анализ. Стандарт	Нет

Формат	Время начала	Время окончания
Информатика. Полуавтомат	09:30	10:30
Информатика. Стандарт	11:00	12:30
Информатика. Стандарт	13:00	14:30
Мат. Анализ. Полуавтомат	10:00	11:00
Мат. Анализ. Полуавтомат	11:30	12:30
Мат. Анализ. Стандарт	13:00	14:30

4НФ

- НФ БК
- Не содержит нетривиальных многозначных зависимостей

Многозначная зависимость:

A -> B | C

4НФ. Пример

- Не соответствует:

Предмет	Учебник	Преподаватель
Мат.анализ	Кудрявцев	Преподаватель1
Мат.анализ	Никольский	Преподаватель1
Алгебра	Ильин-Позняк	Преподаватель2
Алгебра	Ильин-Позняк	Преподаватель3
Алгебра	Гантмахер	Преподаватель2
Алгебра	Гантмахер	Преподаватель3

Многозначная зависимость:
Предмет -> Учебник | Преподаватель
Пусть учебник зависит только от предмета

4НФ. Пример

- Декомпозируем. Теперь соответствует:

Предмет	Учебник
Мат.анализ	Кудрявцев
Мат.анализ	Никольский
Алгебра	Ильин-Позняк
Алгебра	Гантмахер

Предмет	Преподаватель
Мат.анализ	Преподаватель1
Алгебра	Преподаватель2
Алгебра	Преподаватель3

4НФ или нет?

- А если каждый преподаватель подбирает учебники на своё усмотрение, какой случай получаем?

4НФ или нет?

- А если каждый преподаватель подбирает учебники на своё усмотрение, какой случай получаем?
  - Транзитивную зависимость
    - Предмет -> Преподаватель -> Учебник

4НФ соблюдена, если

- Есть атрибут, функционально зависящий от первичного ключа
- Хотим знать связь преподавателя и учебника с успеваемостью

Предмет	Учебник	Преподаватель	Балл
Мат.анализ	Кудрявцев	Преподаватель1	95
Мат.анализ	Никольский	Преподаватель1	90
Алгебра	Ильин-Позняк	Преподаватель2	100
Алгебра	Ильин-Позняк	Преподаватель3	98
Алгебра	Гантмахер	Преподаватель2	96
Алгебра	Гантмахер	Преподаватель3	93

5НФ

- См. прямое соединение, зависимость соединения, декомпозиция без потерь

Предмет	Факультет	Преподаватель
Информатика	ИТИС	Марченко
Информатика	ИТИС	Абрамский
Дискретная математика	ИМиМ	Калимуллин
Дискретная математика	ИТИС	Калимуллин
Теория алгоритмов	ИМиМ	Калимуллин

5НФ

Ограничения:

- Предмет может читаться определенным преподавателем
- Определенные предметы читаются только на определенных факультетах
- Преподаватели читают только на определенных факультетах

Соединение

- Нельзя так просто взять и попарно соединить предметы с преподавателями, а затем с факультетами

Предмет	**Факультет**	**Преподаватель**
Информатика	ИТИС	Марченко
Информатика	ИТИС	Абрамский
~~Информатика~~	~~ИТИС~~	~~Калимуллин~~
	…	

5НФ

- *Каждая нетривиальная зависимость соединения определяется только возможным ключом*

5НФ

Предмет	Преподаватель
Информатика	Марченко
Информатика	Абрамский
Дискретная математика	Калимуллин
Дискретная математика	Калимуллин
Теория алгоритмов	Калимуллин

Предмет	Факультет
Информатика	ИТИС
Дискретная математика	ИМиМ
Дискретная математика	ИТИС
Теория алгоритмов	ИМиМ

Факультет	Преподаватель
ИТИС	Марченко
ИТИС	Абрамский
ИМиМ	Калимуллин
ИМиМ	Калимуллин

6НФ, ДКНФ

- Доменно-ключевая нормальная форма
- 6 нормальная форма

- Посмотрите самостоятельно

Structured Query Language (SQL)

- Декларативный
- Основан на исчислении кортежей (см. теорема Кодда о равномощности реляционной алгебры, исчисления кортежей и SQL)
- Имеет диалекты (PL/SQL, Transact-SQL,…)

SQL

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Control Language (DCL)
- Transaction Control Language (TCL)

SQL

- Data Definition Language (DDL)
  create, alter, drop
- Data Manipulation Language (DML)
  select, insert, update, delete
- Data Control Language (DCL)
  grant, revoke
- Transaction Control Language (TCL)
  commit, rollback

Пример. Пусть будет

```
CREATE TABLE "STUDENTS" (
      "ID" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
      "NAME" VARCHAR(30) NOT NULL,
      "GROUP_NAME" VARCHAR(10)
);
…
SELECT NAME
FROM STUDENTS
WHERE GROUP_NAME = '11-607';
```

СУБД

ПО для работы с базами данных

Классификация по способу доступа:

- Файл-серверный:
  - FoxPro, Access
- Клиент-серверный:
  - MS SQL, PostgreSQL, Oracle
- Встроенный:
  - SQLite

