



# БАЗЫ ДАННЫХ

**Дьячкова Марина Сергеевна**

Кафедра ПМик (1 корпус, ауд. 430а)

*mar-1999@mail.ru*

*Telegram через старосту группы*

~~другие соц.сети~~

# Курс

## **Курс БД\_ПОСВТ (Базы данных)**

<https://eios.sibsutis.ru/course/view.php?id=1242>

*Кодовое слово: BD2025*

### **Лекции**

- Посещения
- Тесты на лекциях

### **Практические занятия**

- Практические работы
- РГР

### **Экзамен (возможен автомат)**

# План курса

| ЛЕКЦИИ                           | ПРАКТИЧЕСКИЕ РАБОТЫ |
|----------------------------------|---------------------|
| Блок 1 – SQL и Теория БД         |                     |
| 8                                | 6+1*                |
| РГР (8 и 9 неделя)               |                     |
| Блок 2 – PostgresPro, курс DEV-1 |                     |
| 9                                | 8                   |

**Возможны изменения!!**

| ПРАКТИЧЕСКИЕ РАБОТЫ                 |                 |
|-------------------------------------|-----------------|
| ЛР 1 SQL                            | Неделя 1        |
| ЛР 2 SQL                            | Неделя 2        |
| ЛР 3 SQL                            | Неделя 3        |
| ЛР 4 SQL                            | Неделя 4        |
| ЛР 5 SQL                            | Неделя 5        |
| ЛР 6 SQL                            | Неделя 6        |
| ЛР 7* (не обязательна на 3 и 4) SQL | <b>Неделя 7</b> |
| РГР Проектирование БД               | Недели 8-9      |
| ЛР 8(1) PostgresPro DEV-1           | Неделя 10       |
| ЛР 9(2) PostgresPro DEV-1           | Неделя 11       |
| ЛР 10(3) PostgresPro DEV-1          | Неделя 12       |
| ЛР 11(4) PostgresPro DEV-1 PL/pgSQL | Неделя 13       |
| ЛР 12(5) PostgresPro DEV-1 PL/pgSQL | Неделя 14       |
| ЛР 13(6) PostgresPro DEV-1 PL/pgSQL | Неделя 15       |
| ЛР 14(7) PostgresPro DEV-1 PL/pgSQL | Неделя 16       |
| ЛР 15(8) PostgresPro DEV-1 PL/pgSQL | Неделя 17       |



PosgresPro

PANGOLIN



# Определения

- **Информационная система (ИС)** - это совокупность аппаратно-программных средств, используемых для решения некоторой **прикладной задачи**.

**ИС** реализует автоматизированный сбор, обработку и манипулирование данными и включает технические средства обработки данных, программное обеспечение и обслуживающий персонал.

- **База данных (БД)** — набор специальным образом **организованных** данных, хранимых в памяти вычислительной системы и отображающих состояние **объектов и их взаимосвязей** в рассматриваемой **предметной области**.

# Определения

- **Система управления базами данных (СУБД)** — это комплекс языковых и программных средств, предназначенный для создания, ведения и совместного использования БД многими пользователями.



# Определения

- **Банк данных (БнД)** – базы данных, СУБД, словаря данных, администратора, вычислительной системы и обслуживающего персонала.

Банк данных – **ИС**, где реализованы функции **централизованного хранения** и накопления информации, организованной в БД.

- **Словарь данных (СД)** централизованно хранить **информацию** о структурах данных, взаимосвязях файлов БД друг с другом, типах данных и форматах их представления, принадлежности данных пользователям, кодах защиты и разграничения доступа и т. п.



# Определения

- **Приложение** — программа или комплекс программ, обеспечивающих автоматизацию обработки информации для прикладной задачи.

Приложения могут создаваться в среде СУБД или вне СУБД (внешние приложения).

- **Администратор базы данных (АБД)** —лицо или группа лиц, отвечающих за выработку требований к БД, ее проектирование, создание, эффективное использование и сопровождение. Администратор следит за функционированием ИС, обеспечивает защиту от несанкционированного доступа, контролирует избыточность, непротиворечивость, сохранность и достоверность хранимой в БД информации.

# Модели данных

- **Модель представления данных** –логическая структура данных, хранимых в базе.

Основные модели представления данных:

- 1) иерархическая;
- 2) сетевая;
- 3) реляционная;
- 4) постреляционная;
- 5) многомерная;
- 6) объектно-ориентированная.

