



Java Levelup #4

Hello, World

Проекты



DB



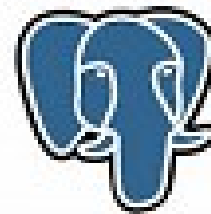
Реляционные базы данных

- Популярные СУБД (Системы Управления Базами Данных)

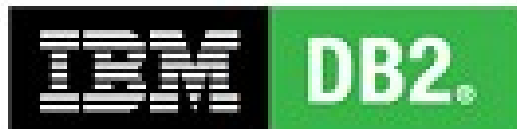


ORACLE®

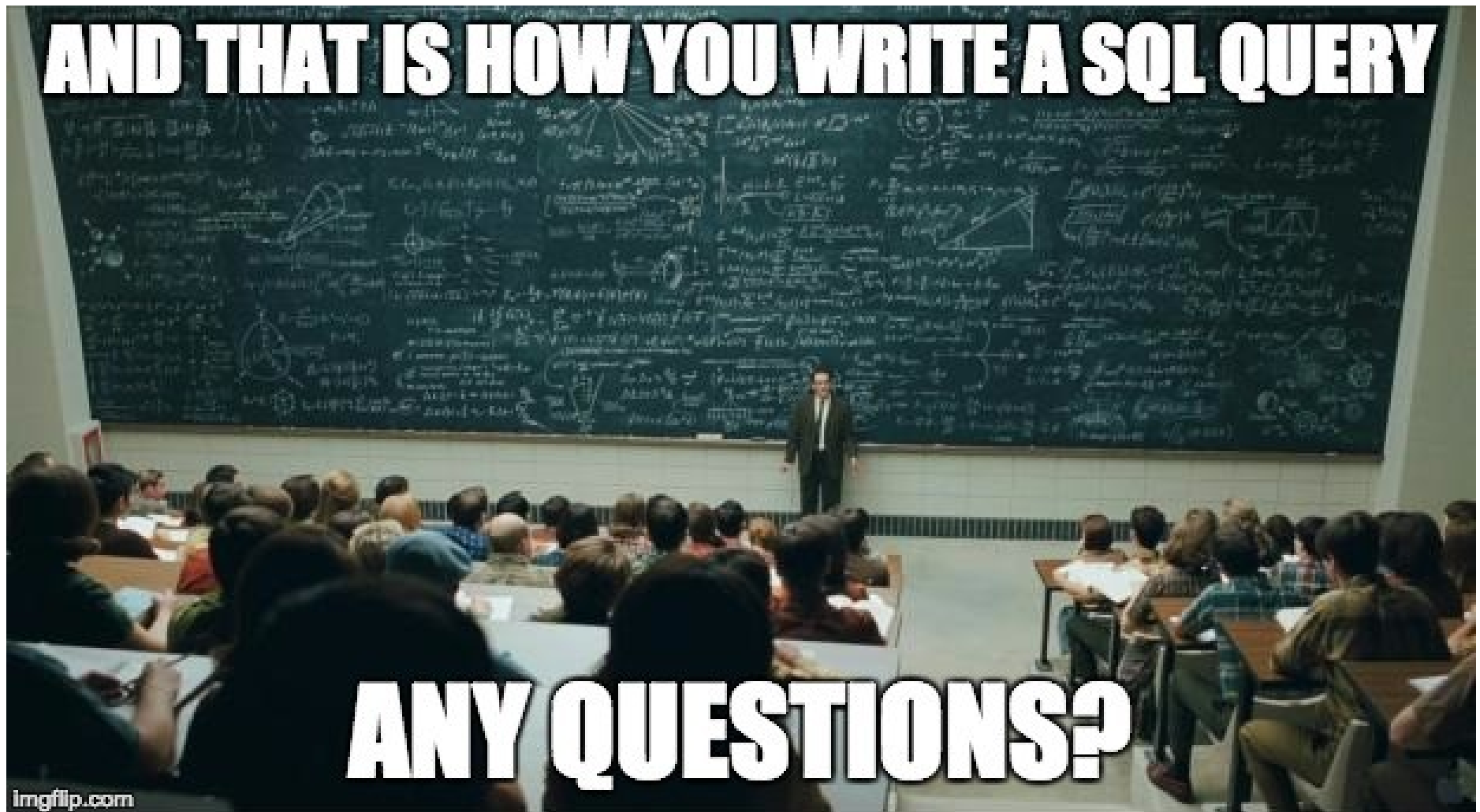
PostgreSQL



