Loeng 4

Jdbc, PostgreSql

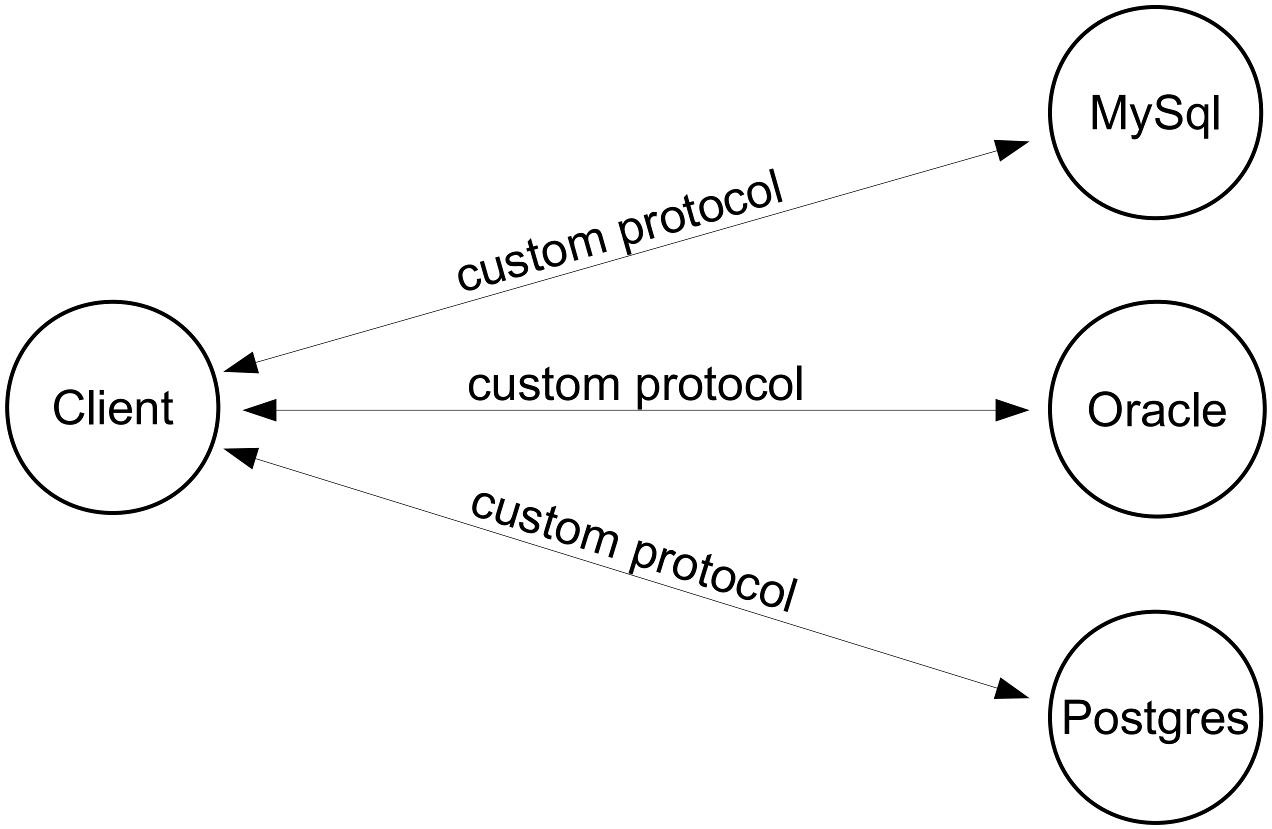
Kordamine

- Checked / Unchecked exception

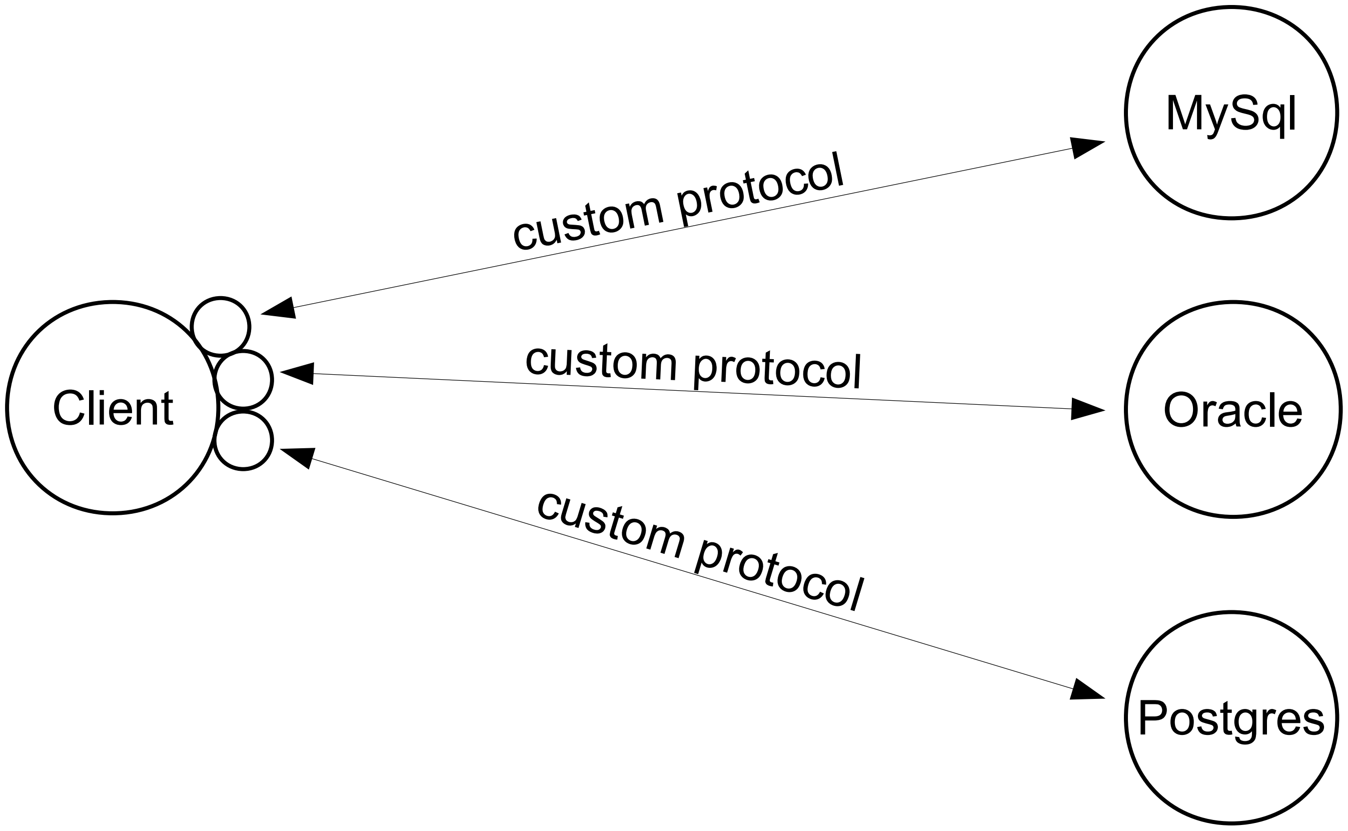
Jdbc

- Java Database Connectivty
- Andmebaasi API
- Madal abstraktsiooni tase

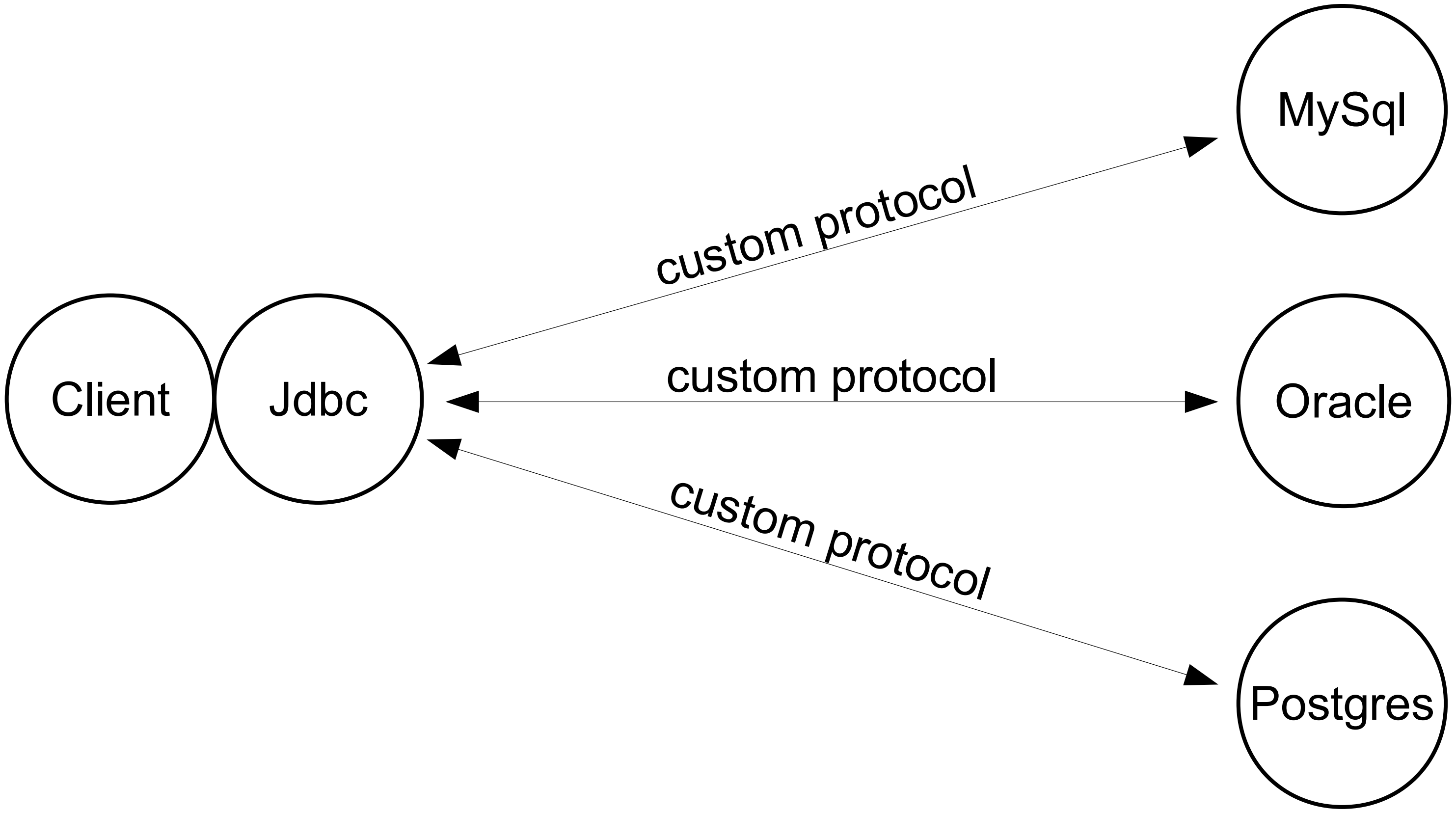
Jdbc


MySql
custom protocol
Client
custom protocol
Oracle
custom protocol
Postgres

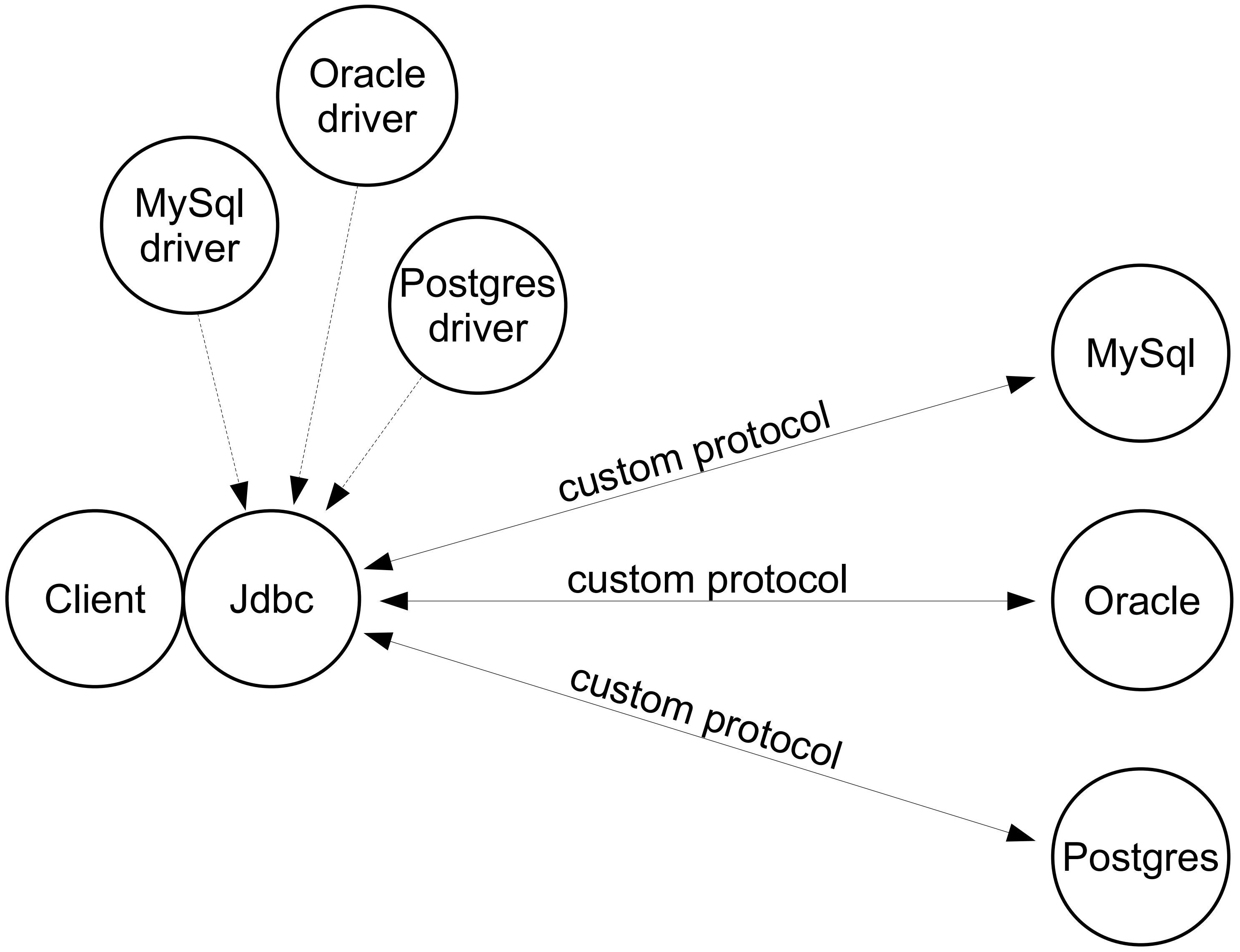
Jdbc


Client
MySql
Oracle
Postgres
custom protocol
custom protocol
custom protocol

Jdbc


Client
Jdbc
custom protocol
custom protocol
custom protocol
MySql
Oracle
Postgres

Jdbc


Oracle driver
MySql driver
Postgres driver
MySql
custom protocol
Client
Jdbc
custom protocol
Oracle
custom protocol
Postgres

Andmebaas

- Päringute optimeerimine
- Ühenduste ja ühenduste puuli optimeerimine
- Transaktsiooni piiride optimeerimine
- Hulgi (batch) operatsioonid
- jne.

```java
Connection conn = DriverManager.getConnection(
        "jdbc:oracle:thin:@//localhost:1521/orcl", "scott", "tiger");
try {
    Statement stmt = conn.createStatement();
    try {
        ResultSet rset = stmt.executeQuery("select BANNER from SYS.V_$VERSION");
        try {
            while (rset.next())
                System.out.println(rset.getString(1));
        } finally {
            try {
                rset.close();
            } catch (Exception ignore) {
            }
        }
    } finally {
        try {
            stmt.close();
        } catch (Exception ignore) {
        }
    }
} finally {
    try {
        conn.close();
    } catch (Exception ignore) {
    }
}
```