# Иерархическая модель данных

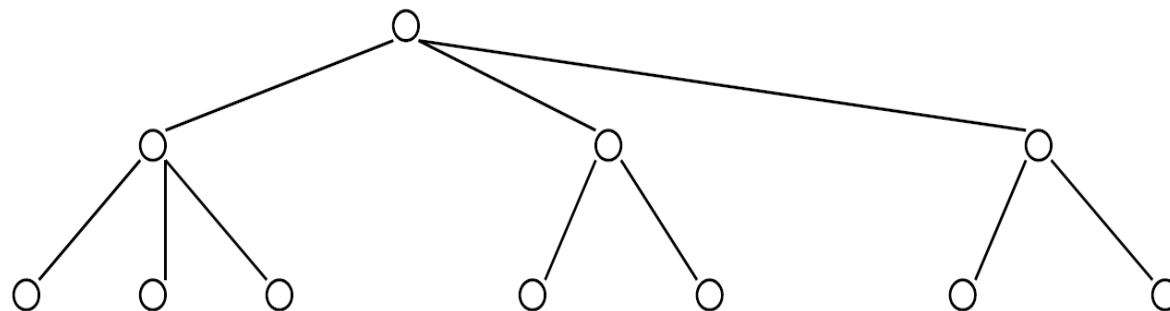
В иерархической модели связи между данными можно описать с помощью **упорядоченного графа** (или дерева).

## Достоинства:

- Эффективное использование памяти компьютера (хранение в линейных списках)
- Подходит для иерархических структур

## Недостатки:

- Громоздкость и сложность связей

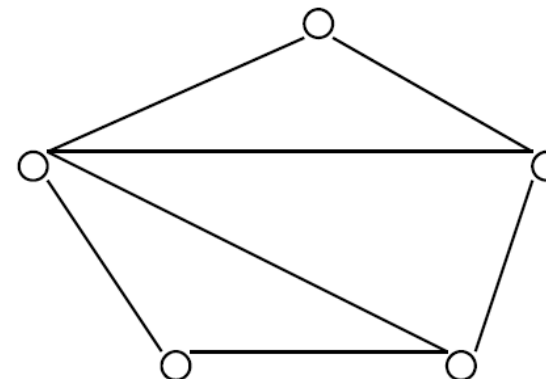


# Сетевая модель данных

Сетевая модель данных позволяет отображать разнообразные взаимосвязи элементов данных в виде **произвольного графа**.

## Достоинства:

- Эффективная реализация по памяти и оперативности
- Больше возможностей в создании произвольных связей



На формирование связи между записями особых ограничений не накладывается

## Недостатки:

- Высокая сложность схемы
- Ослаблен контроль целостности связей



# Реляционная модель данных

Была предложена сотрудником фирмы IBM Эдгаром Коддом в 1970 г.  
Основана на понятии отношение (*relation*).

ОТДЕЛЫ

| Отд_Номер | Отд_Размер | Отд_Зарп |
|-----------|------------|----------|
| 3         | 10         | 50 000   |
| 4         | 15         | 75 000   |

НАЧАЛЬНИКИ

| Отд_Номер | Нач_Номер | Нач_Имя  | Нач_Телеф |
|-----------|-----------|----------|-----------|
| 3         | 31        | Иванов   | 0-11      |
| 4         | 41        | Васильев | 0-12      |

СОТРУДНИКИ

| Отд_Номер | Сотр_Номер | Сотр_Имя | Сотр_Зарп |
|-----------|------------|----------|-----------|
| 3         | 32         | Петров   | 4 000     |
| 3         | 33         | Сидоров  | 4 000     |

**Отношение** – двумерная таблица

**Кортежи** – строки (записи)

**Атрибуты** – записи (поля)

# Реляционная модель данных

Данные хранятся на диске в виде файлов.

## Достоинства:

- Простота, понятность
- Удобство физической реализации на ЭВМ
- Проблемы эффективности разрешимы

## Недостатки:

- Отсутствие стандартных средств идентификации отдельных записей
- Сложность описания иерархических и сетевых связей

# Постреляционная модель данных

Представляет собой расширенную реляционную модель, снимающую ограничение неделимости данных в записях таблиц

- Допускает многозначные поля, «таблица в таблице»

## Достоинства:

- Одна таблица вместо нескольких связанных реляционных таблиц

## Недостатки:

- Сложность обеспечения целостности и непротиворечивости данных

НАКЛАДНЫЕ

| ИНВ_N | ПОКУП_N | НАЗВАНИЕ | КОЛИЧ |
|-------|---------|----------|-------|
| 73    | 23      | Ручка    | 3     |
|       |         | Линейка  | 2     |
| 74    | 45      | Тетрадь  | 1     |
|       |         | Карандаш | 6     |
|       |         | Блокнот  | 2     |
| 88    | 23      | Папка    | 1     |

# Многомерная модель данных

Многомерные СУБД являются узкоспециализированными СУБД, предназначенными для интерактивной аналитической обработки информации (OLAP, системы поддержки принятия решений)

## Реляционная модель:

| МОДЕЛЬ    | МЕСЯЦ  | ОБЪЕМ |
|-----------|--------|-------|
| «Жигули»  | июнь   | 12    |
| «Жигули»  | июль   | 24    |
| «Жигули»  | август | 5     |
| «Москвич» | июнь   | 2     |
| «Москвич» | июль   | 18    |
| «Волга»   | июль   | 19    |