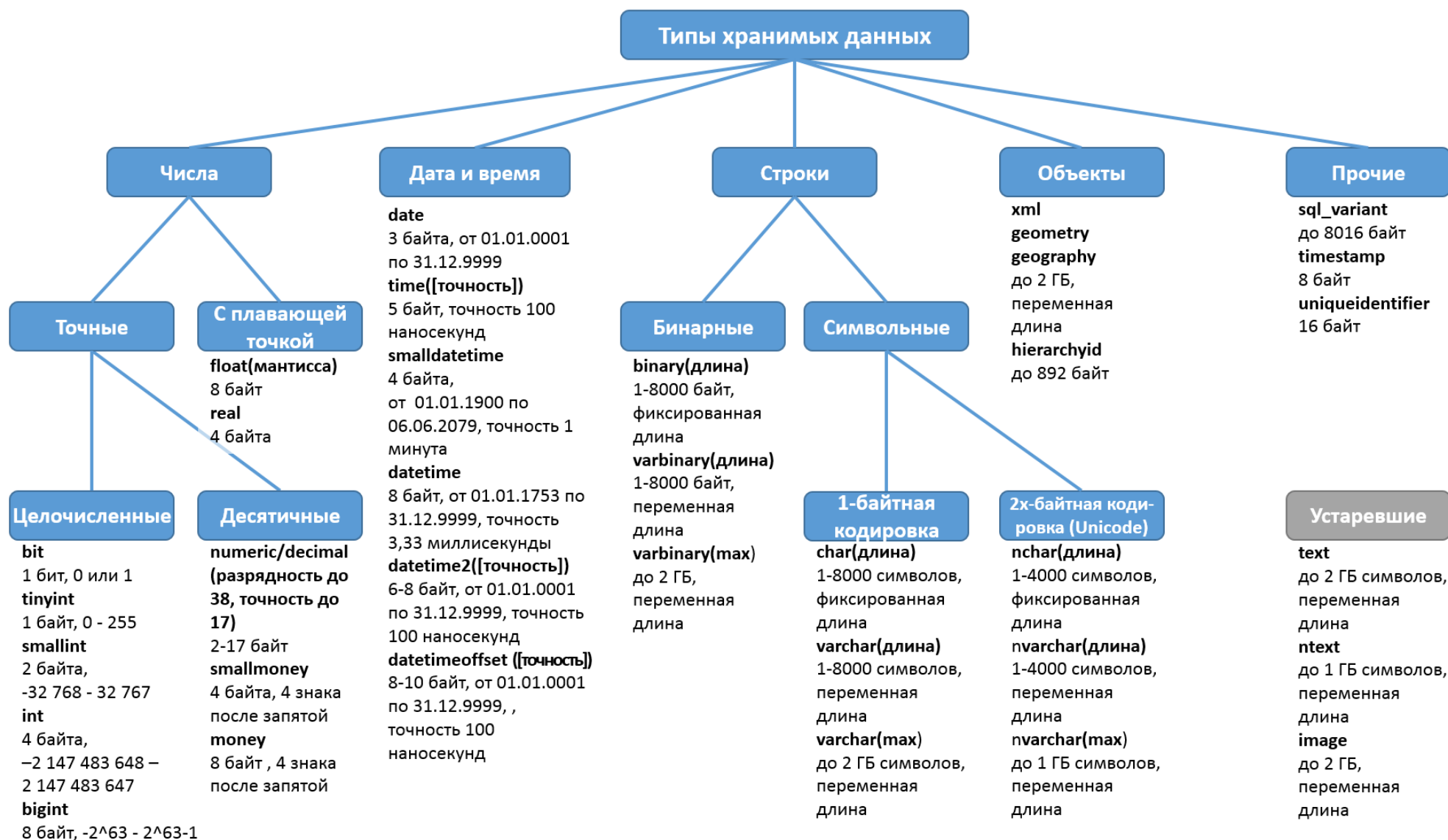
SYBASE®



SQL — Язык запросов



MySQL — типы данных



Пример таблицы

	id	first_name	last_name	dept_name
▶	1	John	Doe	Sales
	2	Bush	Lily	Marketing
	3	David	Dave	Finance
	4	Mary	Jane	Accounting
	5	Jonatha	Josh	Warehouses
	6	Mateo	More	Sales