Try with resources

```
try (Connection conn = ...) {

} catch (SQLException e) {
    throw new RuntimeException(e);
}
```

Try with resources

```java
Connection conn = ...

try (conn; ...) {

} catch (SQLException e) {
    throw new RuntimeException(e);
}
```

Pisut puhtamalt

```
try (Connection conn = DriverManager.getConnection(...);
     Statement stmt = conn.createStatement()) {

    ResultSet rset = stmt.executeQuery(
            "SELECT banner FROM sys.v_$version");

    while (rset.next()) {
        System.out.println(rset.getString("banner"));
    }

} catch (SQLException e) {
    throw new RuntimeException(e);
}
```

JDBC

- (Laadida draiver)
- Luua ühendus
- Teha päringud
- Sulgeda ressursid (Connection, Statement, PreparedStatement, ResultSet)

Draiveri laadimine

```
static {
    try {
        Class.forName("org.postgresql.Driver");
    } catch (ClassNotFoundException e) {
        throw new RuntimeException(e);
    }
}
```

Ühenduse loomine

```
String url = "jdbc:postgresql://db.mkalmo.xyz:5432/db";

Connection conn = DriverManager
        .getConnection(url, "user1", "s3cret");
```

JDBC Url

jdbc:oracle:thin@masinanimi:1521:baasinimi

jdbc:postgresql://db.mkalmo.xyz:5432/db

jdbc:alamprotokolli_nimi:driveri_spetsiifiline_info

```java
String url = "jdbc:postgresql://db.mkalmo.xyz:5432/db";

Connection conn = DriverManager
        .getConnection(url, "user1", "s3cret");
```

Tabeli loomine

```
try (Connection conn = DriverManager.getConnection(...);
     Statement stmt = conn.createStatement()) {

    stmt.executeUpdate(
        "CREATE TABLE person (id INT, name VARCHAR(100))");

} catch (SQLException e) {
    throw new RuntimeException(e);
}
```

Andmete sisestus

```
...

stmt.executeUpdate("INSERT INTO person VALUES (1, 'John')");

...
```

Eraldi meetod

```
private void executeUpdate(String queryString) {
    try (Connection conn =
                     DriverManager.getConnection(...);
         Statement stmt = conn.createStatement()) {

        stmt.executeUpdate(queryString);

    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}

executeUpdate(
  "CREATE TABLE person (id INT, name VARCHAR(100))");
executeUpdate("INSERT INTO person VALUES (1, 'John')");
executeUpdate("INSERT INTO person VALUES (2, 'Jill')");
```

Tagastusega päring

```
try (Connection conn = DriverManager.getConnection(...);
     Statement stmt = conn.createStatement()) {

    ResultSet rset = stmt.executeQuery(
            "SELECT id, name FROM person");

    while (rset.next()) {
        System.out.println(
             rset.getLong("id")
             + ", "
             + rset.getString("name"));
    }

} catch (SQLException e) {
    throw new RuntimeException(e);
}
```

```
1, John
2, Jack
3, Jill
```

Parameetritega päring

```
try (Connection conn = DriverManager.getConnection(...);
     PreparedStatement ps = conn.prepareStatement(
             "SELECT id, name FROM person WHERE id = ?")) {

    ps.setLong(1, 3L);

    ResultSet rset = ps.executeQuery();

    ...
```

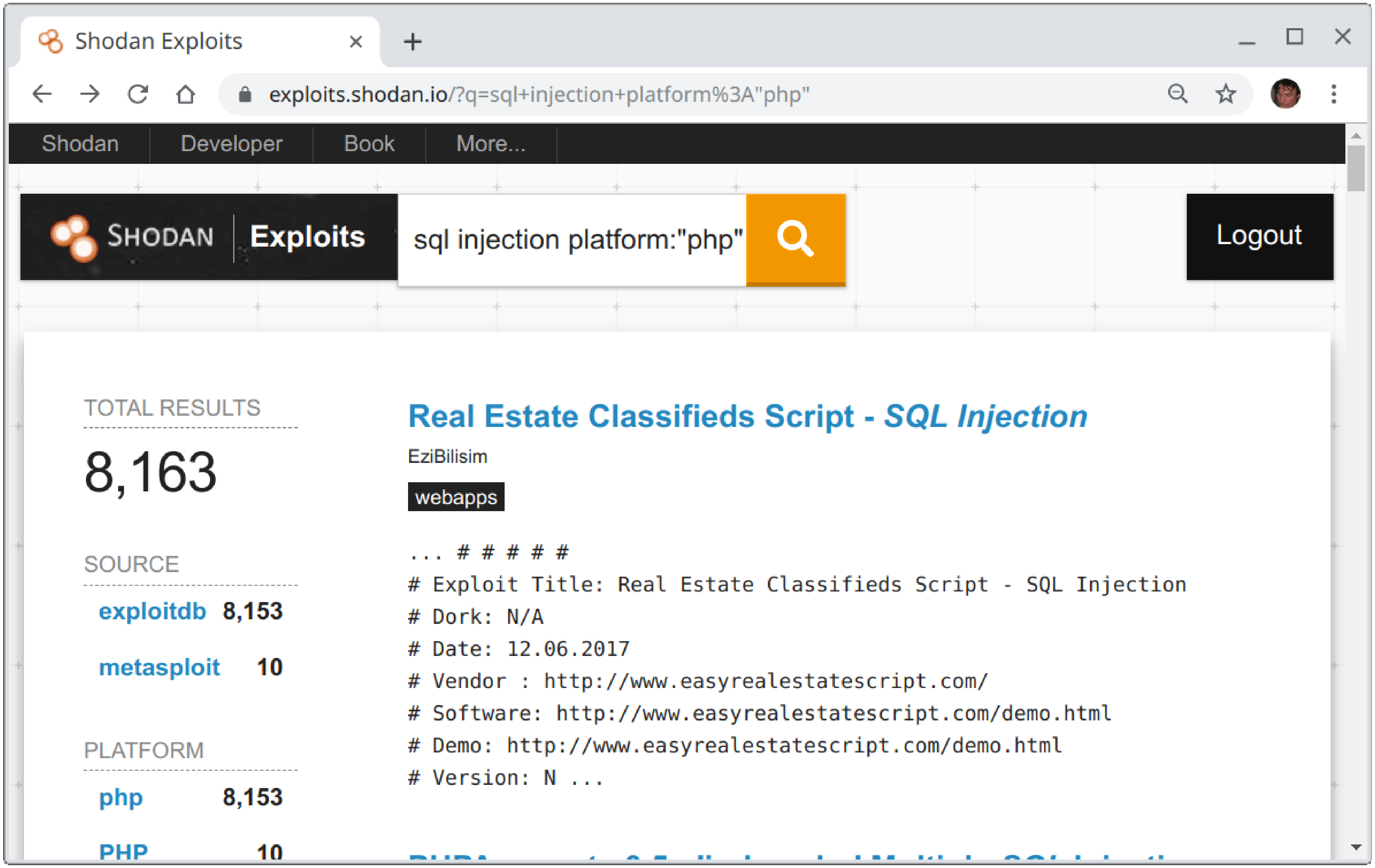
"...&id = " + id; // EI TOHI!!!

