

ОСНОВЫ WEB-РАЗРАБОТКИ

Лекция 10. СУБД & SQL

Курс читают:

Шульман В.Д.

@ShtuzerVD

Пелевина Т.В.

@anivelat

Шабанов В.В.

@ZeroHug

Шумилин В.В

@Nodthar1107

ПЛАН ЛЕКЦИИ (СПИСОК ТЕМ)

2

- СУБД
- SQL
- Межсетевое взаимодействие Golang-PostgreSQL

05.11.2024

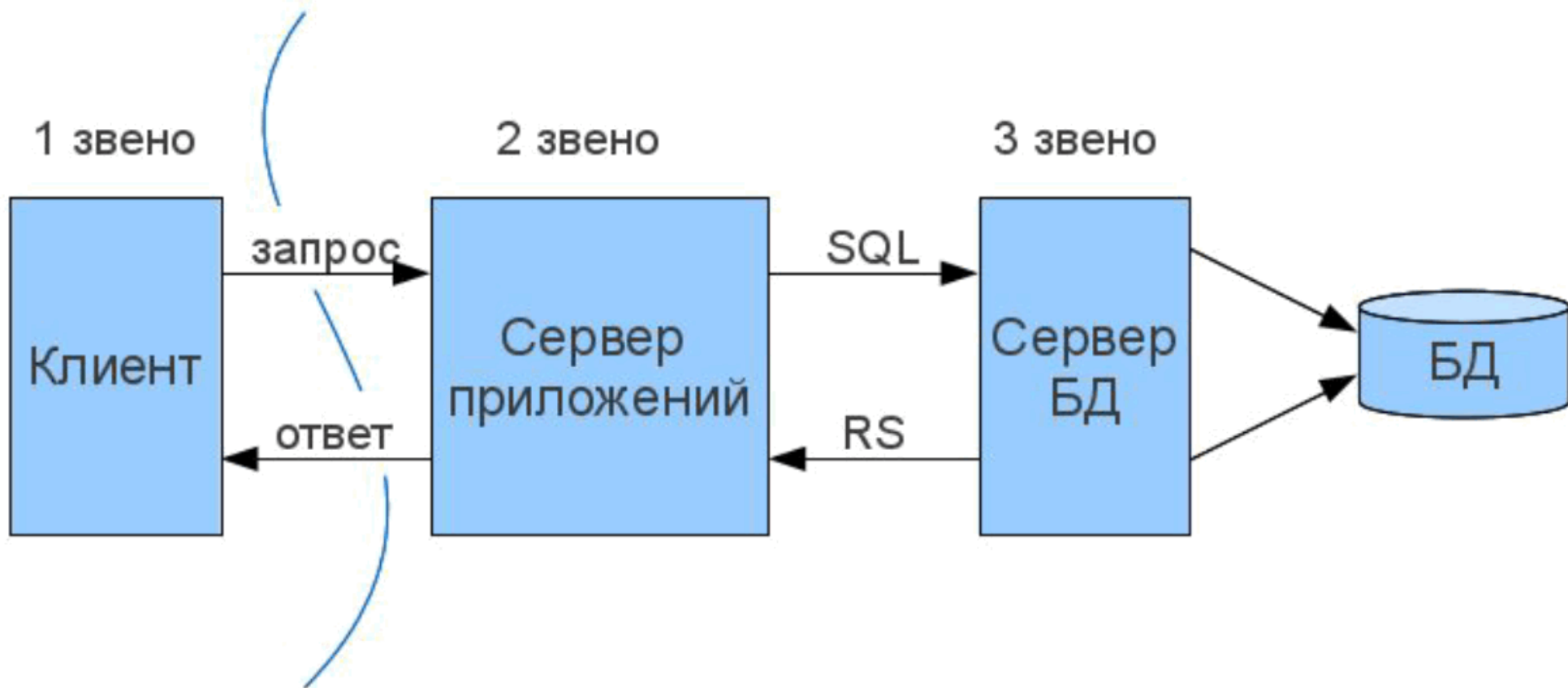
Системы управления базами данных (далее по тексту – СУБД) – набор компонентов, с помощью которого можно создавать, хранить, передавать и управлять базами данных.

База данных (БД) — это упорядоченная совокупность связанных между собой данных

Часто под БД понимаются системы управления базами данных (СУБД). То есть не сам набор информационных сведений, а средства и инструменты для работы с ним.

