



Java DB

based on

Apache Derby:
A full-featured, light weight,
100% Java database

Francois Orsini

Senior Staff Engineer

Database Technology Group

Sun Microsystems, Inc.



**UNLOCK
OPPORTUNITY**

What will you open?

SUN TECH DAYS 2006-2007


A Worldwide Developer Conference

Agenda

- Introduction
- Features
- Configurations
- Performance
- Demo
- Open-Source Community (Apache Derby)

Java DB – Introduction

What is Java DB?

- The database for Java?
- Sun's supported distribution of Apache Derby
 - > Same source
 - > Same binaries
 - > Sun-supported
- Open source database technology
- Derby is a s  of the Apache DB Project

Main Characteristics

- Complete relational database engine
- Embeddable and client/server database
- Easy to use, zero maintenance
- Small footprint (2MB)
- Standards-based (JDBC, SQL92/99/2003)
- Secure, mature and robust
- Pure Java (write once run anywhere)

Java DB History

- 1996: Cloudscape founded – Release 1.0 97'
- 1999: Cloudscape acquired by Informix
- 2001: DB part of Informix acquired by IBM
- 2004: IBM donated Cloudscape to Apache as Derby
- Early 2005: Sun starts contributing to Apache Derby
- July 2005: Derby graduated from Apache Incubator
- December 2005: Sun announces Java DB

Java DB in Sun Products

- Java DB is used by
 - > Java System Application Server (incl. Glassfish)
 - > Java System Portal Server and Service Registry
 - > Java Studio Creator
- Java DB is supported by
 - > Netbeans
 - > Java Studio Enterprise
- Sun offers support for Java DB as a stand-alone product
- Available now at:
<http://developers.sun.com/prodtech/javadb/downloads/>

Java DB – Features

Pure Java



- Database code written in Java
- Write Once Run Anywhere
 - > Requires a J2SE 1.3, 1.4, 1.5 or 1.6 virtual machine
 - > Any hardware, any operating system, any vendor
- Single binary runs everywhere
 - > Solaris, Solaris x86, Linux, FreeBSD, Windows, MacOs, AIX, Z/OS, AS400, OS/390, ...
- Database on-disk format is platform independent, too!

Embeddable Database

- Database engine may run in application's virtual machine
 - > No additional process
 - > Database requests are method calls within the JVM
- Startup & shutdown controlled by application
- Just one JAR file
- Invisible to the user
- Easy to use, zero maintenance
- Can act as an embeddable server as well

Client/Server Database

- Database engine can run in a client/server configuration
- Standalone server
 - > Server runtime management tool
- Secure and support for various server connection mechanisms
- Easy to use, zero maintenance
- Can also run embedded in other server frameworks

Small Footprint

- Engine jar file is around 2MB
 - > Optional Jar files
 - > Network server ~150k
 - > Tools ~200k
- Runtime memory use
 - > Dependent on application, data caching etc.
 - > Can run when Java heap memory restricted to 4MB
 - > Have run in machines with only 16MB physical memory
- Engine jar can be compressed down to ~600K for faster download (Java 5 Pack200)
<http://blogs.sun.com/roller/page/FrancoisOrsini/20060118>

Standards Based

- SQL
 - > SQL92, SQL99, SQL2003, SQL/XML, ...
- Java
 - > J2SE 1.3, 1.4, 1.5 and 1.6
 - > JDBC 2.0 & 3.0 (4.0 as part of Java DB 10.2.2)
 - > Java ME CDC (JSR 169) (JDBC for CDC)

Complete Relational Engine

- Multi-user, transactions, isolation levels, deadlock detection, crash recovery
- SQL
 - > Tables, indexes, views, triggers, procedures, functions
 - > Foreign keys and check constraints
 - > Joins, cost based optimizer
- Data caching, statement caching, write ahead logging, group commit
- Online backup/restore
- Database encryption