Sql injection

../users?id=105; DROP TABLE customers

DELETE FROM users WHERE id = 105; DROP TABLE customers

Sql injection



Shodan Exploits
exploits.shodan.io/?q=sql+injection+platform%3A"php"
Shodan
Developer
Book
More...
SHODAN
Exploits
sql injection platform:"php"
Logout

TOTAL RESULTS

8,163

SOURCE

exploitdb	8,153
metasploit	10

PLATFORM

php	8,153
PHP	10

Real Estate Classifieds Script - *SQL Injection*

EziBilisim

webapps

```
... # # # # #
# Exploit Title: Real Estate Classifieds Script - SQL Injection
# Dork: N/A
# Date: 12.06.2017
# Vendor : http://www.easyrealestatescript.com/
# Software: http://www.easyrealestatescript.com/demo.html
# Demo: http://www.easyrealestatescript.com/demo.html
# Version: N ...
```

PreparedStatement

- Sql lausete kompileerimine ja taaskasutamine

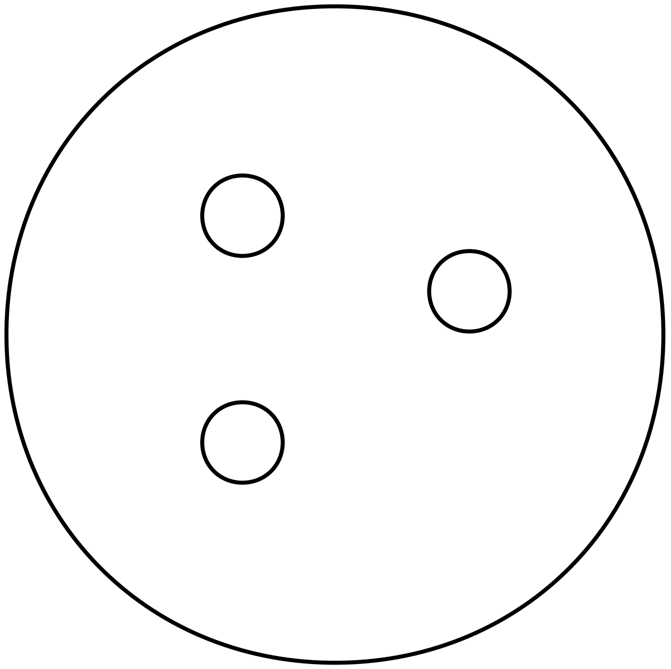
Parameetritega päring

```java
public void executeUpdate(String query,
                          Object ... parameters) {

    try (Connection conn = ...
         PreparedStatement ps = conn.prepareStatement(query)) {

        int pos = 1;
        for (Object parameter : parameters) {
            ps.setObject(pos++, parameter);
        }
        ps.executeUpdate();

    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}
```

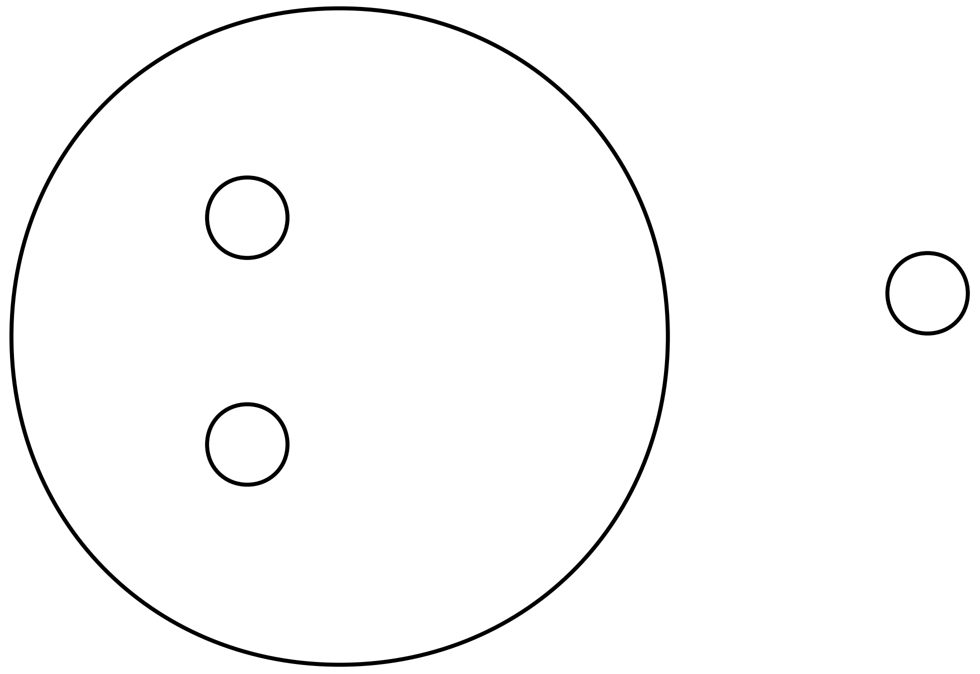
JDBC

- Andmebaasi API: DriverManager, Connection, Statement, PreparedStatement, ResultSet, …
- Asuvad java.sql pakis (nt. java.sql.Connection)

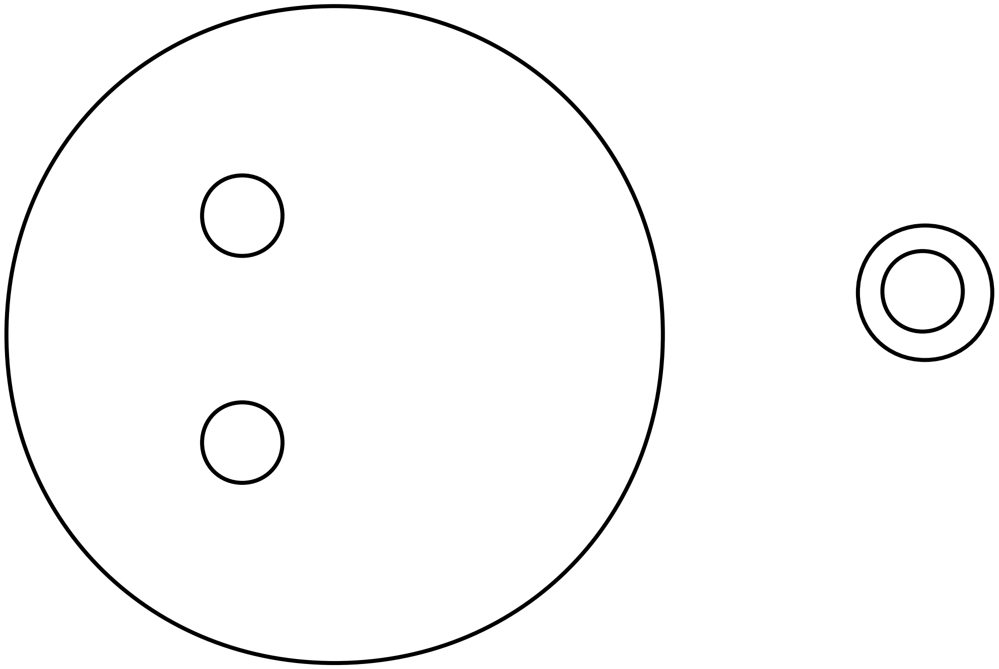
Ühenduste puul (connection pool)



Ühenduste puul (connection pool)

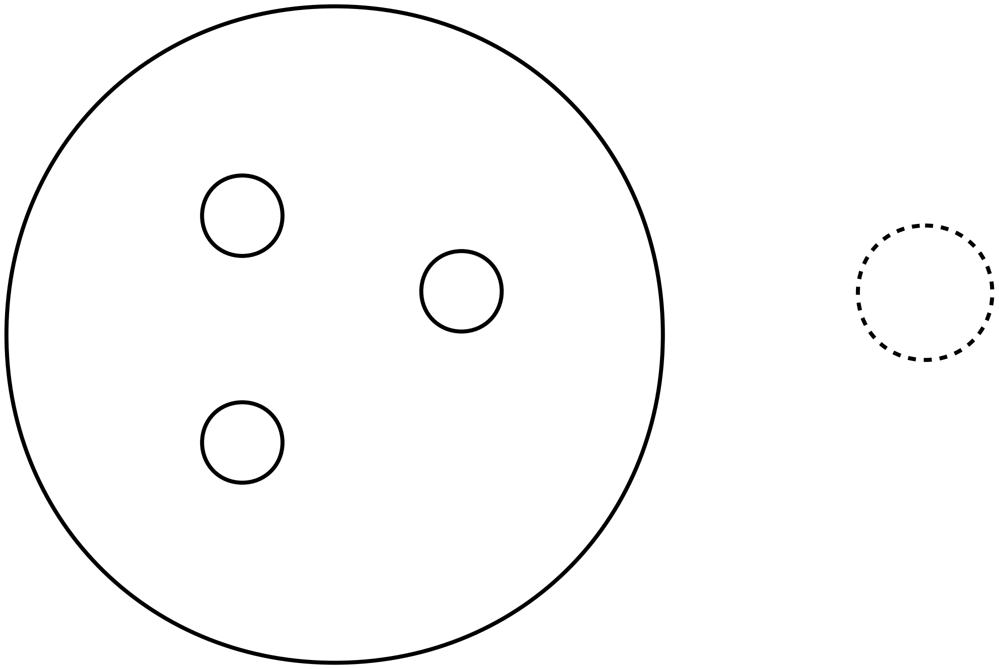


Ühenduste puul (connection pool)



```
public interface Connection extends AutoCloseable, ...
```

