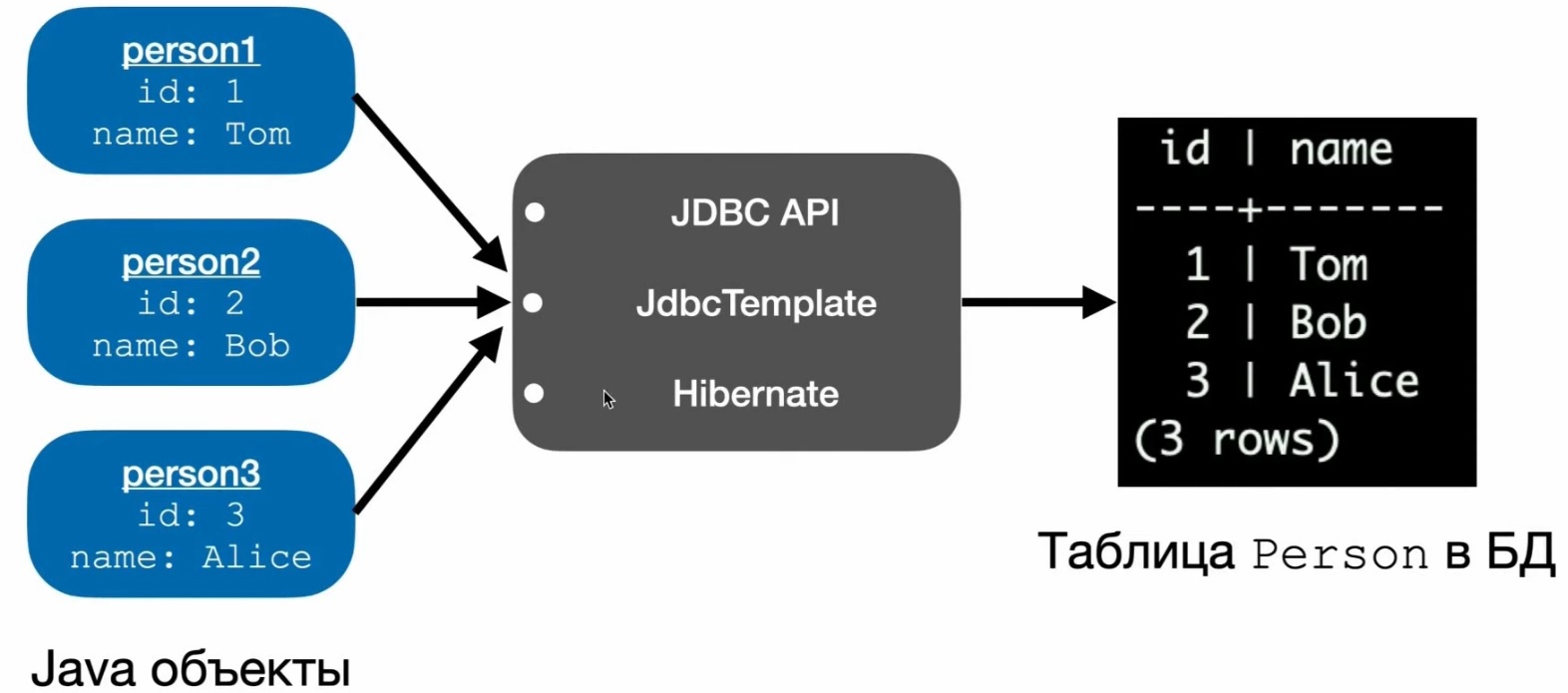
**JDBC APL Базы данных**

В реляционных базах данных объекты хранятся в виде строк в таблице.

Есть три способоа связи java приложения и базы данных: **JDBC API**, **JdbcTemplate**, **Hibernate**.



**JDBC API** – это самый низкоуровневый способ связи java приложения и базы данных. (мы сами делаем все запросы к БД и сами переводим Java объекты в строки таблицы и наоборот).

**JdbcTemplate –** тонкая обертка вокруг JDBC API. Часть Spring Framework. Предоставляет некоторые абстракции, берет часть дел на себя.

**Hibernate** – самый высокий уровень абстракции. Практически не пишем вручную запросы к БД. Автоматически переводит Java объекты в строки таблицы и наоборот. Может автоматически создавать таблицы в бд на основании наших Java классов. Этот функционал называется ORM (Object Relational Mapping).

**Базовые команды SQL**

**DDL(Data definition Language)**

* **CREATE DATABSE db\_name** - создает базу данных с указанным именем.
* **DROP DATABASE db\_name** – удаляет базу данных с указанным именем.
* **CREATE TABLE table\_name (**

**id int,**

**name varchar)** – создает таблицу с указанным именем

* **DROP TABLE** table\_name – удаление таблицы с указанным именем.

**DML(Data Manipulation Language)**

* **SELECT id, name FROM table\_name** – выбираем строки из существующей таблицы.
* **SELECT \* FROM table\_name** – выбирает все колонки из таблицы.
* **INSERT INTO Person(id, name) VALUES (1, ‘Tom’) –** добавляет строки в таблицу
* **UPDATE Person SET name=’Tom123’ WHERE id=1** – изменяет значение в какой то колонке по условию.
* **DELETE FROM Person WHERE id=1** – удаляет строку из таблицы по условию.

**Пример подключения к базе данных postgreSQL:**



**Пример выполнения запроса к базе данных:**

****

**1.** В первую очередь нужно завести объект **Statement** – это тот объект который содержит в себе SQL запрос к базе данных. Тоесть мы на нашем соединении(connection) создаем объект запрос к базе данных.

**2.** создаем в отдельной переменной SQL запрос.

**3.** Теперь на нашем statement`e мы должны выполнить этот SQL запрос с помощью метода ***executeQuery(SQL)***. ***executeQuery(SQL) -*** получает данные.

**4.** При выполнении SQL запроса нам вернётся несколько строчек из нашей базы данных.

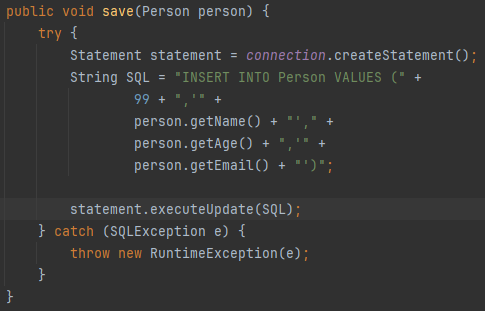
После выполнения запроса полученные строки помещаются в объект **ResultSet.**

**ResultSet –** это бъект который инкапсулирует результат выполнения запроса к базе данных. В этом объекте лежат наши строки.

**5.** Теперь пройдемся по нашим строкам и каждую из этих строк переводим в Java объект.

С помощью методов ***getInt(), getString()***  из **resulSet**`a получаем значения из колонок названия которых мы передали в аругменты этих методов.

**Пример добавления объекта в базу данных:**

****

**1.** Создадим **Statement** на нашем соединении.

**2.** Создадим SQL запрос.

**3.** Выполняем запрос с помощью метода ***executeUpdate()***.

***ExecuteUpdate()*** *-* служит для изменения данных в базе данных, при этом он никаких данных не возвращает.