СУБД В 3-ЗВЕННОЙ АРХИТЕКТУРЕ

4



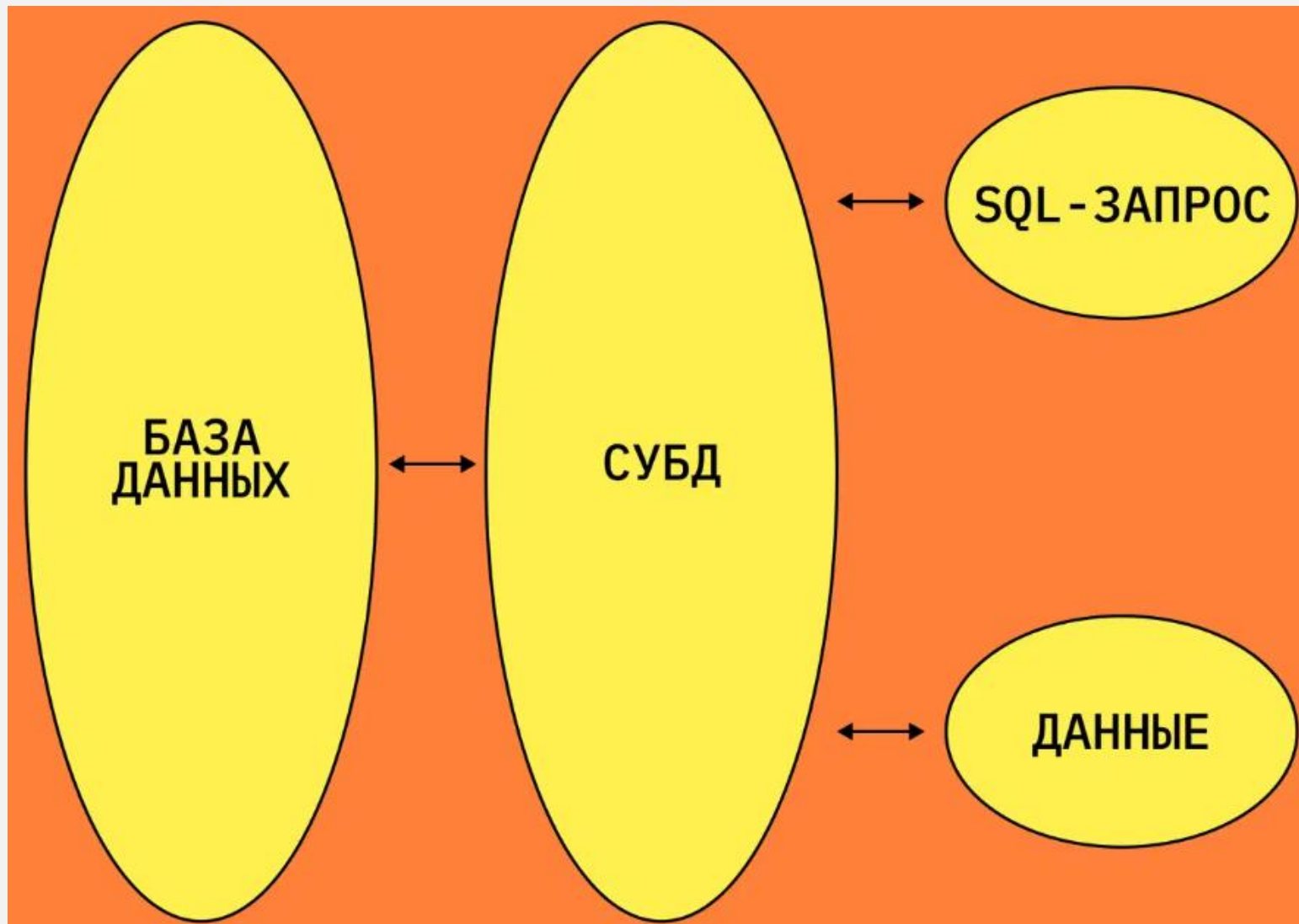
05.11.2024

СУБД В 3-ЗВЕННОЙ АРХИТЕКТУРЕ

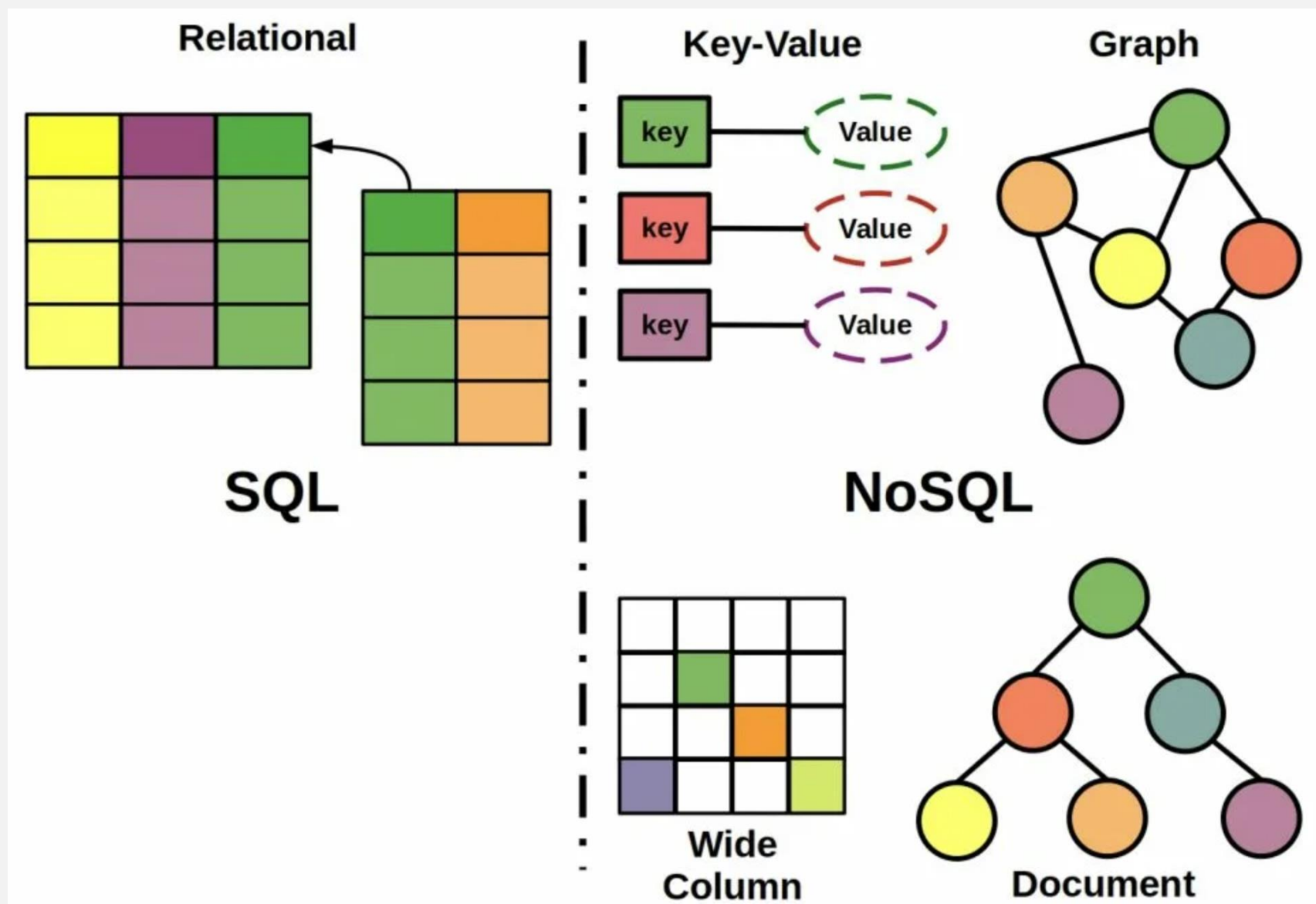
5



05.11.2024



- Создание баз данных, изменение, удаление и объединение их по определённым признакам.
- Хранение данных, в том числе больших массивов, в структурированном виде и нужном формате.
- Защита данных от взлома и нежелательных изменений при помощи распределённого доступа: когда разным группам пользователей доступны разный объём и сегменты данных.
- Выгрузка и сортировка данных по заданным фильтрам при помощи SQL-запросов.
- Поддержка целостности баз данных, резервное копирование и восстановление после сбоев.