Ühenduste puul (connection pool)



javax.sql.DataSource

- Abstraktne koht ühenduste saamiseks.
- Connection pool on üks võimalik implementatsioon

javax.sql.DataSource

```
package javax.sql;

import java.sql.Connection;
import java.sql.SQLException;
import java.sql.Wrapper;
import javax.sql.CommonDataSource;

public interface DataSource ... {
    Connection getConnection() throws SQLException;

    ...
}
```

DataSource (connection pool)

```
BasicDataSource ds = new BasicDataSource();

ds.setDriverClassName("org.postgresql.Driver");

ds.setUrl("jdbc:postgresql://db.mkalmo.xyz:5432/db");
ds.setUsername("user1");
ds.setPassword("s3cret");
```

DataSource

```
DataSource dataSource = ... // loo puul

Connection c1 = dataSource.getConnection();
Connection c2 = dataSource.getConnection();


c1.close();


Connection c3 = dataSource.getConnection();
                // saame sama ühenduse, mis oli muutujas c1
```

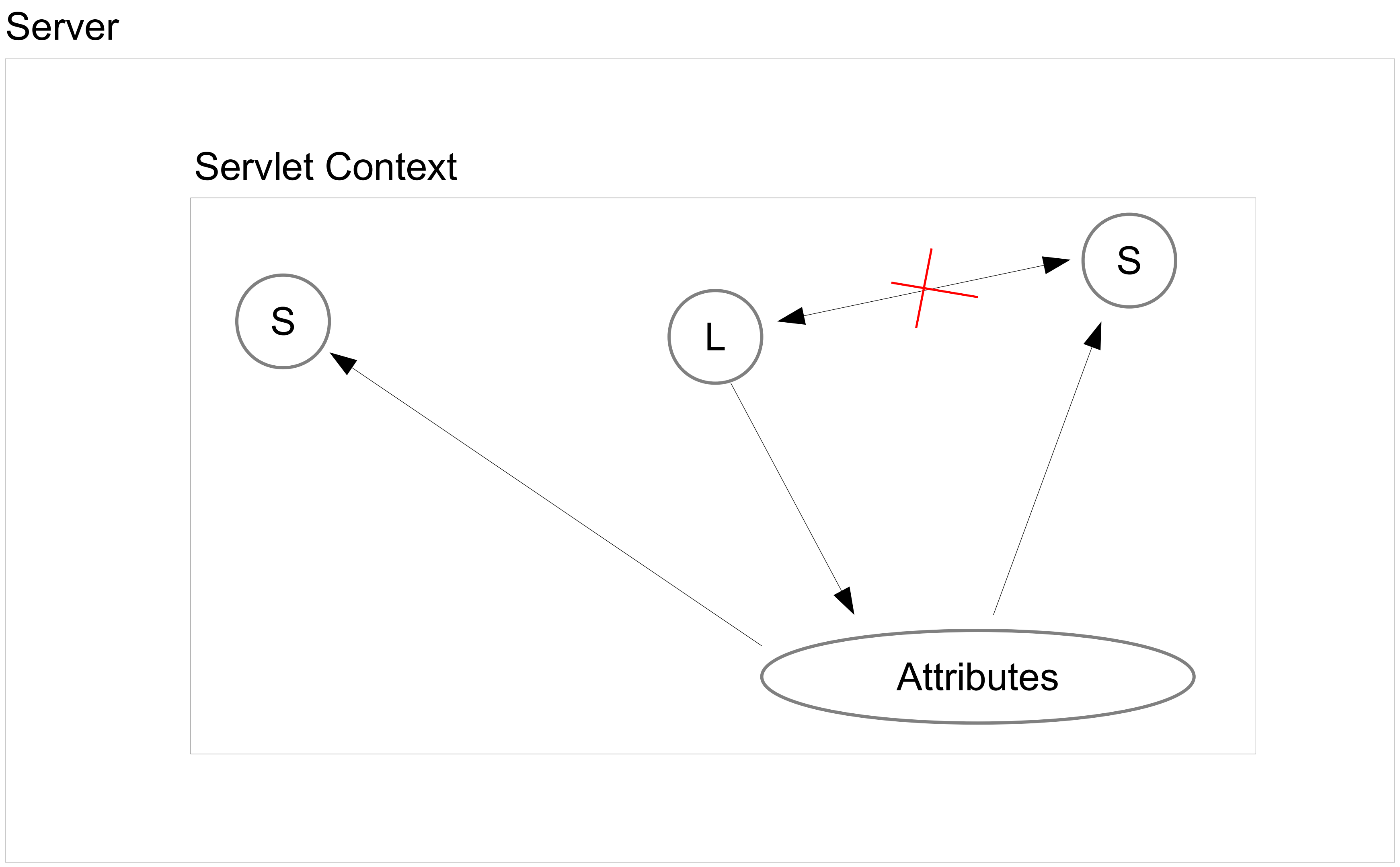
BasicDataSource

```
compile group: 'org.apache.commons',
        name: 'commons-dbcp2',
        version: '2.7.0'
```

Puuli jagamine

- Luuakse rakenduse käivituses
- Pöörata tähelepanu, et loodaks ainult üks koopia, mida kasutab kogu rakendus

Puuli jagamine



PostgreSQL andmebaas

- https://www.postgresql.org/
- https://www.techrepublic.com/article/theres-one-big-reason-that-postgres-cant-kill-oracle-and-its-not-the-technology/

Oracle hind

	Named User Plus	Software Update License & Support	Processor License
Database Products			
Oracle Database			
Standard Edition 2	350	77.00	17,500
Enterprise Edition	950	209.00	47,500
Personal Edition	460	101.20	-
Mobile Server	-	-	23,000
NoSQL Database Enterprise Edition	200	44	10,000
Enterprise Edition Options:			
Multitenant	350	77.00	17,500
Real Application Clusters	460	101.20	23,000
Real Application Clusters One Node	200	44.00	10,000
Active Data Guard	230	50.60	11,500
Partitioning	230	50.60	11,500
Real Application Testing	230	50.60	11,500
Advanced Compression	230	50.60	11,500
Advanced Security	300	66.00	15,000
Label Security	230	50.60	11,500
Database Vault	230	50.60	11,500
OLAP	460	101.20	23,000
Advanced Analytics	460	101.20	23,000
Spatial and Graph	350	77.00	17,500
TimesTen Application-Tier Database Cache	460	101.20	23,000
Database In-Memory	460	101.20	23,000

PostgreSQL andmebaas

```
compile group: 'org.postgresql',
        name: 'postgresql',
        version: '42.2.16'
```

PostgreSQL andmebaas siin kursusel

Host: db.mkalmo.xyz

User: teie Bitbucket-i kasutajanimi (võimalike muudatustega)

Password: projekti esimeses osas valitud salasõna räsi neli esimest märki

Database: sama, mis kasutajanimi

PostgreSql süntaks

- Toetab suurt osa SQL:2011 standardist

Create table

```
CREATE TABLE employee (
    id BIGINT NOT NULL PRIMARY KEY,
    name VARCHAR(255) NOT NULL
);
```