Создание таблицы

```
CREATE TABLE table_name (  
    column1 datatype constraint,  
    column2 datatype constraint,  
    column3 datatype constraint,  
    ....  
);
```

Statement 1

```
1 CREATE TABLE test_results  
2 (  
3     name                TEXT,  
4     student_id          INTEGER,  
5     birth_date          DATE,  
6     test_result         DECIMAL,  
7     grade               TEXT,  
8     passed              BOOLEAN  
9 );
```

Table test_results created

Execution time: 0.12s

Statement 1 of 1 finished

Основные Ограничения (Constraints)

- NOT NULL — Низя NULL
- UNIQUE - Гарантирует, что все значения в столбце уникальные
- PRIMARY KEY - Уникально идентифицирует каждую строку в таблице. == NOT NULL && UNIQUE
- FOREIGN KEY - Уникально идентифицирует строку/запись в другой таблице
- CHECK - Гарантирует, что все значения в столбце удовлетворяют определенному условию
- DEFAULT - Устанавливает значение по умолчанию для столбца, когда значение не указано.

Добавление записи в таблицу

- `INSERT INTO table_name (column1, column2, column3, ...)`
`VALUES (value1, value2, value3, ...);`

Обновление данных в таблице

- UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;

Выборка из таблицы

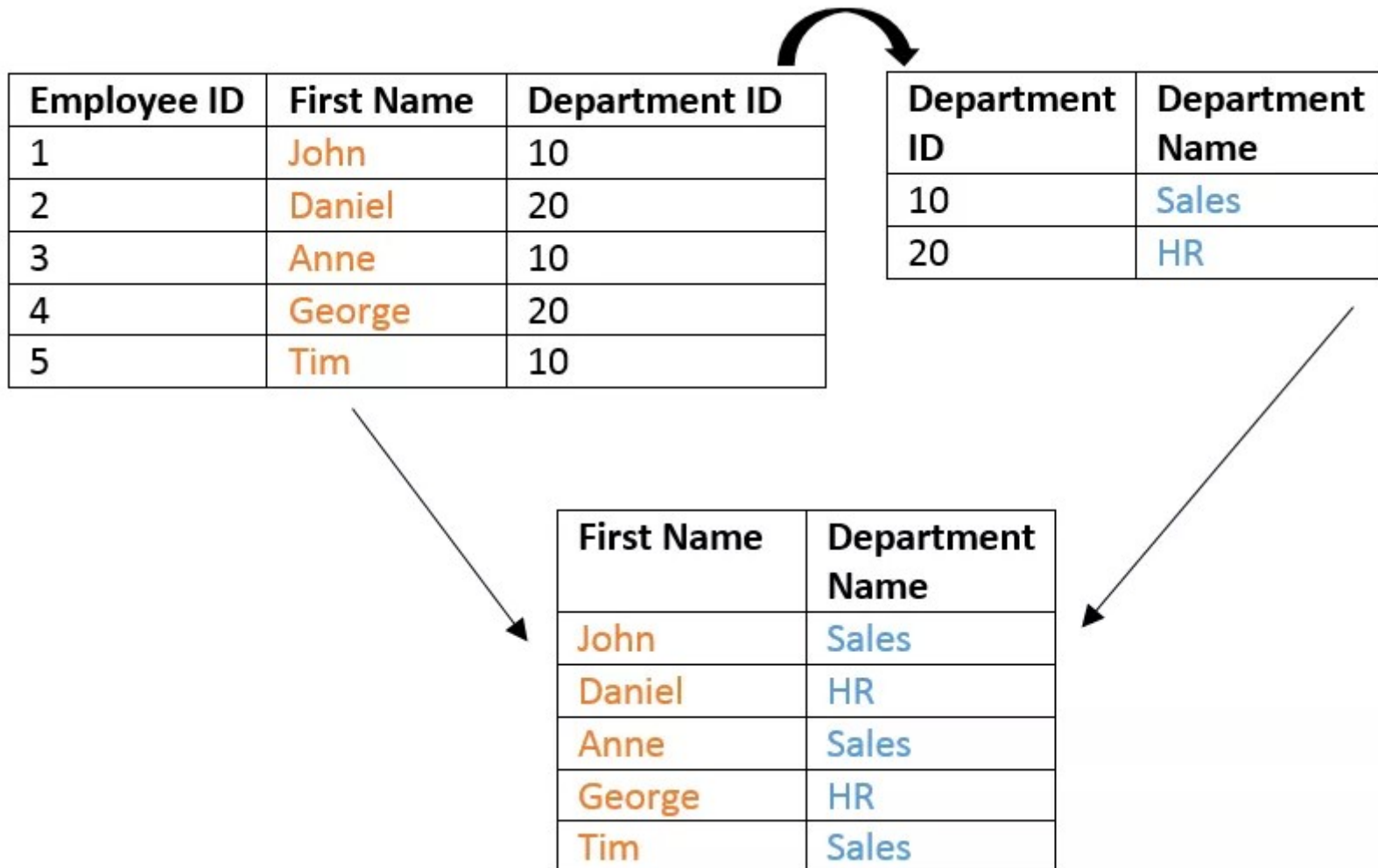
- `SELECT column1, column2, ...
FROM table_name;`
- Возможно упорядочивание:
 - `ORDER BY`
- Число строк:
 - `LIMIT N;`