СУБД. РАЗНОВИДНОСТИ

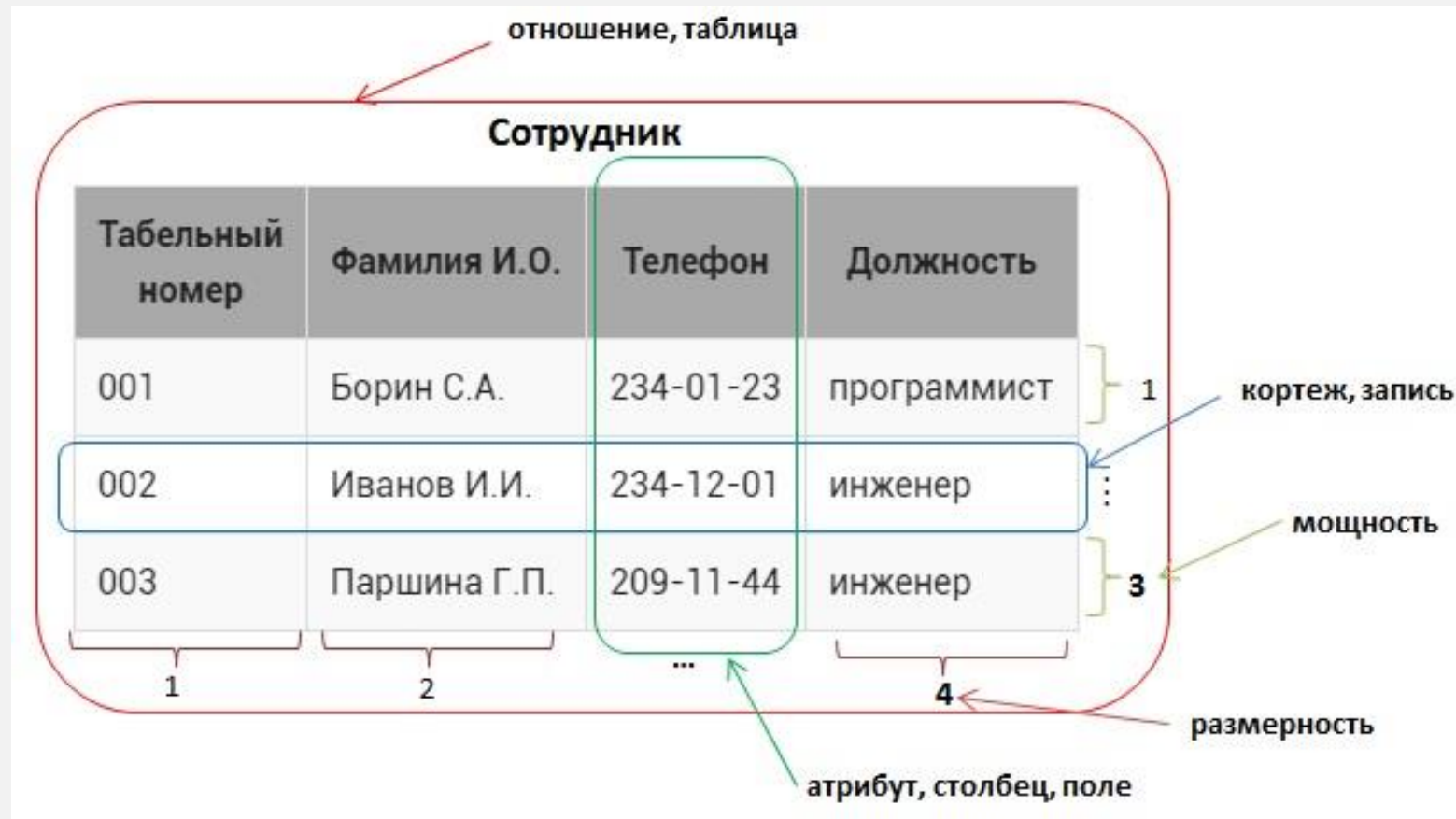
9

Тип СУБД	Когда выбирать	Примеры популярных СУБД
Реляционные	Нужна транзакционность; высокая нормализация; большая доля операций на вставку	Oracle, MySQL, Microsoft SQL Server, PostgreSQL
Ключ-значение	Задачи кэширования и брокеры сообщений	Redis, Memcached
Документные	Для хранения объектов в одной сущности, но с разной структурой; хранение структур на основе JSON	CouchDB, MongoDB, Amazon DocumentDB
Графовые	Задачи подобные социальным сетям; системы оценок и рекомендаций	Neo4j, Amazon Neptune, InfiniteGraph, InfoGrid
Колоночные	Хранилища данных; выборки со сложными аналитическими вычислениями; количество строк в таблице превышает сотни миллионов	Vertica, ClickHouse, Google BigTable, Sybase \ SAP IQ, InfoBright, Cassandra