Create table (foreign key)

```
CREATE TABLE employee (
       id BIGINT NOT NULL PRIMARY KEY,
       name VARCHAR(255) NOT NULL,
       superior_id BIGINT,
       FOREIGN KEY (superior_id)
          REFERENCES employee(id) ON DELETE CASCADE
);
```

Insert

```
INSERT INTO employee (id, name, code) VALUES (1, 'CEO', '1');
```

NB! ühekordsed jutumärgid

Select

```
SELECT * FROM employee where code = ?
```

Update

```sql
UPDATE employee SET name = ?, code = ? WHERE id = ?
```

Join

```
SELECT e.* FROM employee e
  LEFT JOIN phone p ON p.owner_id = e.id
     WHERE p.number = ?
```

id (counter)

```
public static Long id = 1L;

...

System.out.println(id++);
```

NB! Kehv lahendus.

id (counter)

```
public static AtomicLong id = new AtomicLong(1L);

...
```

NB! Kehv lahendus.

```
System.out.println(id.incrementAndGet());
```

id (select max)

```sql
select max(id) + 1 from person;
```

NB! Kehv lahendus.

id (serial)

```
CREATE TABLE post (
  id SERIAL NOT NULL PRIMARY KEY,
  title VARCHAR(255)
);

insert into post (title) values ('Title 1');
```

id (serial)

```sql
CREATE TABLE post (
  id BIGERIAL NOT NULL PRIMARY KEY,
  title VARCHAR(255)
);
```

id (sequence)

```
CREATE SEQUENCE seq1 START WITH 1;


INSERT INTO person (id, name)
   VALUES (nextval('seq1'), 'John');
```

id (sequence)

```
CREATE SEQUENCE seq1 START WITH 1;  -- bigint

CREATE TABLE person (
   id BIGINT NOT NULL
      PRIMARY KEY DEFAULT nextval('seq1'),
   name VARCHAR(255) NOT NULL,
   age int
);
```

Id (UUID)

- Universally unique identifier
- „3FDBC820-1F8D-4855-9A69-9FDCFA31B7DF”
- 16 baiti
- Globaalselt unikaalne

UUID (pros)

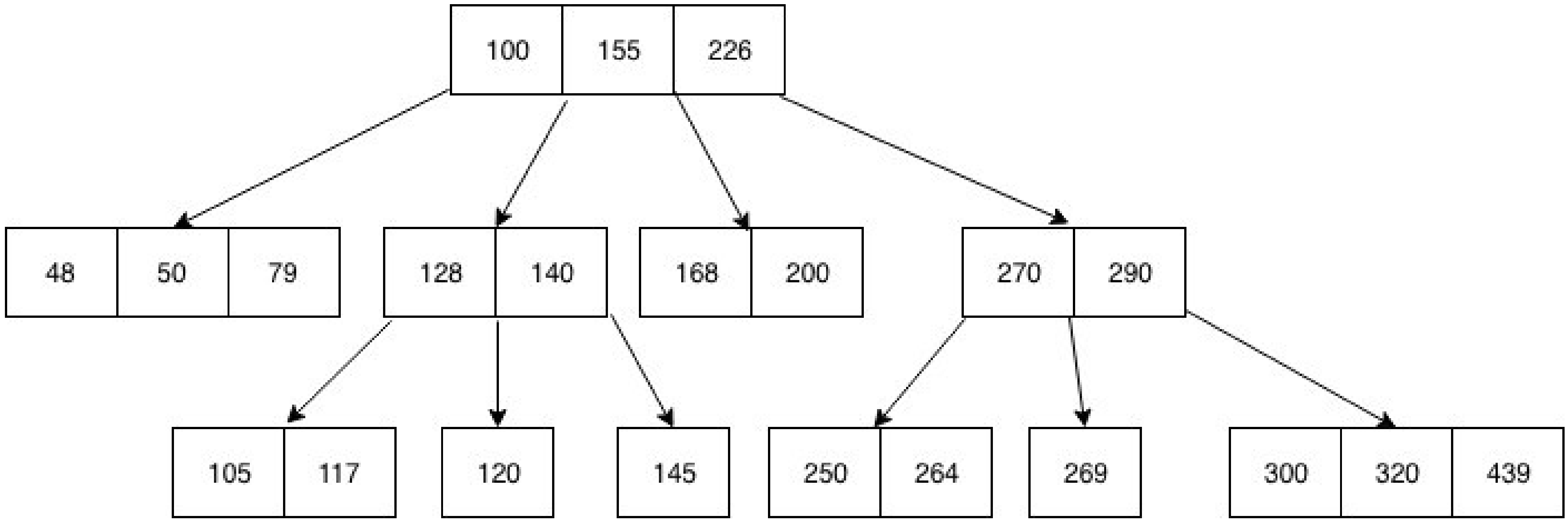
- Aitab varajata infot
- Ei ole äraarvatav
- Saab genereerida baasist väljas
- Unikaalne üle kõigi serverite

UUID (cons)

- Võib anda infot
- Arenduse ja testimise ajal ebamugav kasutada
- Pikad lingid (?id=123456 vs. ?id=3FDBC820–1F8D-4855–9A69–9FDCFA31B7DF)
- Andmemaht
- Ühilduvus erinevate andmebaasidega ja raamistikega
- Jõudlus*

* oleneb konkreetsest andmebaasist ja muudest teguritest

B-tree


100
155
226
48
50
79
128
140
168
200
270
290
105
117
120
145
250
264
269
300
320
439

Genereeritud info tagastamine

```
INSERT INTO person (id, name)
   VALUES (nextval('seq1'), 'John');
```

```
PreparedStatement ps =
     conn.prepareStatement(sql, new String[] { "id" });

...

ps.executeUpdate();

ResultSet rs = ps.getGeneratedKeys();
```

Genereeritud info tagastamine

```
 PreparedStatement ps =
      conn.prepareStatement(sql, new String[] { "id" });

...

ps.executeUpdate();

ResultSet rs = ps.getGeneratedKeys();

if(!rs.next()) {
    throw new RuntimeException("unexpected errror!");
}

System.out.println(rs.getLong("id"));
```

Baasi sisu kustutamine

```
DROP TABLE IF EXISTS person;

DROP SEQUENCE IF EXISTS seq1;

...
```

Dao muster

- Data Access Object
- aka. Repository

• Et eraldada andmebaasi kood ülejäänud rakenduse koodist.

Dao muster

```java
public class PersonDao {

    public Person getPersonForId(String personId) {
        // fetch a person object
    }

    public List<Person> getAllPersons() {
        // fetch a list of person objects
    }

    public void savePerson(Person person) {
        // save a person object
    }

    // ...
}
```

Dao muster

```
Person person = personDao.getPersonForId("1");
```

Baasi skeemi laadimine

- Rakenduse käivitades
- Lugeda skeem failist
- Käivitada laused

Faili lugemine Java veebirakenduses

- Absoluutne asukoht (c:/Users/guest/schema.sql)
  vs
- Relatiivne asukoht (./schema.sql)