Удаление из таблицы

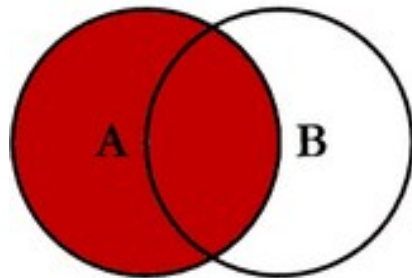
- `DELETE FROM table_name WHERE condition;`

JOIN

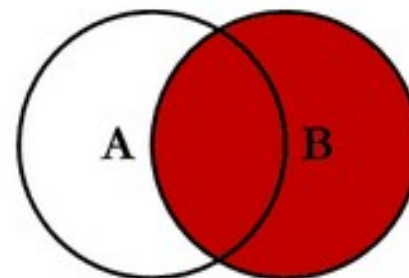


JOIN

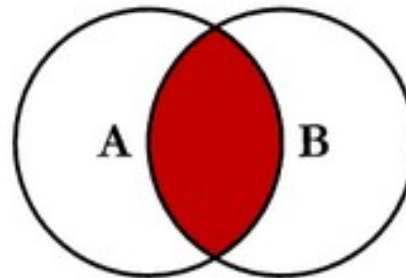
SQL JOINS



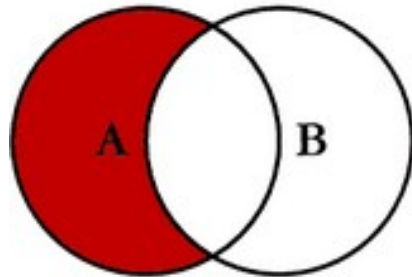
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



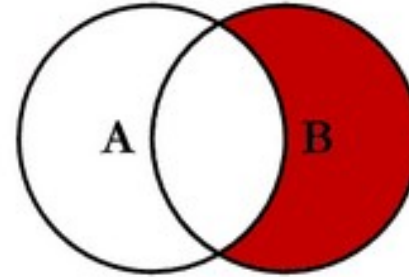
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



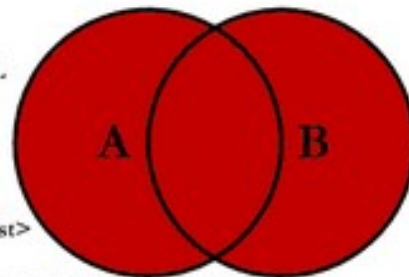
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