05.11.2024

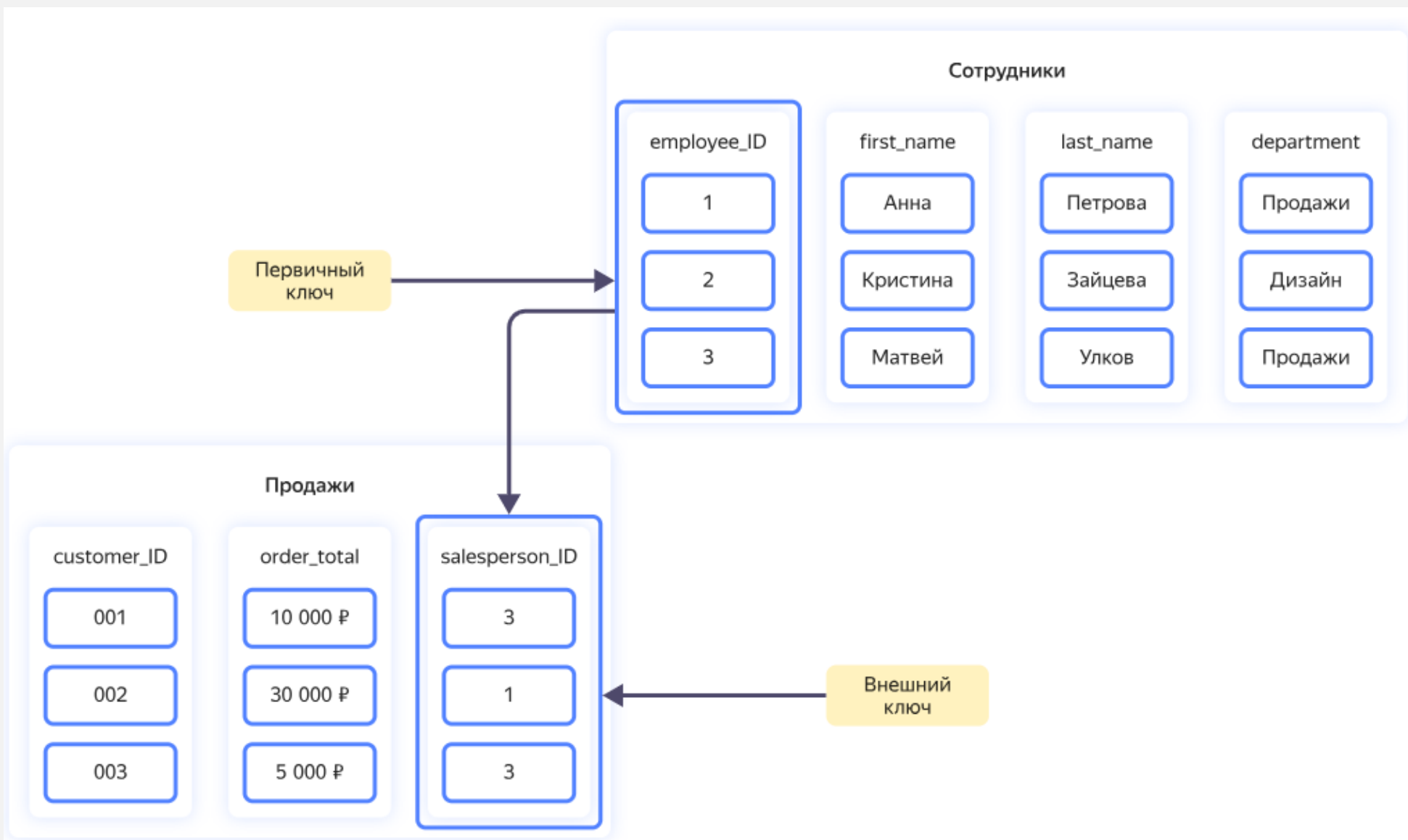
- Данные представлены совокупностью таблиц и связей между ними
- Предоставляют полный CRUD операций над хранимыми данными
- Обеспечивают транзакционность, удовлетворяющую свойствам ACID
- Позволяют обеспечить хранение данных в нормализованном (или не очень) виде





РЕЛЯЦИОННЫЕ СУБД. ТАБЛИЧКИ

12



05.11.2024

Operation	SQL	HTTP
Create	INSERT	PUT / POST
Read (Retrieve)	SELECT	GET
Update (Modify)	UPDATE	PUT / POST / PATCH
Delete (Destroy)	DELETE	DELETE

- **Атомарность.** Гарантирует, что никакая транзакция не будет зафиксирована в СУБД частично. Будут либо выполнены все её подоперации, либо не выполнено ни одной.
- **Согласованность.** Каждая успешная транзакция по определению фиксирует только допустимые результаты.
- **Изоляция.** Во время выполнения транзакции параллельные транзакции не должны оказывать влияния на её результат.
- **Устойчивость.** Независимо от проблем на нижних уровнях (к примеру, обесточивание системы или сбои в оборудовании) изменения, сделанные успешно завершённой транзакцией, должны остаться сохранёнными после возвращения системы в работу.

РЕЛЯЦИОННЫЕ СУБД. НОРМАЛИЗАЦИЯ

15

Фирма	Модели
BMW	M5, X5M, M1
Nissan	GT-R

Нарушение нормализации 1НФ происходит в моделях BMW, т.к. в одной ячейке содержится список из 3 элементов: M5, X5M, M1, т.е. он не является атомарным. Преобразуем таблицу к 1НФ:

Фирма	Модели
BMW	M5
BMW	X5M
BMW	M1
Nissan	GT-R

05.11.2024



PostgreSQL (произносится «Пост-Грес-Кью-Эл») — свободная объектно-реляционная система управления базами данных (СУБД).

SQL (Structured Query Language) — это декларативный язык программирования (язык запросов), который используют для создания, обработки и хранения данных в реляционных базах данных.