Java Stored Procedures/Functions

```
CREATE FUNCTION SEND_MAIL(  
    TO_ADDRESS VARCHAR(320),  
    SUBJECT VARCHAR(320),  
    BODY VARCHAR(32000)) RETURNS INT  
LANGUAGE JAVA PARAMETER STYLE JAVA NO SQL  
EXTERNAL NAME 'testing.MailTest.sendSMTP_F';  
  
-- Send a Welcome e-mail when new customers are added.  
  
CREATE TRIGGER WELCOME_CUSTOMER  
AFTER INSERT ON CUSTOMER REFERENCING new_table AS newtab  
FOR EACH STATEMENT MODE DB2SQL  
SELECT SEND_MAIL(c.email, 'Welcome to AcmeWidgets', M.email_text)  
FROM newtab C, MAILINGS M  
WHERE C.TYPE = M.CUST_TYPE AND M.OFFER_TYPE = 'welcome'
```

Java SQL function

```
public static int sendSMTP_F
    (String toAddress, String subject, String content)
{
    recipient = new InternetAddress(toAddress);
    ...

    msg = new MimeMessage(session);
    msg.setFrom(from);
    msg.setSubject(subject);
    msg.setText(content);
    msg.addRecipient(Message.RecipientType.TO, recipient);
    javax.mail.Transport.send(msg);
    return 0;
}
```

See <http://wiki.apache.org/db-derby/SendEmailRoutine>

Security

- On-disk database encryption
- Built-in, LDAP, NIS+ and pluggable authentication
 - > Support for various network authentication mechanisms
 - > Authentication provider interface
- Support for strong network client authentication
 - > Derby network client driver
- ANSI grant/revoke privileges support
- Java Security Manager enabled

Some Architectural Aspects

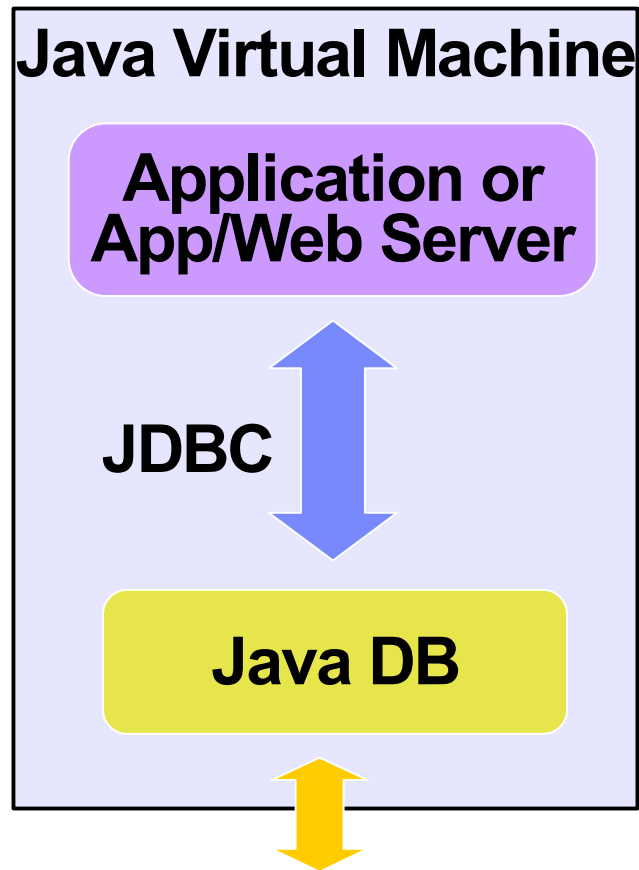
- SQL compiled into Java byte-code
 - > Pro: Fast. Utilizes JIT compiler
 - > Con: Compilation + class-loading
- Pluggable storage layer:
 - > File system (single directory)
 - > Jar file (read-only)
 - > In-Memory
 - > Future: HTTP, Email, etc.

Tools

- ij – Interactive SQL/Scripting tool
 - > JDBC neutral, can be used against other JDBC drivers
- dblook – schema extraction tool for Java DB
 - > Generates the DDL to be used to recreate database schema
- sysinfo – Java DB version information
 - > Output useful for bug reporting in Jira

Java DB – Configurations

Embedded Java DB



Database(s)
on disk

- Database only accessible from a single JVM
- May have multiple applications per JVM (e.g., app server)
- Easy to use
- Fast
- Zero administration

Starting an Embedded Java DB

- No specific start-up code!
- Loading the JDBC driver starts the embedded engine
- Making a connection request to the database starts that database, if it was not already running

Start Up Code

```
String driverClassName =  
    "org.apache.derby.jdbc.EmbeddedDriver";  
String databaseURL = "jdbc:derby:dbName";  
  
// Load the JDBC Driver  
Class.forName(driverClassName);  
  
// Open a connection to the database  
Connection conn =  
    DriverManager.getConnection(databaseURL);
```

JDBC Driver URL

- Format:
`jdbc:derby:[database][;attribute=value]*`
- Attributes: user, password, create, databaseName, logDevice, territory, *encryption options*, *recovery options*
- To create a new database, connect with URL:
`jdbc:derby:dbName;create=true`
- Attributes can also be passed in the Properties parameter of `DriverManager.getConnection` methods.

