PreparedStatement Performance

It takes time for a database to parse an SQL string, and create a query plan for it. A query plan is an analysis of how the database can execute the query in the most efficient way.

If you submit a new, full SQL statement for every query or update to the database, the database has to parse the SQL and for queries create a query plan. By reusing an existing `PreparedStatement` you can reuse both the SQL parsing and query plan for subsequent queries. This speeds up query execution, by decreasing the parsing and query planning overhead of each execution.

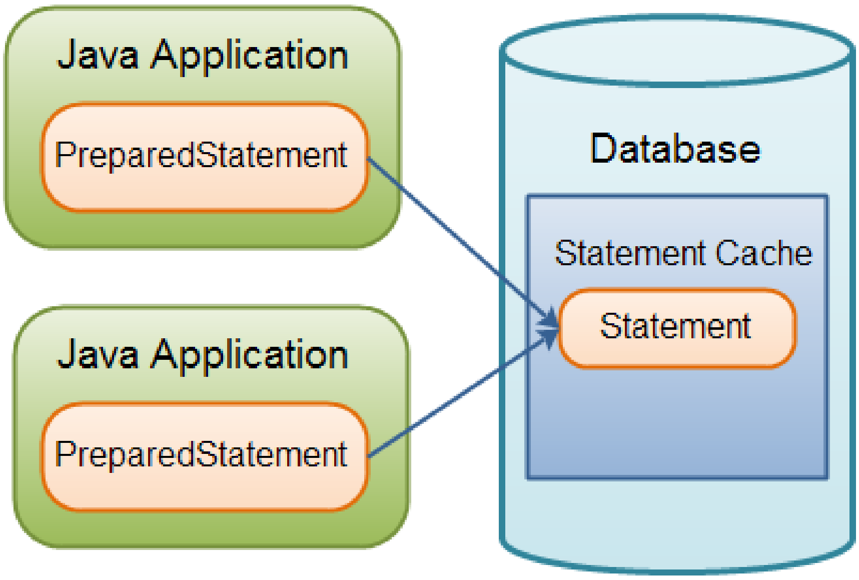
There are two levels of potential reuse for a `PreparedStatement`.

1. Reuse of PreparedStatement by the JDBC driver.
2. Reuse of PreparedStatement by the database.

First of all, the JDBC driver can cache `PreparedStatement` objects internally, and thus reuse the`PreparedStatement` objects. This may save a little of the `PreparedStatement` creation time.

Second, the cached parsing and query plan could potentially be reused across Java applications, for instance application servers in a cluster, using the same database.

Here is a diagram illustrating the caching of statements in the database:



The caching of PreparedStatement's in the database.

The diagram does not show the JDBC driver `PreparedStatement` cache. You will have to imagine that.