Диалекты SQL в различных СУБД имеют некоторые отличия, но в базовых операциях они на 90% совпадают

```
CREATE TABLE genre(  
    genre_id INT PRIMARY KEY AUTO_INCREMENT,  
    name_genre VARCHAR(30)  
);
```

```
1 CREATE TABLE book (  
2     book_id INT PRIMARY KEY AUTO_INCREMENT,  
3     title VARCHAR (50),  
4     author VARCHAR (30),  
5     price DECIMAL (8,2),  
6     amount INT  
7 );  
8
```

```
INSERT INTO таблица(поле1, поле2)  
VALUES (значение1, значение2);
```

genre_id	name_genre
INT PRIMARY KEY AUTO_INCREMENT	VARCHAR(30)
1	Роман

Запрос:

```
INSERT INTO genre (name_genre)  
VALUES ('Роман');
```

```
SELECT * FROM genre;
```

Результат:

Affected rows: 1

Query result:

genre_id	name_genre
1	Роман

Affected rows: 1

[Свернуть](#)

book_id	title	author	price	amount
INT PRIMARY KEY AUTO_INCREMENT	VARCHAR(50)	VARCHAR(30)	DECIMAL(8,2)	INT
1	Мастер и Маргарита	Булгаков М.А.	670.99	3

Affected rows: 1

Query result:

book_id	title	author	price	amount
1	Мастер и Маргарита	Булгаков М.А.	670.99	3

Affected rows: 1

[Свернуть](#)

```
1 INSERT INTO book (title, author, price, amount)
2 VALUES ("Мастер и Маргарита", "Булгаков М.А.", 670.99, 3);
3 SELECT * FROM book;
4
```

```
SELECT title, amount FROM book;
```

Результат:

title	amount
Мастер и Маргарита	3
Белая гвардия	5
Идиот	10
Братья Карамазовы	2
Стихотворения и поэмы	15

```
SELECT title AS Название, amount  
FROM book;
```

Результат:

Название	amount
Мастер и Маргарита	3
Белая гвардия	5
Идиот	10
Братья Карамазовы	2
Стихотворения и поэмы	15

```
SELECT title, author, price, amount,  
       price * amount AS total  
FROM book;
```

Результат:

title	author	price	amount	total
Мастер и Маргарита	Булгаков М.А.	670.99	3	2012.97
Белая гвардия	Булгаков М.А.	540.50	5	2702.50
Идиот	Достоевский Ф.М.	460.00	10	4600.00
Братья Карамазовы	Достоевский Ф.М.	799.01	2	1598.02
Стихотворения и поэмы	Есенин С.А.	650.00	15	9750.00

Запрос:

```
SELECT title, price  
FROM book  
WHERE price < 600;
```

Результат:

+	-	-	-	-	+	-	-	-	-	+
	title					price				
+	-	-	-	-	+	-	-	-	-	+
	Белая гвардия					540.50				
	Идиот					460.00				
+	-	-	-	-	+	-	-	-	-	+

Запрос:

```
SELECT title, author, price * amount AS total  
FROM book  
WHERE price * amount > 4000;
```

Результат:

title	author	total
Идиот	Достоевский Ф.М.	4600.00
Стихотворения и поэмы	Есенин С.А.	9750.00

Запрос:

```
SELECT title, author, price
FROM book
WHERE (author = 'Булгаков М.А.' OR author = 'Есенин С.А.') AND price > 600;
```

Результат:

title	author	price
Мастер и Маргарита	Булгаков М.А.	670.99
Стихотворения и поэмы	Есенин С.А.	650.00

Запрос:

```
SELECT title, author, price
FROM book
WHERE author = 'Булгаков М.А.' OR author = 'Есенин С.А.' AND price > 600;
```

Результат (сравните с предыдущим):

title	author	price
Мастер и Маргарита	Булгаков М.А.	670.99
Белая гвардия	Булгаков М.А.	540.50
Стихотворения и поэмы	Есенин С.А.	650.00

Запрос:

```
SELECT title, amount
FROM book
WHERE amount BETWEEN 5 AND 14;
```

Результат:

+	-----+	-----+
	title	amount
+	-----+	-----+
	Белая гвардия	5
	Идиот	10
+	-----+	-----+

Этот запрос можно реализовать по-другому, результат будет точно такой же.

```
SELECT title, amount
FROM book
WHERE amount >= 5 AND amount <=14;
```

Запрос:

```
SELECT title, price
FROM book
WHERE author IN ('Булгаков М.А.', 'Достоевский Ф.М.');
```

Результат:

title	price
Мастер и Маргарита	670.99
Белая гвардия	540.50
Идиот	460.00
Братья Карамазовы	799.01

Этот запрос можно реализовать по-другому, результат будет точно такой же.

```
SELECT title, price
FROM book
WHERE author = 'Булгаков М.А.' OR author = 'Достоевский Ф.М.';
```

Запрос:

```
SELECT title, author, price  
FROM book  
ORDER BY title;
```

Результат:

title	author	price
Белая гвардия	Булгаков М.А.	540.50
Братья Карамазовы	Достоевский Ф.М.	799.01
Идиот	Достоевский Ф.М.	460.00
Мастер и Маргарита	Булгаков М.А.	670.99
Стихотворения и поэмы	Есенин С.А.	650.00

Запрос:

```
SELECT author, title, amount AS Количество
FROM book
WHERE price < 750
ORDER BY author, amount DESC;
```

Результат:

author	title	Количество
Булгаков М.А.	Белая гвардия	5
Булгаков М.А.	Мастер и Маргарита	3
Достоевский Ф.М.	Идиот	10
Есенин С.А.	Стихотворения и поэмы	15

Вывести названия книг, начинающихся с буквы «Б».

Запрос:

```
SELECT title
FROM book
WHERE title LIKE 'Б%';
/* эквивалентное условие
title LIKE 'Б%'
*/
```

Результат:

title
Белая гвардия
Братья Карамазовы

Запрос:

```
SELECT title FROM book  
WHERE title LIKE "_____"
```

Результат:

```
+-----+  
| title |  
+-----+  
| Идиот |  
| Поэмы |  
+-----+
```


Запрос:

```
SELECT title FROM book
WHERE  title LIKE "_% и %" /*отбирает слово И внутри названия */
      OR title LIKE "и %" /*отбирает слово И в начале названия */
      OR title LIKE "% и" /*отбирает слово И в конце названия */
      OR title LIKE "и" /* отбирает название, состоящее из одного слова И */
```

Результат:

title
Мастер и Маргарита
Стихотворения и поэмы

Выбрать различных авторов, книги которых хранятся в таблице **book** .