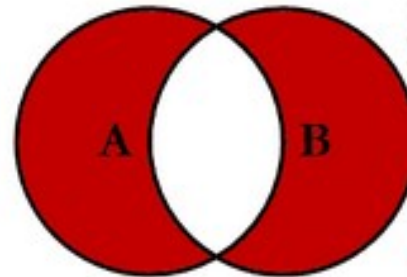
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



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



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

GROUP BY

```
SELECT working_area, COUNT(*)  
FROM agents  
GROUP BY working_area;
```

agents

AGENT_NAME	WORKING_AREA
Alex	London
Subbarao	Bangalore
Benjamin	Hampshair
Ramasundar	Bangalore
Alford	New York
Ravi Kumar	Bangalore
Santakumar	Chennai
Lucida	San Jose
Anderson	Brisban
Mukesh	Mumbai
McDen	London
Ivan	Torento

the working_area have
been grouped and appearing
once

WORKING_AREA	COUNT(*)
San Jose	1
Torento	1
London	2
Hampshair	1
New York	1
Brisban	1
Bangalore	3
Chennai	1
Mumbai	1

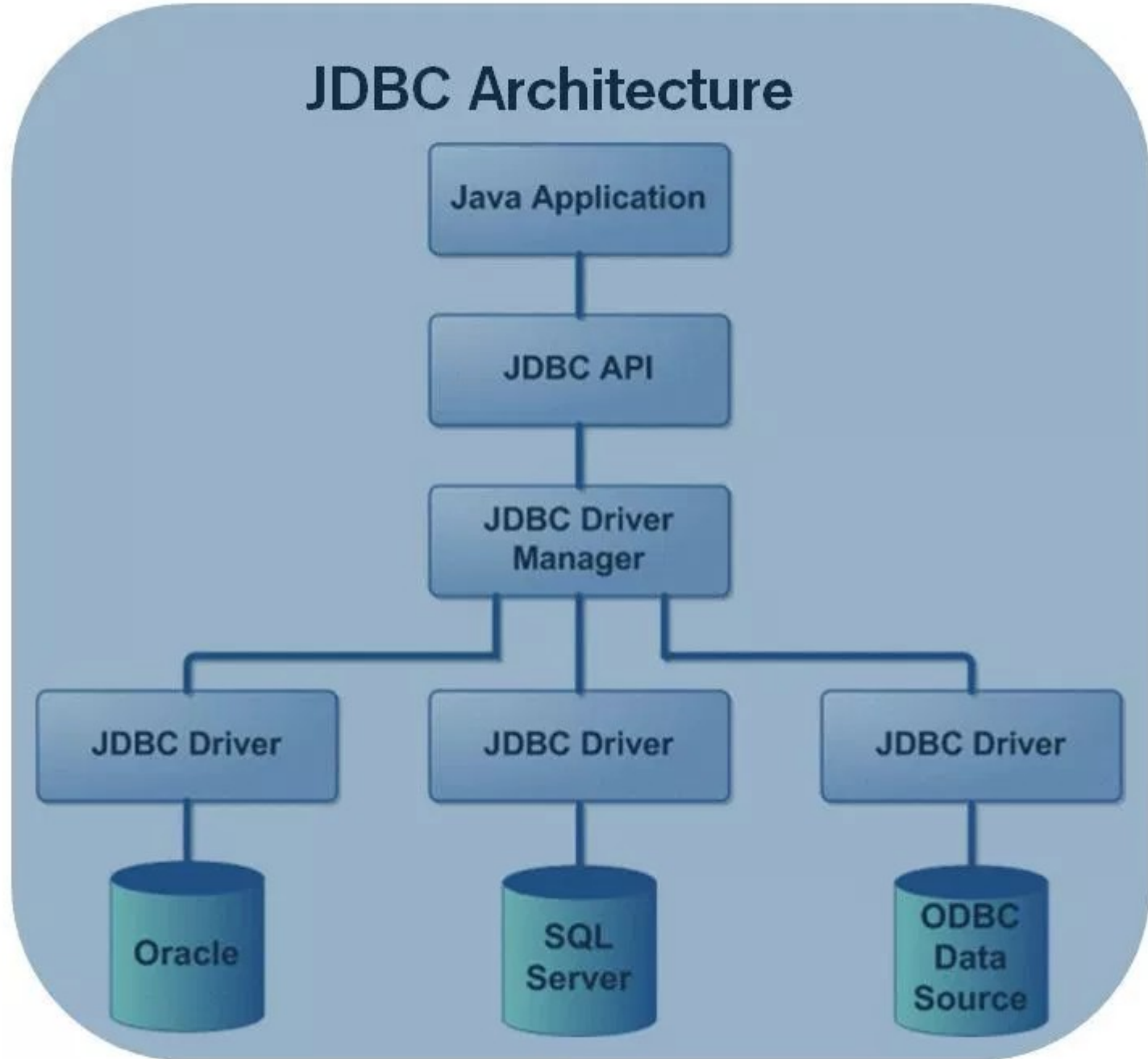
result



