Requirement

• Implement your version of
  • java.sql.Driver
  • java.sql.Connection
  • java.sql.Statement
  • java.sql.ResultSet
  • java.sql.ResultSetMetaData
What Happens in the Client

• `Class.forName("driverClassName");`

• find and load driver class in JVM

• your `driverClassName` must be `fatworm.driver.Driver`

• `java.sql.Connection conn = DriverManager.getConnection("dbURL");`

• create connection to database
What Happens in the Client

• `java.sql.Statement stat = conn.createStatement();`

• `stat.execute("fwSQL statement");`

• `java.sql.ResultSet res = stat.getResultSet();`

• `java.sql.ResultSetMetaData schema = res.getMetaData();`
What Happens in the Client

- while (res.next()){}
  - move res to the next tuple if true
- res.beforeFirst();
  - point res to before the first tuple, so that after res.next(), it points to the first
- ind <= schema.getColumnCount();
- Object o = res.getObject(ind);
  - get the field specified by ind in the tuple res is pointing to
- int type = schema.getColumnType(ind)
  - return java.sql.Types
To Sum Up

- You should implement JDBC driver according to JDBC Specification on course website
A Hypothetical File System

- Why do we need a file system?
  - Because your database needs to be able to continue where it ended last time
A Hypothetical File System

TUPLE

# of attr  data  data
start position  data length  data type
A Hypothetical File System

Tuple

\[ \text{TO\_BYTE()} \]

Block

- data_size (15)
  - next_addr
  - prev_addr
  - finished (0)
  - block_addr (71)
  - dirty

- data_size (10)
  - next_addr
  - prev_addr
  - finished (1)
  - block_addr (73)
  - dirty
A Hypothetical File System

TABLE_ENV.FIRST_TUPLE = 1024

INSERT_TUPLE(71)

TABLE_ENV.FIRST_TUPLE = 71
A Hypothetical File System

Block
Block
Block
Block

Page

Cache (Memory)
Page

File System (Disk)
Page
A Hypothetical File System

• Tips:
  • You can fire up your file system in Driver.connect();
  • You can save to disk in Connection.close();
  • You should also store metadata like databases, tables, current database
To Sum Up

• You should implement a file system that enables your database to continue where it was left last time

• You should forget about all the details above and make your own design

• You can design however you like
Questions?