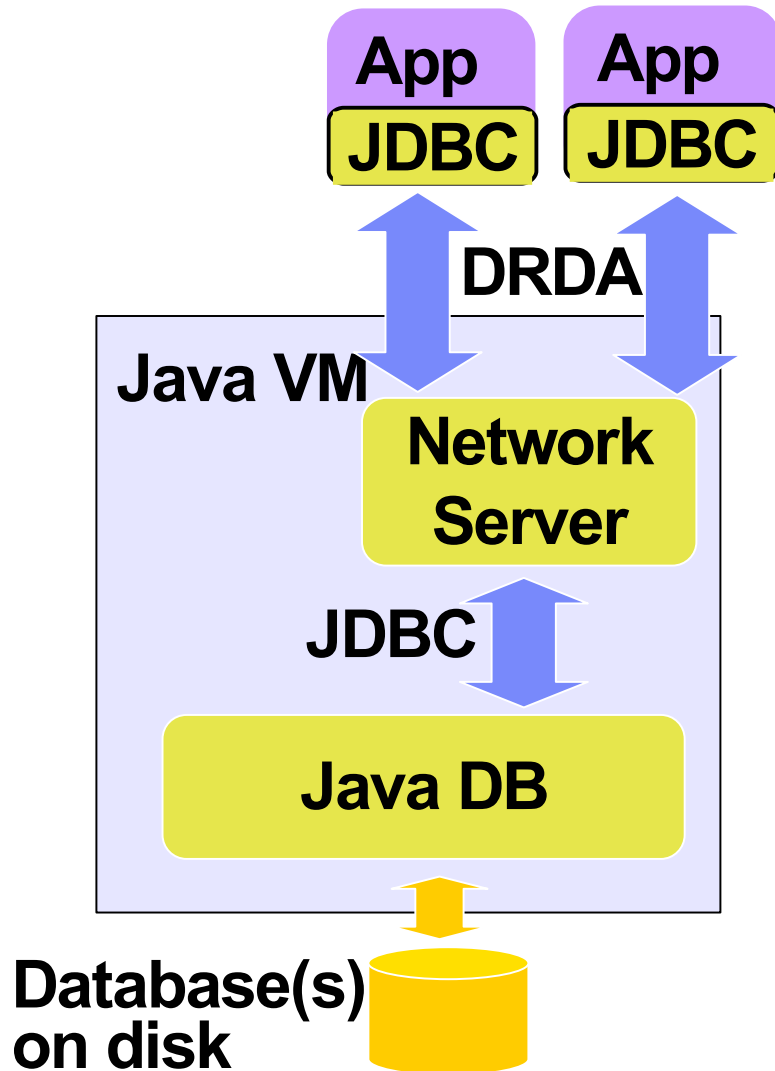
Shutting Down an Embedded DB

- Do nothing – exiting VM will stop Java DB
 - > Recovery will be run on next start
- Shut down a single database
 - > Make a connection request to the database with URL:

```
jdbc:derby:dbName;shutdown=true
```
- Shut down all databases and the engine
 - > Make a connection request without a database name:

```
jdbc:derby:;shutdown=true
```
- Note: A connection request for shutdown will throw an exception.