Запрос:

```
SELECT DISTINCT author  
FROM book;
```

Результат:

+-----+
author
+-----+
Булгаков М.А.
Достоевский Ф.М.
Есенин С.А.
+-----+

```
SELECT author, sum(amount)
FROM book
GROUP BY author;
```

```
+-----+-----+
| author          | SUM(amount) |
+-----+-----+
| Булгаков М.А.   | 8           |
| Достоевский Ф.М. | 23          |
| Есенин С.А.     | 15          |
+-----+-----+
```

book_id	title	author	price	amount
1	Мастер и Маргарита	Булгаков М.А.	670.99	3
2	Белая гвардия	Булгаков М.А.	540.50	5
3	Идиот	Достоевский Ф.М.	460.00	10
4	Братья Карамазовы	Достоевский Ф.М.	799.01	3
5	Игрок	Достоевский Ф.М.	480.50	10
6	Стихотворения и поэмы	Есенин С.А.	650.00	15

$$3 + 5 = 8$$

$$10 + 3 + 10 = 23$$

$$15 = 15$$

author	SUM(amount)
Булгаков М.А.	8
Достоевский Ф.М.	23
Есенин С.А.	15

```
SELECT author, COUNT(author), COUNT(amount), COUNT(*)  
FROM book  
GROUP BY author;
```

Результат:

author	COUNT(author)	COUNT(amount)	COUNT(*)
Булгаков М.А.	2	2	2
Достоевский Ф.М.	3	3	3
Есенин С.А.	2	1	2

Запрос:

```
SELECT SUM(amount) AS Количество  
FROM book;
```

Результат:

+	-	-	-	-	-	+
	Количество					
+	-	-	-	-	-	+
	46					
+	-	-	-	-	-	+

```
SELECT author,  
       MIN(price) AS Минимальная_цена,  
       MAX(price) AS Максимальная_цена  
FROM book  
GROUP BY author  
HAVING SUM(price * amount) > 5000;
```

Результат:

author	Минимальная_цена	Максимальная_цена
Достоевский Ф.М.	460.00	799.01
Есенин С.А.	650.00	650.00

Запрос:

```
SELECT title, author, price, amount
FROM book
WHERE price = (
    SELECT MIN(price)
    FROM book
);
```

Результат:

+	-----+	-----+	-----+	-----+
	title	author	price	amount
+	-----+	-----+	-----+	-----+
	Идиот	Достоевский Ф.М.	460.00	10
+	-----+	-----+	-----+	-----+

Запрос:

```
SELECT title, author, amount, price
FROM book
WHERE author IN (
    SELECT author
    FROM book
    GROUP BY author
    HAVING SUM(amount) >= 12
);
```

Результат:

title	author	amount	price
Идиот	Достоевский Ф.М.	10	460.00
Братья Карамазовы	Достоевский Ф.М.	3	799.01
Игрок	Достоевский Ф.М.	10	480.50
Стихотворения и поэмы	Есенин С.А.	15	650.00

Запрос:

```
SELECT title, author, amount, price
FROM book
WHERE amount < ALL (
    SELECT AVG(amount)
    FROM book
    GROUP BY author
);
```

Результат:

title	author	amount	price
Мастер и Маргарита	Булгаков М.А.	3	670.99
Братья Карамазовы	Достоевский Ф.М.	3	799.01

Запрос:

```
SELECT title, author, amount,  
       (  
         SELECT AVG(amount)  
         FROM book  
       ) AS Среднее_количество  
FROM book  
WHERE abs(amount - (SELECT AVG(amount) FROM book)) >3;
```

title	author	amount	Среднее_количество
Мастер и Маргарита	Булгаков М.А.	3	7.6667
Братья Карамазовы	Достоевский Ф.М.	3	7.6667
Стихотворения и поэмы	Есенин С.А.	15	7.6667

UPDATE таблица SET поле = выражение

Запрос:

```
UPDATE book
SET price = 0.7 * price;

SELECT * FROM book;
```

Результат:

Affected rows: 5

Query result:

book_id	title	author	price	amount
1	Мастер и Маргарита	Булгаков М.А.	469.69	3
2	Белая гвардия	Булгаков М.А.	378.35	5
3	Идиот	Достоевский Ф.М.	322.00	10
4	Братья Карамазовы	Достоевский Ф.М.	559.31	2
5	Стихотворения и поэмы	Есенин С.А.	455.00	15

```
UPDATE book, supply
SET book.amount = book.amount + supply.amount
WHERE book.title = supply.title AND book.author = supply.author;

SELECT * FROM book;
```

Результат:

Affected rows: 2

Query result:

book_id	title	author	price	amount
1	Мастер и Маргарита	Булгаков М.А.	670.99	3
2	Белая гвардия	Булгаков М.А.	540.50	12
3	Идиот	Достоевский Ф.М.	460.00	13
4	Братья Карамазовы	Достоевский Ф.М.	799.01	2
5	Стихотворения и поэмы	Есенин С.А.	650.00	15

```
DELETE FROM таблица;
```