- **Агрегируемость** данных – степень детальности информации зависит от уровня пользователя
- **Историчность** данных – неизменность данных, их привязка ко времени.
- **Прогнозируемость** данных

## Многомерная модель:

| МОДЕЛЬ    | Июнь | Июль | Август |
|-----------|------|------|--------|
| «Жигули»  | 12   | 24   | 5      |
| «Москвич» | 2    | 18   | нет    |
| «Волга»   | нет  | 19   | нет    |



# Многомерная модель данных

- Обычно требуется большее количество измерений (гиперкубы и поликубы)
- Используются «срезы»
- Есть измерение времени

## Достоинства:

- Удобство и эффективность аналитической обработки больших объемов данных, связанных со временем

## Недостатки:

- Громоздкость для простых задач обычной оперативной обработки информации



# Объектно-ориентированная модель данных

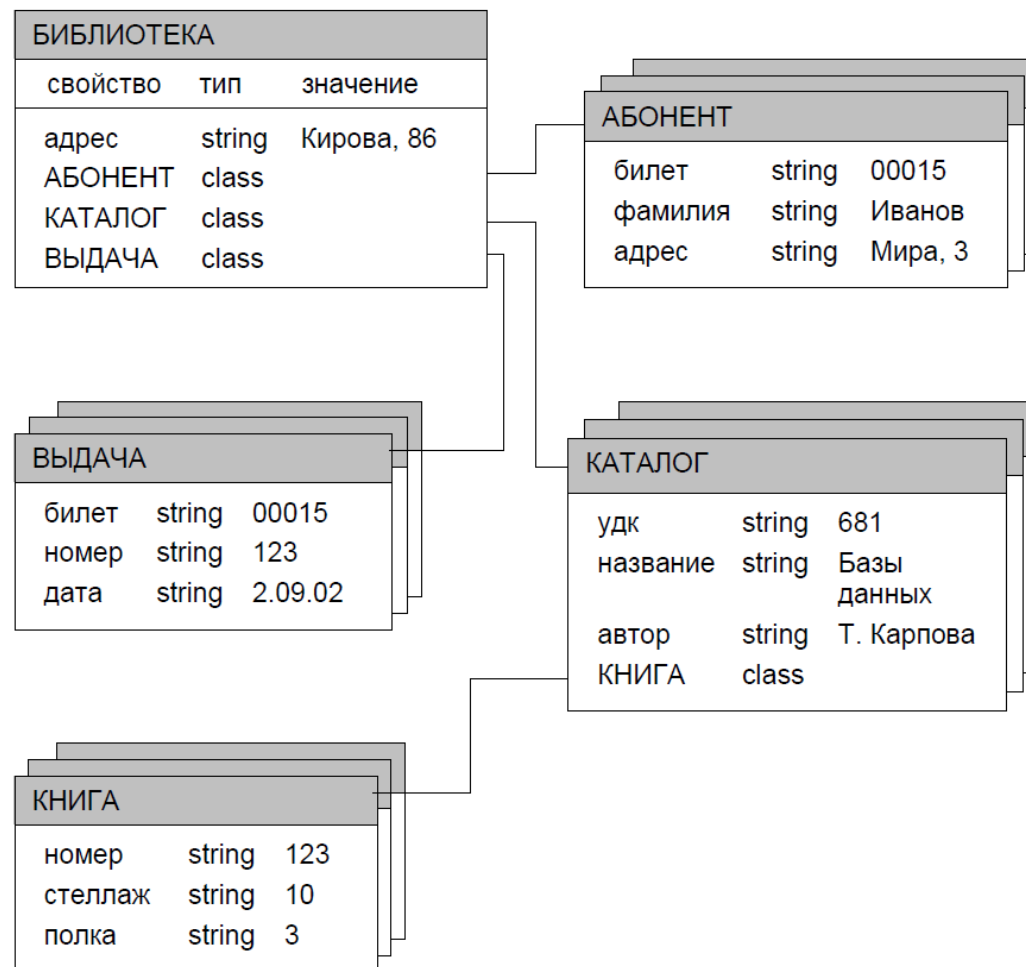
Имеется возможность идентифицировать отдельные записи базы. Между записями БД и функциями их обработки устанавливаются взаимосвязи с помощью механизмов, подобных ООП.

## Достоинства:

- Возможность отображения информации о сложных взаимосвязях объектов.
- Позволяет идентифицировать отдельную запись базы данных и определять функции их обработки.

## Недостатки:

- Высокая понятийная сложность, неудобство обработки данных и низкая скорость выполнения запросов



# Модели данных

- **Модель представления данных** –логическая структура данных, хранимых в базе.