Client/Server

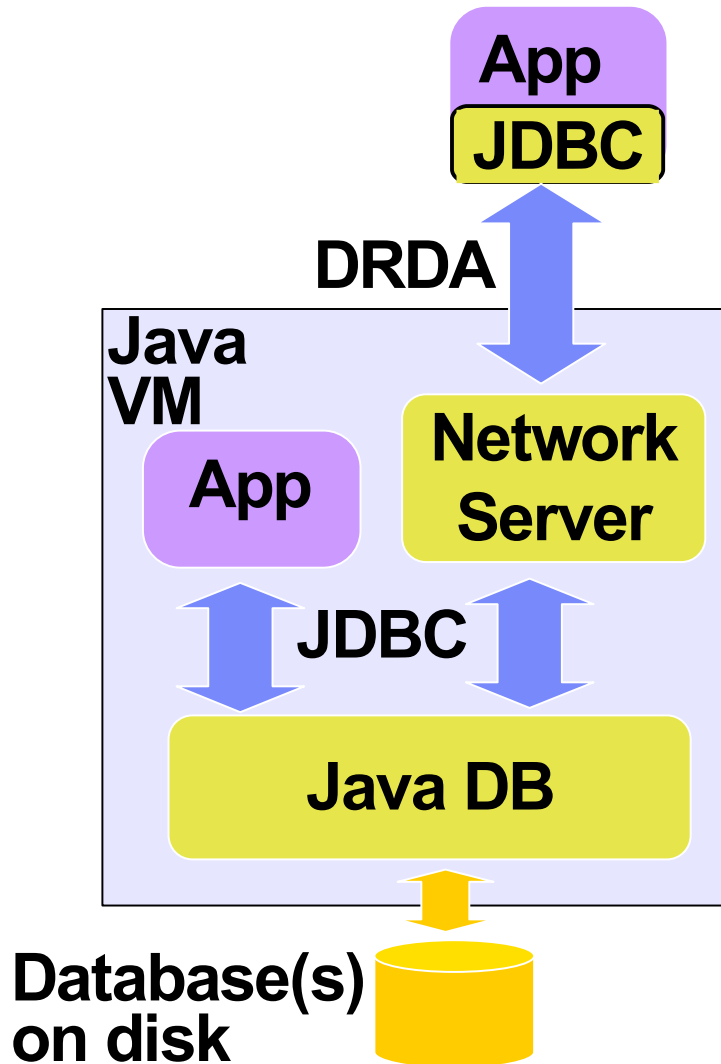


- Network Server uses embedded driver against Java DB
- Standard protocol (DRDA). May use drivers from other vendors.
- Scripts provided to start and stop a Network Server.

Avoid Network Server Security Risks

- Enable user authentication
- Run the network server with the Java 2 Security Manager enabled
 - > Procedures and functions will have no permissions
 - > Application code can be granted permissions
- If Internet access is not needed, ensure your firewall blocks the network server port

Embedded Network Server



- Provides access to database from outside the application's VM.
- Adds DB reporting and debugging capabilities to stand-alone application
- No code changes to application. Enabled by setting property `derby.drda.startNetworkServer`

Sample Deployments

- Local client store for web 2.0 apps (demo follows)
 - > Local AJAX
- Embedded local cache in the web tier
- Embeddable in rich client apps
- Read-only DB in jar file
- Java DB on a memory stick
- Great, simple, easy-to-use, easy-to-deploy DB for departmental client/server deployments