Запрос:

```
DELETE FROM supply;  
  
SELECT * FROM supply;
```

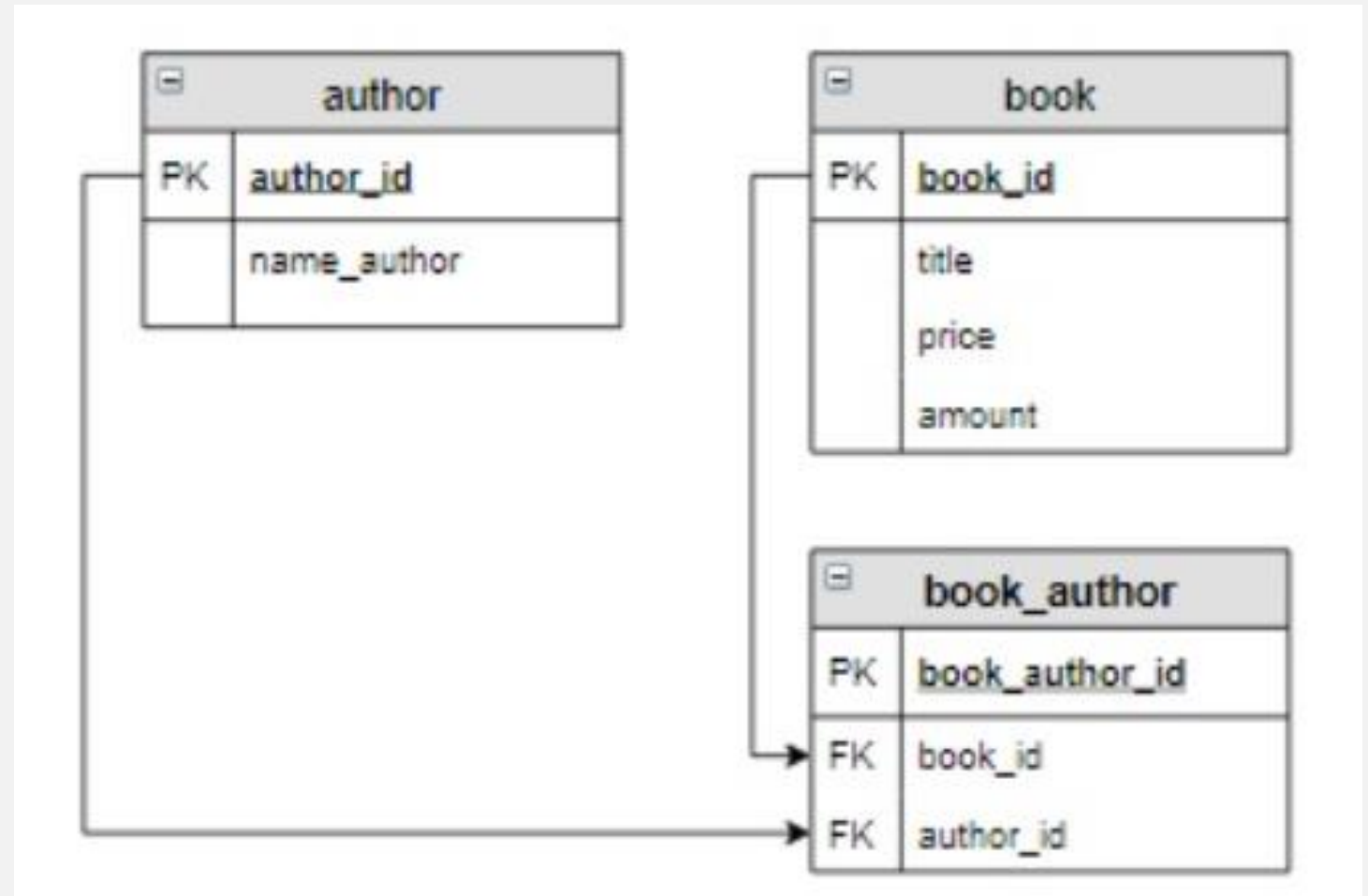
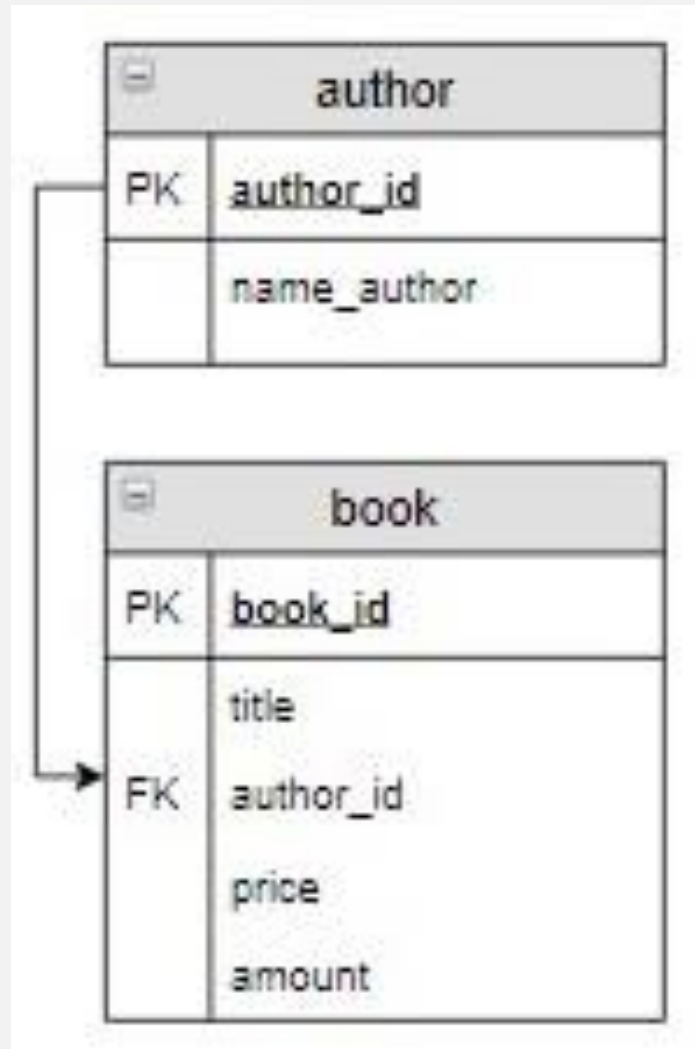
Результат:

```
Affected rows: 4  
Affected rows: 0
```

```
DELETE FROM таблица  
WHERE условие;
```

```
DELETE FROM supply  
WHERE title IN (  
    SELECT title  
    FROM book  
);
```

```
SELECT * FROM supply;
```



```
SELECT title, name_author
FROM
    author INNER JOIN book
    ON author.author_id = book.author_id;
```

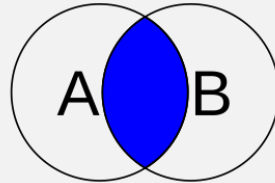
title	name_author
Мастер и Маргарита	Булгаков М.А.
Белая гвардия	Булгаков М.А.
Идиот	Достоевский Ф.М.
Братья Карамазовы	Достоевский Ф.М.
Игрок	Достоевский Ф.М.
Стихотворения и поэмы	Есенин С.А.
Черный человек	Есенин С.А.
Лирика	Пастернак Б.Л.

```
SELECT name_author, title
FROM author LEFT JOIN book
      ON author.author_id = book.author_id
ORDER BY name_author;
```