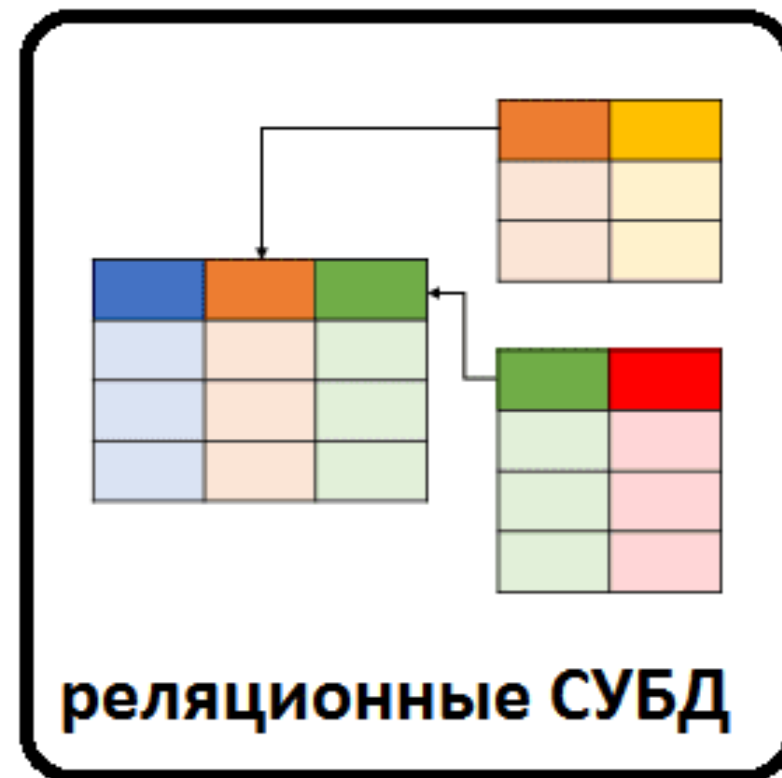
Основные модели представления данных:

- 1) иерархическая;
- 2) сетевая;
- 3) реляционная;
- 4) постреляционная;
- 5) многомерная;
- 6) объектно-ориентированная.

# Реляционные БД



SQL

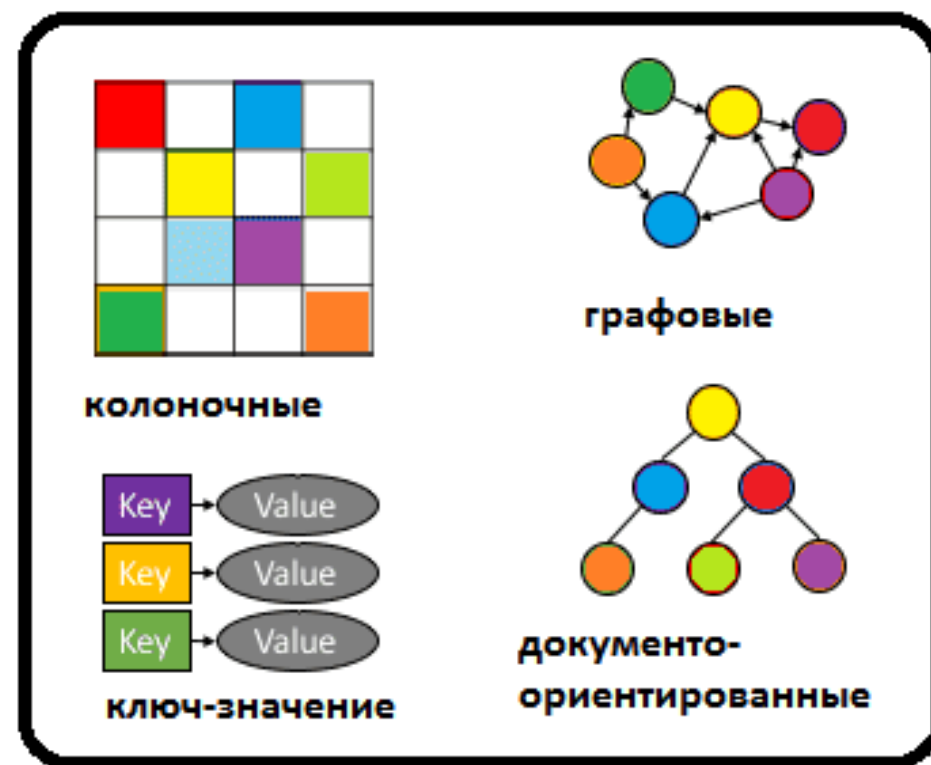


# Нереляционные БД

Причина популярности: «Данные стали сложнее и больше»



## NoSQL



# Реляционная БД



|  | 123  id_stud | A-Z name |
|---|---|----------|
| 1   | 1   | Alice    |
| 2   | 2   | Ivan     |
| 3   | 3   | Denis    |
| 4   | 4   | Eva      |
| 5   | 5   | Petr     |