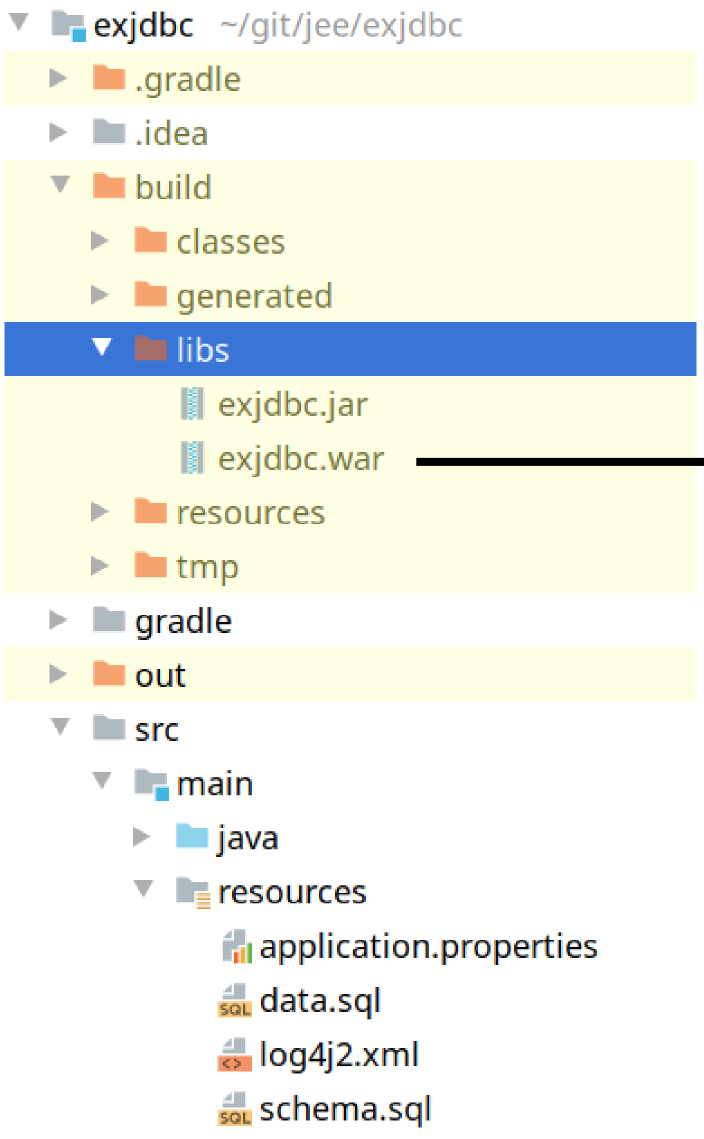
Faili lugemine Java veebirakenduses

```java
FileUtil.readFileFromClasspath("schema.sql");
```

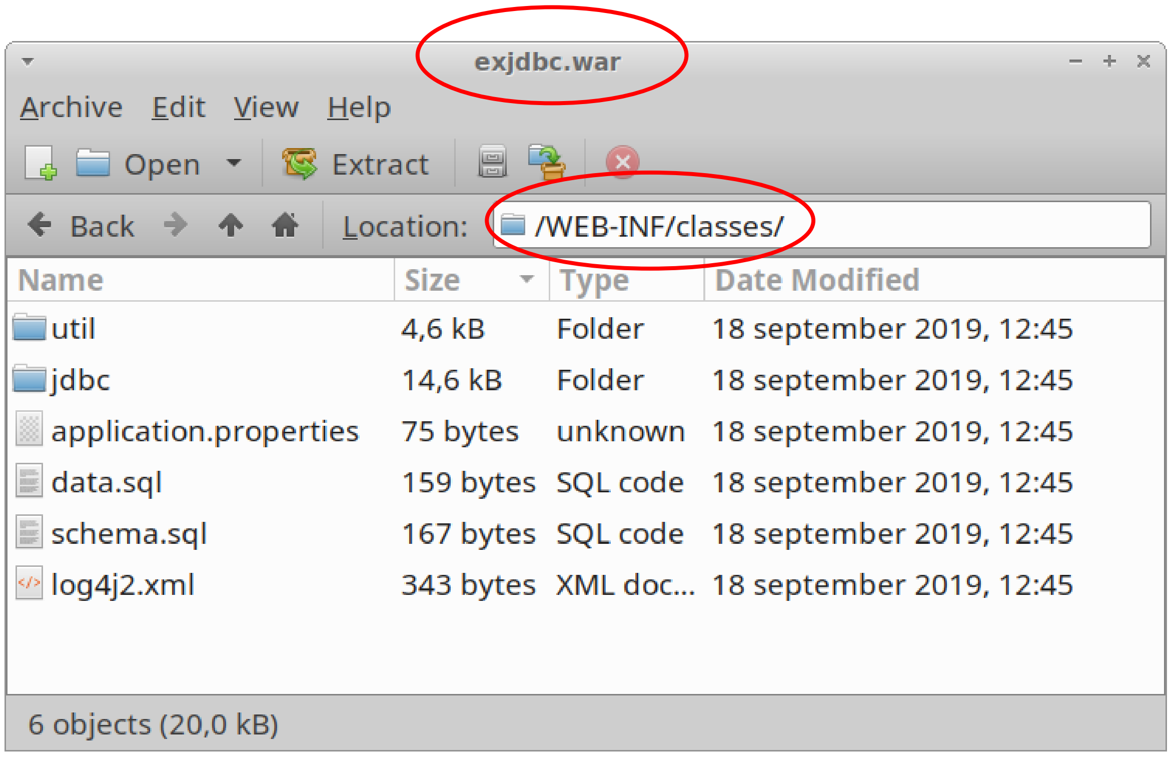
FileUtil klass on harjutustunni materjalides:

https://bitbucket.org/mkalmo/exjdbc/src/master/src/main/java/util/FileUtil.java

Faili lugemine Java veebirakenduses



```
exjdbc  ~/git/jee/exjdbc
  .gradle
  .idea
  build
    classes
    generated
    libs
      exjdbc.jar
      exjdbc.war
    resources
    tmp
  gradle
  out
  src
    main
      java
      resources
        application.properties
        data.sql
        log4j2.xml
        schema.sql
```