SQL

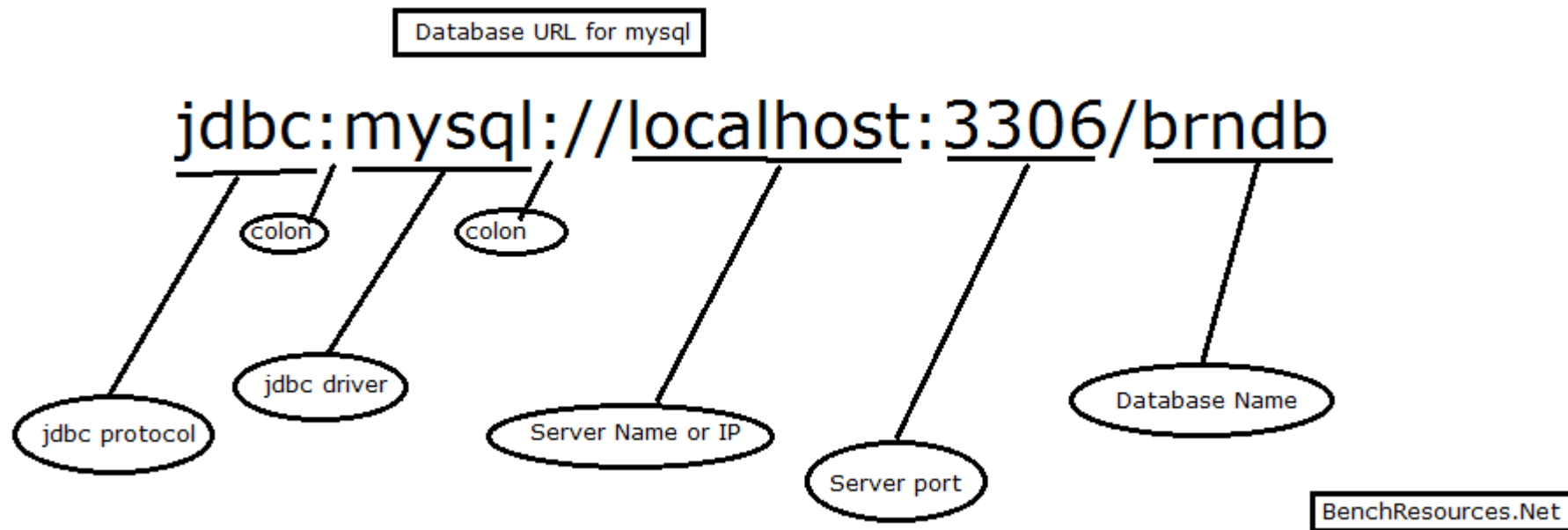
- Про SQL можно рассказывать еще очень много
- Подробнее — в гугле ;)

JDBC



Подключение к БД

```
1 Class.forName(driverClass);  
2 Connection connection = DriverManager  
3     .getConnection(url, user, password) ;
```



Statement и ResultSet

```
1 Statement statement = connection.createStatement();
2 statement.execute("CREATE TABLE user(" +
3     "id INT PRIMARY KEY AUTO_INCREMENT, " +
4     "name varchar(100));");
5
6
7 statement.execute("INSERT INTO user(name) values('borya'),('petya')");
8 ResultSet rs = statement.executeQuery("SELECT * FROM user");
9 while (rs.next()) {
10     System.out.println(rs.getInt("id") + " : " + rs.getString("name"));
11 }
```

STATEMENT НЕОБХОДИМО ЗАКРЫВАТЬ!