Таблица student



|  | 123  id_sub | A-Z name | A-Z teacher     |
|---|--|----------|-----------------|
| 1   | 1  | TYAP     | Osipova U.V.    |
| 2   | 2  | ZI       | Dyachkova I.S.  |
| 3   | 3  | PMU      | Nechta I.V.     |
| 4   | 4  | BD       | Dyachkova M.S.  |
| 5   | 5  | MMO      | Dementieva K.I. |

Таблица subject






|  | 123  id_d | 123  id_stud | 123  id_sub |  deadline |
|---|--|---|--|--|
| 1   | 1  | 1   | 1  | 2025-09-30   |
| 2   | 2  | 1   | 2  | 2025-09-20   |
| 3   | 3  | 1   | 3  | 2025-09-25   |
| 4   | 4  | 2   | 1  | 2025-10-01   |
| 5   | 5  | 3   | 3  | 2025-09-30   |
| 6   | 6  | 3   | 4  | 2025-12-31   |
| 7   | 7  | 4   | 2  | 2025-10-31   |

Таблица debt

# Реляционная БД


|   |  123 id_stud <b>РК</b> | A-Z name |
|---|---|----------|
| 1 | 1   | Alice    |
| 2 | 2   | Ivan     |
| 3 | 3   | Denis    |
| 4 | 4   | Eva      |
| 5 | 5   | Petr     |

Таблица student


|   |  123 id_sub <b>РК</b> | A-Z name | A-Z teacher     |
|---|--|----------|-----------------|
| 1 | 1  | TYAP     | Osipova U.V.    |
| 2 | 2  | ZI       | Dyachkova I.S.  |
| 3 | 3  | PMU      | Nechta I.V.     |
| 4 | 4  | BD       | Dyachkova M.S.  |
| 5 | 5  | MMO      | Dementieva K.I. |

Таблица subject



|   |  123 id_d <b>РК</b> | 123 id_stud | 123 id_sub |  deadline |
|---|--|-------------|------------|--|
| 1 | 1  | 1           | 1          | 2025-09-30   |
| 2 | 2  | 1           | 2          | 2025-09-20   |
| 3 | 3  | 1           | 3          | 2025-09-25   |
| 4 | 4  | 2           | 1          | 2025-10-01   |
| 5 | 5  | 3           | 3          | 2025-09-30   |
| 6 | 6  | 3           | 4          | 2025-12-31   |
| 7 | 7  | 4           | 2          | 2025-10-31   |

Таблица debt

# Реляционная БД


|   |  123 id_stud <b>ПК</b> | A-Z name |
|---|---|----------|
| 1 | 1   | Alice    |
| 2 | 2   | Ivan     |
| 3 | 3   | Denis    |
| 4 | 4   | Eva      |
| 5 | 5   | Petr     |

Таблица student


|   |  123 id_sub <b>ПК</b> | A-Z name | A-Z teacher     |
|---|--|----------|-----------------|
| 1 | 1  | TYAP     | Osipova U.V.    |
| 2 | 2  | ZI       | Dyachkova I.S.  |
| 3 | 3  | PMU      | Nechta I.V.     |
| 4 | 4  | BD       | Dyachkova M.S.  |
| 5 | 5  | MMO      | Dementieva K.I. |

Таблица subject

- ПК**  
**Primary key**
- Not null
  - Unique

Обычно int  
(может быть другим)



|   |  123 id_d <b>ПК</b> | 123 id_stud | 123 id_sub |  deadline |
|---|--|-------------|------------|--|
| 1 | 1  | 1           | 1          | 2025-09-30   |
| 2 | 2  | 1           | 2          | 2025-09-20   |
| 3 | 3  | 1           | 3          | 2025-09-25   |
| 4 | 4  | 2           | 1          | 2025-10-01   |
| 5 | 5  | 3           | 3          | 2025-09-30   |
| 6 | 6  | 3           | 4          | 2025-12-31   |
| 7 | 7  | 4           | 2          | 2025-10-31   |

Таблица debt



# Реляционная БД


|   |  123 id_stud <b>PK</b> | A-Z name |
|---|---|----------|
| 1 | 1   | Alice    |
| 2 | 2   | Ivan     |
| 3 | 3   | Denis    |
| 4 | 4   | Eva      |
| 5 | 5   | Petr     |

Таблица student


|   |  123 id_sub <b>PK</b> | A-Z name | A-Z teacher     |
|---|--|----------|-----------------|
| 1 | 1  | TYAP     | Osipova U.V.    |
| 2 | 2  | ZI       | Dyachkova I.S.  |
| 3 | 3  | PMU      | Nechta I.V.     |
| 4 | 4  | BD       | Dyachkova M.S.  |
| 5 | 5  | IMMO     | Dementieva K.I. |