Работа с БД

- Для взаимодействия приложения с БД необходим провайдер данных (для каждой СУБД свой)
- Также провайдеры разные в зависимости от технологии работы с БД

Способы работы с БД в C#

- ADO.NET System.Data.SqlClient
- LINQ to SQL System.Data.Linq
- LINQ to Entities System.Data.Entity

SQL Client

- Позволяет подключаться к БД и выполнять команды (NonQuery, Reader, Scalar)

```
string connectionString =
@"Data Source=.\SQLEXPRESS;Initial Catalog=usersdb;Integrated Security=True";

string sqlExpression = "UPDATE Users SET Age=20 WHERE Name='Tom'";
using (SqlConnection connection = new SqlConnection(connectionString))
{
    connection.Open();
    SqlCommand command = new SqlCommand(sqlExpression, connection);
    int number = command.ExecuteNonQuery();
    Console.WriteLine("Обновлено объектов: {0}", number);
}
```

Работа с базой данных

- При работе с базой данных из приложения нам нужно работать с экземплярами классов-сущностей

```
CREATE TABLE "STUDENTS" (
        "ID" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
        "NAME" VARCHAR(30) NOT NULL,
        "GROUP_NAME" VARCHAR(10)
);
```

Присмотримся к STUDENTS

```
CREATE TABLE "STUDENTS" (
        "ID" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
        "NAME" VARCHAR(30) NOT NULL,
        "GROUP_NAME" VARCHAR(10)
);
```

```
public class Student
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string GroupName { get; set; }
}
```

Student -> Group

```
CREATE TABLE “GROUPS”
“ID” INTEGER NOT NULL PRIMARY KEY …,
“NAME” VARCHAR(10) NOT NULL,
);

CREATE TABLE “STUDENTS” (
“ID” INTEGER NOT NULL PRIMARY KEY …,
“NAME” VARCHAR(30) NOT NULL,
“GROUP_ID” INTEGER REFERENCES “GROUPS”(ID)
);
```

Student -> Group

```
class Group
{
    public int Id { get; set; }
    public string Name { get; set; }
}

class Student
{
    public int Id { get; set; }
    public string Name { get; set; }
    public int GroupId { get; set; }
}
```

Student -> Group. Ссылки

```
class Group
{
    public int Id { get; set; }
    public string Name { get; set; }
    public ICollection<Student> Students { get; set; }
}

class Student
{
    public int Id { get; set; }
    public string Name { get; set; }
    public Group Group { get; set; }
    public int GroupId { get; set; }
}
```

Object-relational mapping

- Сопоставление БД с классами
- Самая популярная технология работы с БД
  - EntityFramework, Dapper
  - JPA, Hybernate
  - Django models
  - Ruby on Rails Active Records, etc

Отображение

Relational DB	ORM
Объявление таблицы	Объявление класса
Строка таблицы	Объект – экземпляр класса
Таблица students	Коллекция<Student>
Столбец, поле	Поле, свойство

EntityFramework

- Рекомендованная ORM для .NET Framework (в том числе Core)
- Использует LINQ для запросов к БД
- Entity Data Model для сопоставления классов сущностей с таблицами в БД

Entity Framework

- 3 уровня Entity Data Model:
  - Концептуальный. Определение классов сущностей
  - Хранилица. Определяет таблицы, отношения и типы данных
  - Сопоставления. Посредник, определяет mapping

Взаимодействие с БД

- Database First. Создание классов сущностей по существующей базе
- Model First. По модели данных создается база данных и сущности
- Code First. По классам сущностей генерируется база

Поддержка связей

- Поддерживаются связи
  - Один-к-одному
  - Один-ко-многим
  - Многие-ко-многим
    - При подходе CodeFirst автоматически генерируется вспомогательная таблица

Основа Entity Framework

- Классы
  - DbContext – контекст данных для взаимодействия с базой
  - DbModelBuilder – сопоставляет классы с таблицами
  - DbSet/DbSet<Tentity> - наборы сущностей, соответствующие таблицам

Пример

```
class UserContext : DbContext
{
    public UserContext()
        : base("DbConnection")
    { }

    public DbSet<User> Users { get; set; }
}
```

```
using (UserContext db = new UserContext())
{
    // создаем два объекта User
    User user1 = new User { Name = "Tom", Age = 33 };
    User user2 = new User { Name = "Sam", Age = 26 };

    // добавляем их в бд
    db.Users.Add(user1);
    db.Users.Add(user2);
    db.SaveChanges();
}
```

Миграции

- Обновление базы данных при изменении моделей и контекста данных
  - Создание, удаление столбцов/таблиц, ключей, индексов, хранимых процедур
- enable-migrations
- add-migration “MigrationName”
- update-database (-script)

Почитать

- Нормальные формы (хабр)
- Entity Framework (Metanit)
- Открытая площадка с онлайн курсами от Microsoft (aka Virtual Academy)

Вопросы?

e-mail: marchenko@it.kfu.ru