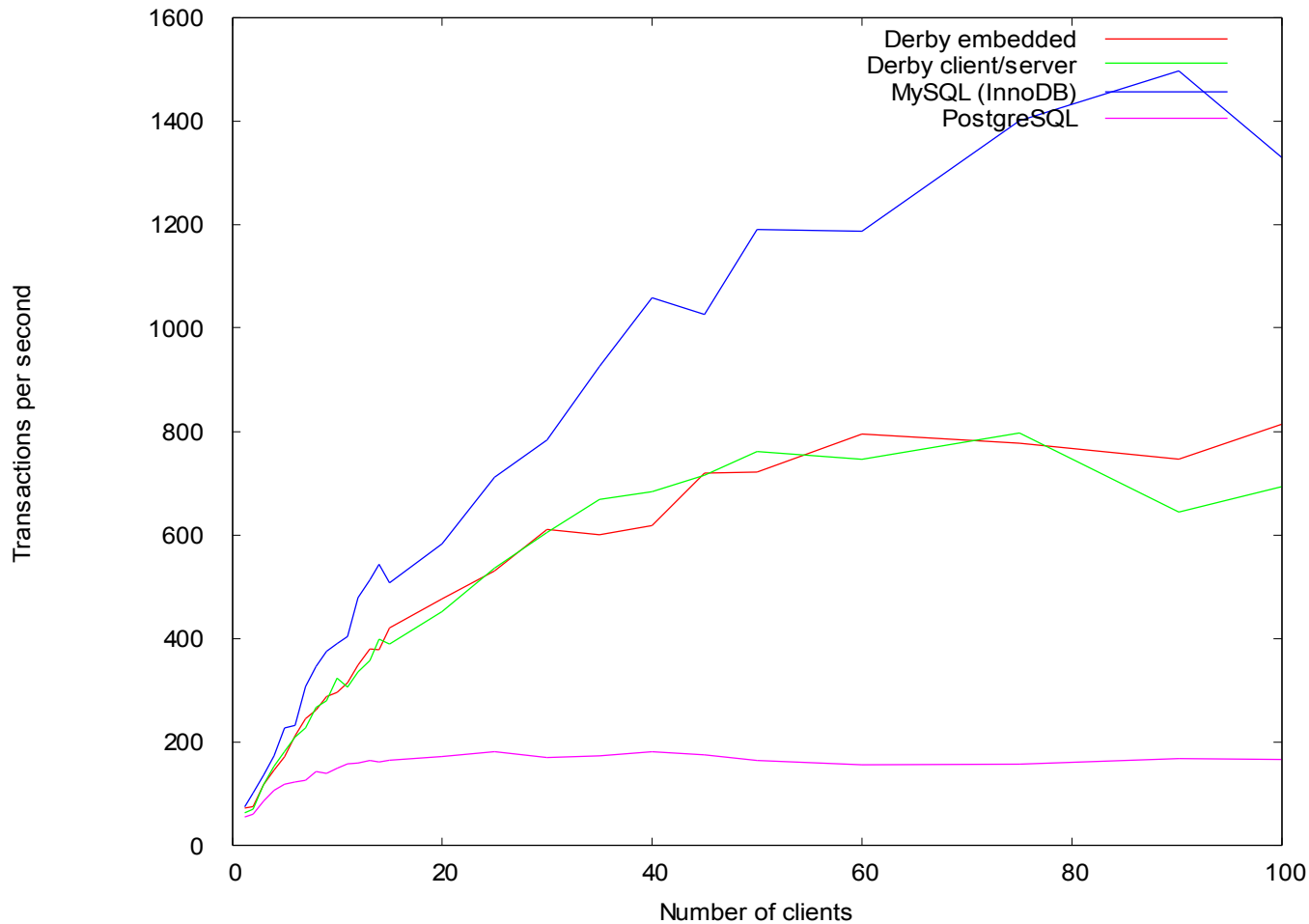
Java DB – Performance

Java DB Performs!

- Java DB performs well
 - > Comparable to competition
- Java is not slow!
 - > Just In Time (JIT) compilers compile interpreted byte code to native machine code
- We have studied and compared the performance of
 - > Java DB
 - > MySQL
 - > PostgreSQL