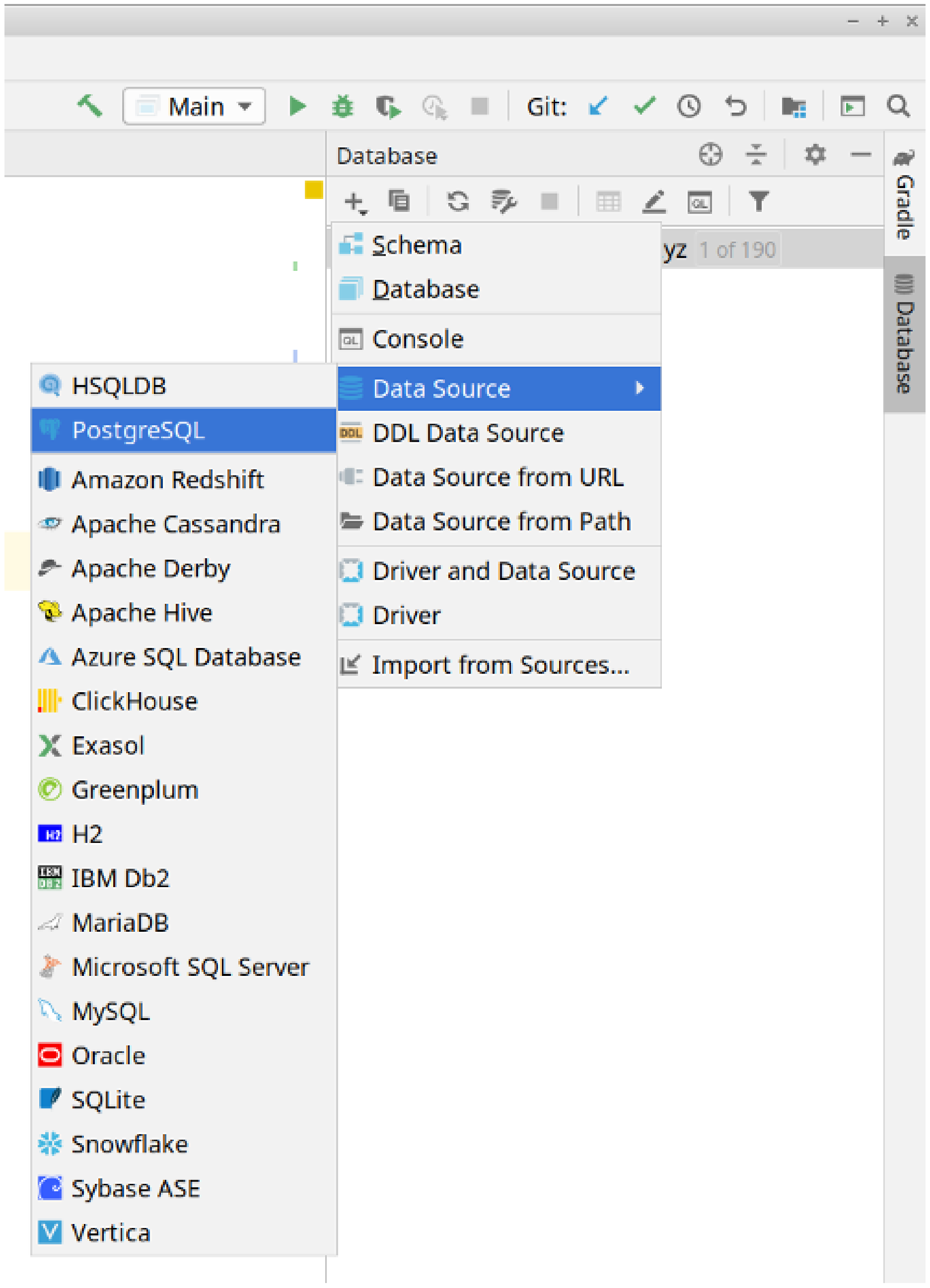
Location: /WEB-INF/classes/

Name	Size	Type	Date Modified
util	4,6 kB	Folder	18 september 2019, 12:45
jdbc	14,6 kB	Folder	18 september 2019, 12:45
application.properties	75 bytes	unknown	18 september 2019, 12:45
data.sql	159 bytes	SQL code	18 september 2019, 12:45
schema.sql	167 bytes	SQL code	18 september 2019, 12:45
log4j2.xml	343 bytes	XML doc...	18 september 2019, 12:45

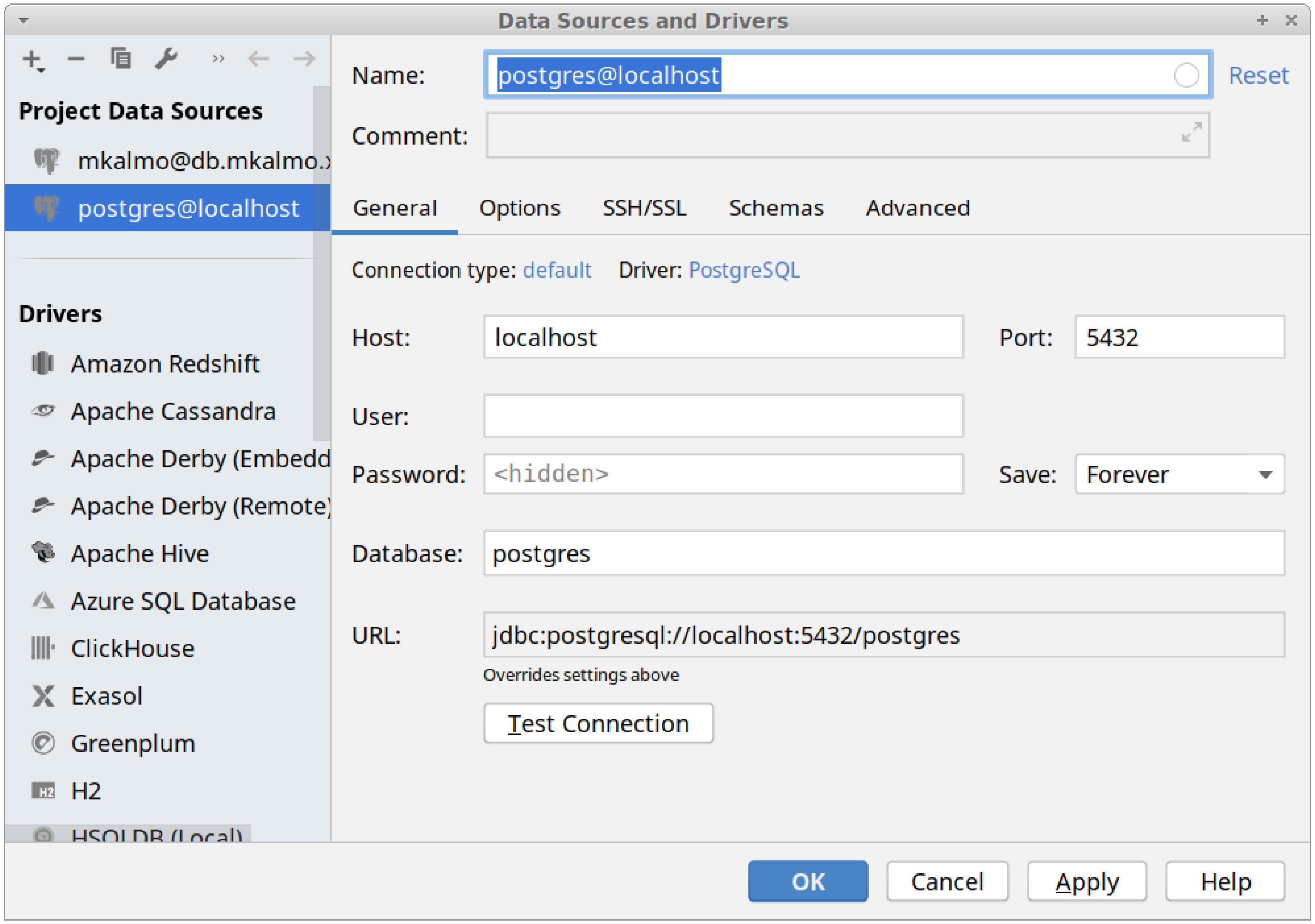
6 objects (20,0 kB)

Andmebaasi graafiline liides

- IDEA (Ultimate version)



Graafiline liides



Data Sources and Drivers

Project Data Sources

- mkalmo@db.mkalmo.x
- postgres@localhost

Drivers

- Amazon Redshift
- Apache Cassandra
- Apache Derby (Embedd
- Apache Derby (Remote)
- Apache Hive
- Azure SQL Database
- ClickHouse
- Exasol
- Greenplum
- H2
- HSQLDB (Local)

Name: postgres@localhost Reset

Comment:

General | Options | SSH/SSL | Schemas | Advanced

Connection type: default Driver: PostgreSQL

Host: localhost Port: 5432

User:

Password: <hidden> Save: Forever

Database: postgres

URL: jdbc:postgresql://localhost:5432/postgres

Overrides settings above

Test Connection

OK Cancel Apply Help

Rakenduse käivitamine

```
@Override
public void contextInitialized(ServletContextEvent sce) {

    // luua ühenduste puul

    // küsida üks ühendus

    // lugeda fail sql lausetega

    // teha päring faili sisuga

    // teha puul servlet-idele kättesaadavaks

}
```

Projekti 4. osa

- Panna eelmises osas tehtud rakendus kasutama PostgreSql andmebaasi
- Rakendus peab kasutama ühendust puuli
- Rakendus ei pea veel toetama tellimuse ridade sisestamist.

Batch operatsioonid

```
String query = "insert into person values (?, ?)";

try (PreparedStatement ps = conn.prepareStatement(query)) {
    for (int i = 1; i <= 300; i++) {
        ps.setLong(1, i);
        ps.setString(2, "John");
        ps.execute();
    }
}
```

kohalik: 0,3 sek
võrgus: 19 sek

Batch operatsioonid

```
String query = "insert into person values (?, ?)";

try (PreparedStatement ps = conn.prepareStatement(query)) {

    conn.setAutoCommit(false);

    for (int i = 1; i <= 300; i++) {
        ps.setLong(1, i);
        ps.setString(2, "John");
        ps.execute();
    }

    conn.commit();
}
```

kohalik: 0,04 sek
võrgus: 17 sek

Batch operatsioonid

```
String query = "insert into person values (?, ?)";

try (PreparedStatement ps = conn.prepareStatement(query)) {

    conn.setAutoCommit(false);

    for (int i = 1; i <= 300; i++) {
        ps.setLong(1, i);
        ps.setString(2, "John");
        ps.addBatch();
    }

    ps.executeBatch();

    conn.commit();
}
```

kohalik: 0,02 sek
võrgus: 0,3 sek