Таблица subject

## FK Foreign key

Ссылается на поле другой  
таблицы

Тип данных и значения  
совпадают с типом поля,  
на которое ссылается





|   |  123 id_d <b>PK</b> |  123 id_stud <b>FK</b> |  123 id_sub <b>FK</b> |  deadline |
|---|--|---|--|--|
| 1 | 1  | 1   | 1  | 2025-09-30   |
| 2 | 2  | 1   | 2  | 2025-09-20   |
| 3 | 3  | 1   | 3  | 2025-09-25   |
| 4 | 4  | 2   | 1  | 2025-10-01   |
| 5 | 5  | 3   | 3  | 2025-09-30   |
| 6 | 6  | 3   | 4  | 2025-12-31   |
| 7 | 7  | 4   | 2  | 2025-10-31   |

Таблица debt

# Реляционная БД


|   |  123 id_stud <b>РК</b> | A-Z name |
|---|---|----------|
| 1 | 1   | Alice    |
| 2 | 2   | Ivan     |
| 3 | 3   | Denis    |
| 4 | 4   | Eva      |
| 5 | 5   | Petr     |

Таблица student


|   |  123 id_sub <b>РК</b> | A-Z name | A-Z teacher     |
|---|--|----------|-----------------|
| 1 | 1  | TYAP     | Osipova U.V.    |
| 2 | 2  | ZI       | Dyachkova I.S.  |
| 3 | 3  | PMU      | Nechta I.V.     |
| 4 | 4  | BD       | Dyachkova M.S.  |
| 5 | 5  | MMO      | Dementieva K.I. |

Таблица subject

Ограничения целостности!





|   |  123 id_d <b>РК</b> |  123 id_stud <b>FK</b> |  123 id_sub <b>FK</b> |  deadline |
|---|--|---|--|--|
| 1 | 1  | 1   | 1  | 2025-09-30   |
| 2 | 2  | 1   | 2  | 2025-09-20   |
| 3 | 3  | 1   | 3  | 2025-09-25   |
| 4 | 4  | 2   | 1  | 2025-10-01   |
| 5 | 5  | 3   | 3  | 2025-09-30   |
| 6 | 6  | 3   | 4  | 2025-12-31   |
| 7 | 7  | 4   | 2  | 2025-10-31   |

Таблица debt

# Язык SQL

Для получения информации из таблиц БД в качестве языка манипулирования данными в теоретическом плане используются три абстрактных языка:

- язык реляционной алгебры;
- язык реляционного исчисления на кортежах;
- язык реляционного исчисления на доменах.

**Structured Query Language** - язык структурных запросов

- Разработан в середине 70-х годов фирмой IBM
- Фактически стал стандартом для работы с БД



# Операторы SQL

Язык SQL содержит 4 **группы** операторов:

- **DDL** – Data Definition Language - операторы описания данных:  
CREATE, DROP, ALTER, TRUNCATE и др.
- **DML** – Data Manipulation Language - операторы манипуляции данными:  
INSERT, DELETE, UPDATE, MERGE, *SELECT*
- **DCL** – Data Control Language – операторы управления правами доступа:  
GRANT / REVOKE , LOCK / UNLOCK , SET LOCK MODE
- **TCL** – Transaction Control Language - операторы управления транзакциями:  
COMMIT, ROLLBACK и SAVE TRANSACTION

# DDL-операторы

- **CREATE** – создание объекта (таблица, представление, индекс, функция и др.)

*CREATE TABLE [IF NOT EXISTS] имя\_таблицы (  
поле\_1 тип\_данных **ограничения\_столбца1**,  
поле\_2 тип\_данных **ограничения\_столбца2**,  
...  
[поле\_n тип\_данных],  
**ограничения\_таблицы**);*

*NOT NULL* – не пустой  
*UNIQUE* – уникальный  
*PRIMARY KEY* – первичный ключ  
*DEFAULT* – значение по умолчанию  
*CHECK* – выполнение условия

...  
*FOREIGN KEY* – внешний ключ

Ограничения можно добавить не только при создании таблицы, но и с помощью команды `ALTER TABLE t_name ADD CONSTRAINT ...`

# DDL-операторы

Ограничения добавлены  
при создании таблицы

```
CREATE TABLE debt (  
    id_d int PRIMARY KEY,  
    id_stud int NOT NULL,  
    id_sub int NOT NULL,  
    deadline date CHECK (deadline > '2025-09-01'),  
  
    FOREIGN KEY (id_stud) REFERENCES student(id_stud),  
    FOREIGN KEY (id_sub) REFERENCES subject(id_sub));
```

Ограничения добавлены отдельно, после создания

```
ALTER TABLE debt ADD PRIMARY KEY (id_d);  
ALTER TABLE debt ADD CONSTRAINT fk_stud FOREIGN KEY (id_stud) REFERENCES student(id_stud);  
ALTER TABLE debt ADD CONSTRAINT chk_deadline CHECK (deadline > '2025-09-01');
```

имя ограничения

# DDL-операторы

- **DROP** – удаление объекта (таблица, представление, индекс, функция и др.)
- Вместе с командой ALTER TABLE для удаления ограничений, столбцов