A ResultSet object is automatically closed by the Statement object that generated it when that Statement object is closed, re-executed, or is used to retrieve the next result from a sequence of multiple results.

SQL-инъекции

```
1 PreparedStatement statement = connection  
2     .prepareStatement("insert into user(id,name)" +  
3     "values(" + id + "," + name + ")");
```

name = "); DROP TABLE user; -- "



Подстановка в запрос

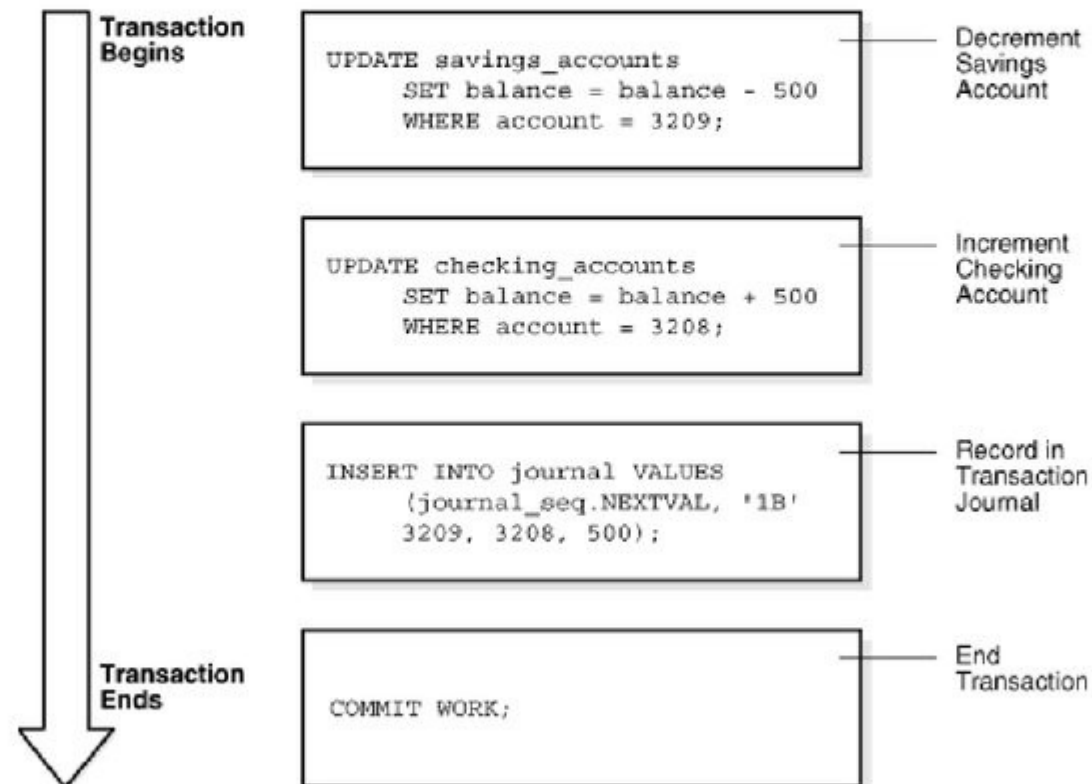
```
1 PreparedStatement statement = connection
2     .prepareStatement("INSERT INTO user(id,name) VALUES(?,?)");
3 statement.setInt(1, 3);
4 statement.setString(2, "Kek");
5 statement.executeUpdate();
```