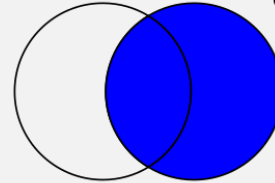
Результат:

name_author	title
Булгаков М.А.	Мастер и Маргарита
Булгаков М.А.	Белая гвардия
Достоевский Ф.М.	Игрок
Достоевский Ф.М.	Идиот
Достоевский Ф.М.	Братья Карамазовы
Есенин С.А.	Стихотворения и поэмы
Есенин С.А.	Черный человек
Лермонтов М.Ю.	NULL
Пастернак Б.Л.	Лирика

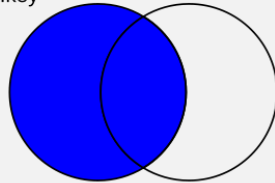

```
SELECT <fields>
FROM TableA A
INNER JOIN TableB B
ON A.key = B.key
```



```
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
```



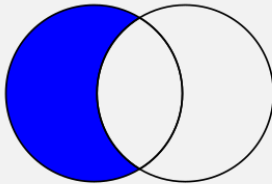
```
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
```



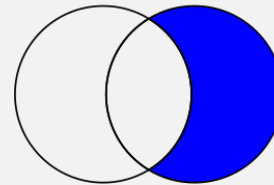
SQL

JOINS

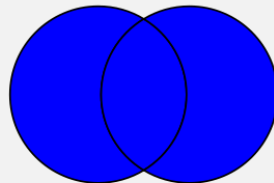
```
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
WHERE B.key IS NULL
```



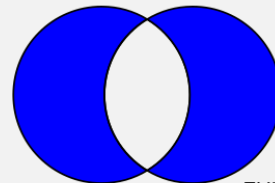
```
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL
```



```
SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key
```



```
SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key
```



```
1 package main
2
3 import (
4     "database/sql"
5     "fmt"
6     _ "github.com/lib/pq"
7 )
8
9 const (
10     host      = "localhost"
11     port      = 5432
12     user      = "postgres"
13     password  = "rusho1234"
14     dbname    = "MyDB"
15 )
16
17 func main() {
18     psqlconn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s sslmode=disable", host, port, user, password, dbname)
19
20     db, err := sql.Open("postgres", psqlconn)
21     CheckError(err)
22
23     defer db.Close()
24 }
```

```
25 // insert
26 // hardcoded
27 insertStmt := `insert into "Students"("Name", "Roll") values('John', 1)`
28 _, e := db.Exec(insertStmt)
29 CheckError(e)
30
31 // dynamic
32 insertDynStmt := `insert into "Students"("Name", "Roll") values($1, $2)`
33 _, e = db.Exec(insertDynStmt, "Jane", 2)
34 CheckError(e)
35 }
```

```
36 // update
37 updateStmt := `update "Students" set "Name"=$1, "Roll"=$2 where "id"=$3`
38 _, e = db.Exec(updateStmt, "Mary", 3, 2)
39 CheckError(e)
40
41 // Delete
42 deleteStmt := `delete from "Students" where id=$1`
43 _, e = db.Exec(deleteStmt, 1)
44 CheckError(e)
45
```

```
47 rows, err := db.Query(`SELECT "Name", "Roll" FROM "Students"`)
48 CheckError(err)
49
50 defer rows.Close()
51 for rows.Next() {
52     var name string
53     var roll int
54
55     err = rows.Scan(&name, &roll)
56     CheckError(err)
57
58     fmt.Println(name, roll)
59 }
60
61 CheckError(err)
```

```
63  var name string
64  err = db.QueryRow("select name from users where id = ?", 1).Scan(&name)
65  CheckError(e)
```

```
import (  
    "gorm.io/gorm"  
    "gorm.io/driver/postgres"  
)  
  
func main() {  
    dsn := "host=localhost user=user dbname=db password=password sslmode=disable"  
    db, err := gorm.Open(postgres.Open(dsn), &gorm.Config{})  
}
```

```
type User struct {  
    gorm.Model  
    Name    string `gorm:"size:255"`  
    Email    string `gorm:"type:varchar(100);unique_index"`  
    Age      int    `gorm:"default:18"`  
}
```



```
type User struct {  
    gorm.Model  
    CreditCards []CreditCard `gorm:"foreignKey:UserID"`  
}  
  
type CreditCard struct {  
    gorm.Model  
    Number string  
    UserID uint  
}
```

```
user := User{Name: "John Doe", Age: 30}
result := db.Create(&user) // создает новую запись в базе данных

if result.Error != nil {
    // обработка ошибки создания
}
```

```
db.Model(&User{}).Where("active = ?", true).Updates(map[string]interface{}{"active": false})
```

```
tx := db.Begin()
defer func() {
    if r := recover(); r != nil {
        tx.Rollback()
    }
}()

if tx.Error != nil {
    return // Возвращает ошибку при неудачной инициализации транзакции
}

// Последовательность операций
if err := tx.Create(&newUser).Error; err != nil {
    tx.Rollback()
    return
}

if err := tx.Update(&existingUser).Error; err != nil {
    tx.Rollback()
    return
}

tx.Commit()
```

- Какую СУБД выбрать:
<https://habr.com/ru/articles/579248/>
- Сравнение SQL- и NoSQL-баз данных:
<https://habr.com/ru/companies/ruvds/articles/727474/>
- Основы PostgreSQL для начинающих:
<https://tproger.ru/articles/osnovy-postgresql-dlya-nachinayushhih--ot-ustanovki-do-pervyh-zaprosov-25085/>
- Интерактивный тренажёр по SQL:
<https://stepik.org/course/63054/info>

СПАСИБО ЗА ВНИМАНИЕ :3

05.11.2024