# Реляционная БД



|  | 123  id_stud | A-Z name |
|---|---|----------|
| 1   | 1   | Alice    |
| 2   | 2   | Ivan     |
| 3   | 3   | Denis    |
| 4   | 4   | Eva      |
| 5   | 5   | Petr     |

Таблица student



|  | 123  id_sub | A-Z name | A-Z teacher     |
|---|--|----------|-----------------|
| 1   | 1  | TYAP     | Osipova U.V.    |
| 2   | 2  | ZI       | Dyachkova I.S.  |
| 3   | 3  | PMU      | Nechta I.V.     |
| 4   | 4  | BD       | Dyachkova M.S.  |
| 5   | 5  | MMO      | Dementieva K.I. |

Таблица subject

Какую таблицу  
можно удалить?






|  | 123  id_d | 123  id_stud | 123  id_sub |  deadline |
|---|--|---|--|--|
| 1   | 1  | 1   | 1  | 2025-09-30   |
| 2   | 2  | 1   | 2  | 2025-09-20   |
| 3   | 3  | 1   | 3  | 2025-09-25   |
| 4   | 4  | 2   | 1  | 2025-10-01   |
| 5   | 5  | 3   | 3  | 2025-09-30   |
| 6   | 6  | 3   | 4  | 2025-12-31   |
| 7   | 7  | 4   | 2  | 2025-10-31   |

Таблица debt



# Реляционная БД


|   |  123 id_stud | A-Z name |
|---|---|----------|
| 1 | 1   | Alice    |
| 2 | 2   | Ivan     |
| 3 | 3   | Denis    |
| 4 | 4   | Eva      |
| 5 | 5   | Petr     |

Таблица student


|   |  123 id_sub | A-Z name | A-Z teacher     |
|---|--|----------|-----------------|
| 1 | 1  | TYAP     | Osipova U.V.    |
| 2 | 2  | ZI       | Dyachkova I.S.  |
| 3 | 3  | PMU      | Nechta I.V.     |
| 4 | 4  | BD       | Dyachkova M.S.  |
| 5 | 5  | MMO      | Dementieva K.I. |

Таблица subject

Можно удалить  
таблицу debt





|   |  123 id_d | 123 id_stud  FK | 123 id_sub  FK |  deadline |
|---|--|--|---|--|
| 1 | 1  | 1  | 1   | 2025-09-30   |
| 2 | 2  | 1  | 2   | 2025-09-20   |
| 3 | 3  | 1  | 3   | 2025-09-25   |
| 4 | 4  | 2  | 1   | 2025-10-01   |
| 5 | 5  | 3  | 3   | 2025-09-30   |
| 6 | 6  | 3  | 4   | 2025-12-31   |
| 7 | 7  | 4  | 2   | 2025-10-31   |

Таблица debt

# DDL-операторы

...

```
... FOREIGN KEY (id_stud) REFERENCES student(id_stud)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE
```

Удаление (изменение) поля повлечет удаление (изменение) всех записей с таким внешним ключом из зависимой таблицы

## ON DELETE/ON UPDATE ...

- SET NULL — устанавливает NULL для внешнего ключа
- SET DEFAULT — устанавливает значение по умолчанию
- RESTRICT — запрещает удаление или обновление (используется по умолчанию)
- NO ACTION — схоже с RESTRICT

# Выборка данных

1. 

`SELECT *  
FROM таблица`

 Минимальный SQL-запрос
2. 

`SELECT поле1 AS a1, поле2 a2  
FROM таблица t`

 Псевдонимы
3. 

`SELECT поле1, ...  
FROM таблица  
WHERE условие`

`>, <, !=, =, <>, IS NULL,  
IN, BETWEEN, ...`
4. 

`SELECT поле1, ...  
FROM таблица  
WHERE условие1 AND условие2`

 AND, OR, NOT

# Выборка данных

5.

```
SELECT поле1, ...  
FROM таблица  
WHERE условие  
ORDER BY поле1
```

Сортировка по полю1 по возрастанию

6.

```
SELECT поле1, ...  
FROM таблица  
WHERE условие  
ORDER BY поле1 DESC
```

Сортировка по полю1 по убыванию

7.

```
SELECT поле1, поле2  
FROM таблица  
WHERE условие  
ORDER BY поле1, поле2
```

