The Statement interface provides methods to execute queries with the database.

**statement.execute(String SQL)**: универсальный запрос, можно и получить данные и зделать вставку, можно получать несколько result set-ов

**statement.executeUpdate(String SQL)-** выполняет insert, update, delete, получать данные нельзя

**statement.executeQuery(String SQL) -** SQL statement, which returns a single ResultSet object. обычно это Select

—-------------------------------------------------------**prepared statment**—-----------------------------------------------

It is a subinterface of Statement. Prepared Statement objects have some useful additional features than Statement objects. Instead of hard coding queries, PreparedStatement object provides a feature to execute a parameterized query.

**We can use the same PreparedStatement and supply with different parameters** at the time of execution.

An important advantage of PreparedStatements is that they prevent SQL injection attacks.

Весь смысл в том, что, можно сделать запрос в бд, но вместо значений определенных параметров поставить знаки вопроса, а не сами значения, это будет выглядеть как шаблон.

**update laptop Set speed=? where price=?**; Вместо этих знаков вопроса мы будем подставлять значения с помощью средств которые есть в preparedStatement, а именно, если в бд подставляемое значение int то используем  **preparedStatement.setInt( index, value),** index ← это порядковый номер знака вопроса в нашем шаблоне(нам же надо знать вместо какого вопроса мы ставим значение), value← значение которое мы хотим подставить. Если подставляемое значение String то **preparedStatement.setString(index, value)** - здесь по аналогии. В общем в зависимости от подставляемого типа данных выбираем соответствующий метод (их там много разных))

Пример:

**String sqlQuery="Select model from laptop where price=?;"; ←**наш запрос в бд с ? вместо него подставим значение

**PreparedStatement preparedStatement=connection.prepareStatement(sqlQuery);** ← создание preparedstatment

Scanner scanner=new Scanner(System.in); ← для ввода заранее неизвестного значения переменной.

int price=scanner.nextInt();

**preparedStatement.setInt(1,price)**; ←подстовляем значения(первое это номер параметра, второе это значение)

**resultSet = preparedStatement.executeQuery();** ←получаем Результат, дальше можно обрабатывать

Таких знаков `?` может быть сколько угодно, главное помнить про их порядковый номер (начиная с 1 слева направо) и правильно указывать при вызове методов set…

**PreparedStatement.executeUpdate():** This method can be used to create, drop, insert, update, delete etc. It returns int type. Параметры не указывать(они указываются при создании preparedStatement)