Транзакции



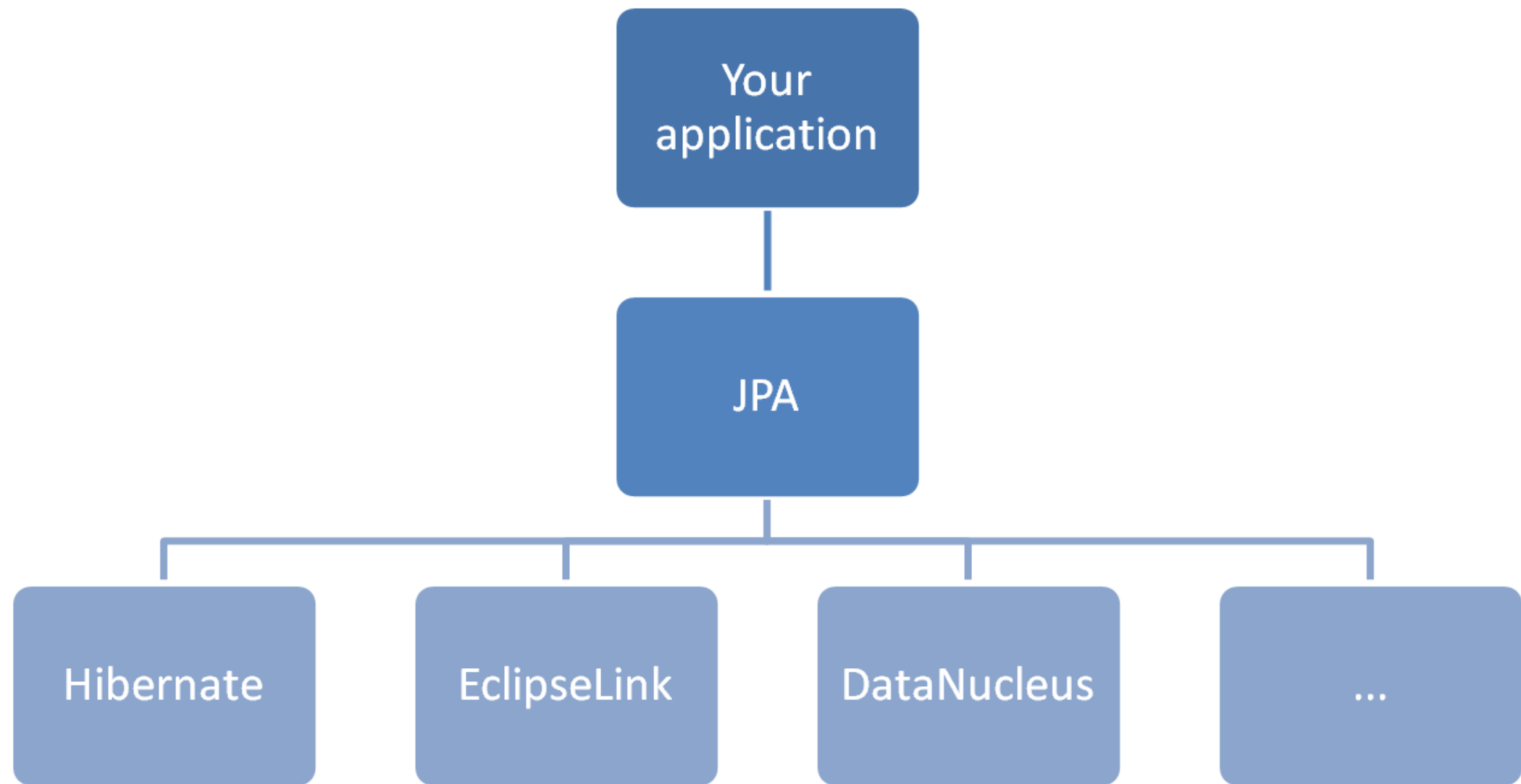
Пример(банковская транзакция)



Транзакции

```
1 connection.setAutoCommit(false);
2
3 Statement st = connection.createStatement();
4 try {
5     st.execute("INSERT INTO user(name) VALUES('Mem')");
6     connection.commit();
7 } catch (SQLException e) {
8     connection.rollback();
9 }
```

JPA



Hibernate

```
@Entity
@Table (name = "users")
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    @Column(name = "name")
    private String name;
    //можно не указывать Column name, если оно совпадает с названием столбца в таблице
    private int age;

    @OneToMany(mappedBy = "user", cascade = CascadeType.ALL, orphanRemoval = true)
    private List<Auto> autos;

    public User() {
    }
}
```



ДЗ

- Начать выполнять свои командные задачи с трелло