Сортировка по полю1, полю2 по возрастанию

8.

```
SELECT поле1, поле2  
FROM таблица  
WHERE условие  
ORDER BY 1, 2
```

Команда **ORDER BY** всегда в конце запроса

# Типы данных в PostgreSQL. Числовые типы

| Наименование            | Размер     | Описание                                  | Диапазон  |
|-------------------------|------------|---|---|
| <b>smallint</b>         | 2 байта    | Целое число малого диапазона              | -32768 до +32767  |
| <b>integer</b>          | 4 байта    | Типичный выбор для целого числа           | -2147483648 до +2147483647                                |
| <b>bigint</b>           | 8 байт     | Целое число большого диапазона            | -9223372036854775808 до 9223372036854775807               |
| <b>decimal</b>          | переменная | Указанное пользователем значение , точное | До 131072 цифр перед запятой; до 16383 цифр после запятой |
| <b>numeric</b>          | переменная | Указанное пользователем значение, точное  | до 131072 цифр до запятой; до 16383 цифр после запятой    |
| <b>real</b>             | 4 байта    | variable-precision,inexact                | Точность 6 десятичных цифр                                |
| <b>double precision</b> | 8 байт     | Переменная точность, неточная             | Точность 15 десятичных цифр                               |
| <b>smallserial</b>      | 2 байта    | Небольшое автоинкрементное целое число    | 1 до 32767  |
| <b>serial</b>           | 4 байта    | Автоинкрементное целое число              | 1 до 2147483647   |
| <b>bigserial</b>        | 8 байт     | Большое автоинкрементное число            | 1 до 92233720368547758                                    |

# Типы данных в PostgreSQL. Строковые и логические типы

## Наименование и Описание

**character varying(n), varchar(n)**

текст с ограничением по длине (максимальная длина строка может быть ограничена)

**character(n), char(n)**

текст фиксированной длины (строка всегда имеет строго заданный размер)

**text**

текст неограниченной длины

| Наименование   | Размер | Описание        |
|----------------|--------|-----------------|
| <b>boolean</b> | 1 байт | Истина или ложь |

# Типы данных в PostgreSQL. Типы даты и времени

| Наименование                                   | Размер  | Описание                                | Нижнее значение | Верхнее значение |
|--|---------|---|-----------------|------------------|
| <b>timestamp</b> [(p)]<br>[без часового пояса] | 8 байт  | Дата и время<br>(без часового пояса)    | 4713 до н.э.    | 294276 н.э.      |
| <b>TIMESTAMPTZ</b>                             | 8 байт  | Дата и время<br>(с часовым поясом)      | 4713 до н.э.    | 294276 н.э.      |
| <b>date</b>                                    | 4 байта | Дата (без времени суток)                | 4713 до н.э.    | 5874897 н.э.     |
| <b>time</b> [(p)]<br>[ без часового пояса ]    | 8 байт  | Время суток (без даты)                  | 00:00:00        | 24:00:00         |
| <b>time</b> [(p)]<br>с часовым поясом          | 12 байт | Только время суток,<br>с часовым поясом | 00:00:00+1459   | 24:00:00-1459    |
| <b>interval</b> [fields ] [(p) ]               | 12 байт | Временной интервал                      | -178000000 лет  | 178000000 лет    |

# Работа со строками

## Оператор LIKE

... WHERE поле\_таблицы [NOT] LIKE шаблон\_строки

Шаблон может включать следующие специальные символы:

% - последовательность любых символов (0 и более символов)

\_ – любой единичный символ

*SELECT \* FROM таблицы  
WHERE поле LIKE 'text%';*

сопоставляется любым строкам, начинающимся на «text»

Для поиска без учета регистра используется оператор **ILIKE**.



# Строковые функции

**CONCAT** – объединяет аргументы в одну строку. Null значения игнорируются.

```
SELECT CONCAT('Post', 'greSQL')
```

## Использование в запросах констант и выражений

```
SELECT pnum || ' продукт ' || weight/1000 from P
```

|| - оператор конкатенации строк

**LENGTH** – возвращает количество символов в строке.

```
SELECT LENGTH('PostgreSQL')  
SELECT LENGTH(name) from prod
```

Подробное описание функций работы со строками приведено в ЛР 2

# Регулярные выражения

*... WHERE table\_field ~ 'pattern'; – с учетом регистра*

*... WHERE table\_field ~\* 'pattern'; – без учета регистра*

Для экранирования специальных символов (., \*, +, ?, [, ], (, ), {, }, |, \) используется одинарный обратный слеш — \. Таблица специальных символов приведена в ЛР 2

```
SELECT * FROM Users WHERE email ~ '@(outlook\.com/mail\.ru)$'
```

Строки, соответствующие запросу:

Vladislav@outlook.com

Ivan@mail.ru

4893-dkdk-dkdk@mail.ru

**Быстрая проверка строки и регулярного выражения:**

```
SELECT 1 WHERE '4893-dkdk-dkdk@mail.ru' ~ '@(outlook\.com/mail\.ru)$'
```

# Функции работы с датами

**CURRENT\_DATE** – возвращает текущую дату.

*SELECT CURRENT\_DATE*

**CURRENT\_TIME** – возвращает текущее время.

*SELECT CURRENT\_TIME*

**NOW** – возвращает текущую дату и время с временной зоной.

*SELECT NOW()*

**DATE** – извлекает часть даты из даты/времени.

*SELECT DATE('2022-12-05 10:37:22')*

Подробное описание функций работы с датами приведено в ЛР 2