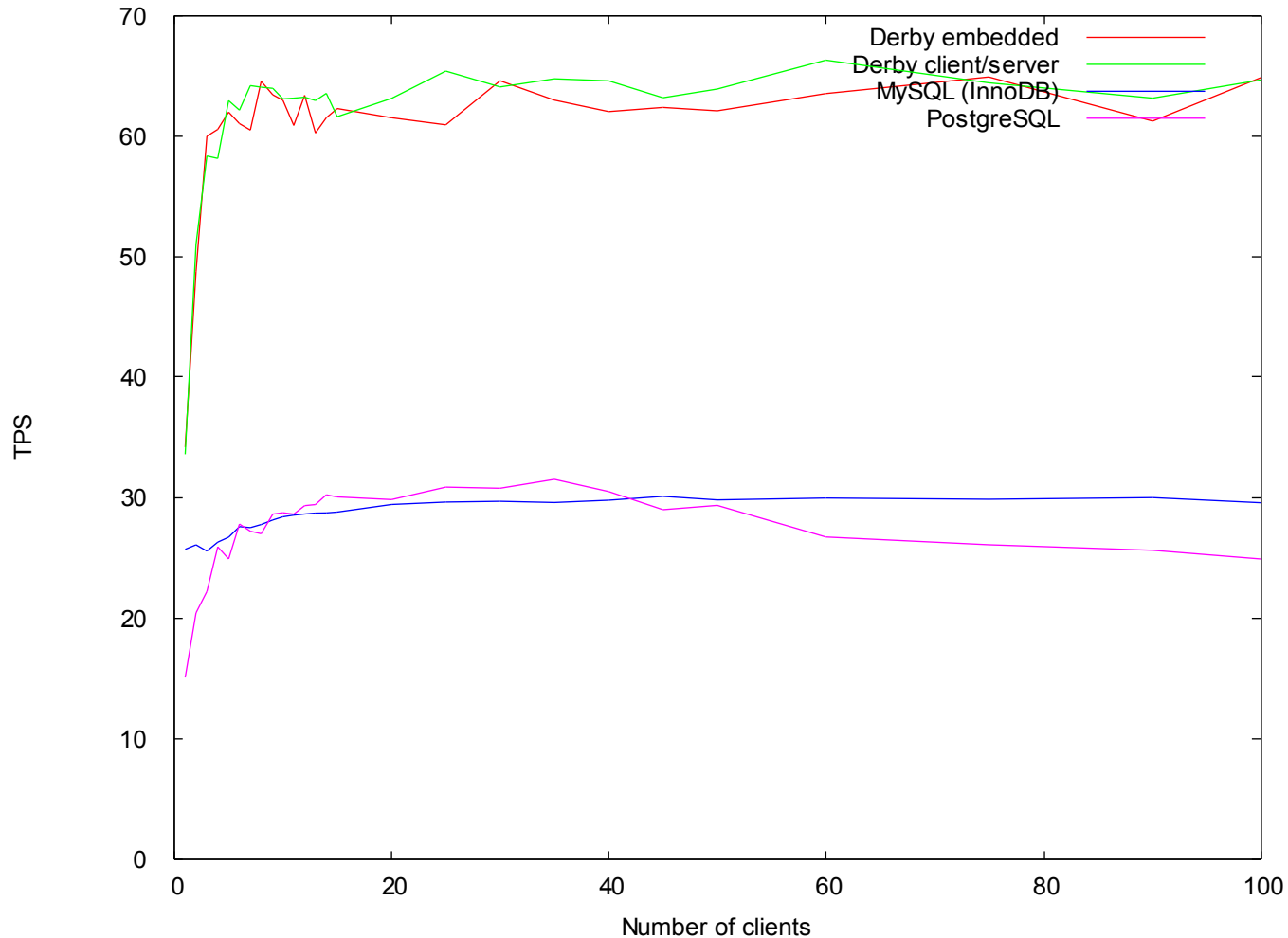
TPC-B Like Load, Main-Memory DB

DB 10 MB, page cache 50 MB



TPC-B Like Load, Disk-Based DB

DB 10 GB, page cache 64 MB



Performance hints

- Use (and reuse) prepare statements
 - > Not: “`SELECT a FROM t WHERE b=4`”
 - > But: “`SELECT a FROM t WHERE b=?`”
- Put DB log and data on separate disks
 - > Specify the `logDevice` attribute when creating DB
- Tune page cache size (default 4 MB)
 - > `derby.storage.pageCacheSize`
- Use indexes to avoid table scans
 - > Check query plans
(`derby.language.logQueryPlan=true`)

Java DB – Demo

Web App with Embedded Java DB

- A web application accessing Java DB embedded within Mozilla Firefox
- Java DB runs as a local client store. It is exposed to the web application through a Javascript interface (LiveConnect) or a local HTTP service (Ajax)
- Allows local caching of data on the client side
 - > No network connectivity to interact with the data store
 - > Store sensitive, private, confidential data on the client's host, instead of a remote server (or both)
 - > Java DB can encrypt the local database
 - > Local store for loosely connected Web apps

Demo – Things to Remember


- Ease of deployment over a large user base (e.g. consumer desktops)
- Transparent - Embeddable and zero-administration
 - > invisible to the end user
- ACID RDBMS - high levels of durability and consistency to prevent data loss
- Ease of upgrade (using Firefox or Java Web Start)
- Small footprint
- Highly secure to ensure desktop data is safe

Demo – More information

- Demo code publicly available at <http://developers.sun.com/prodtech/javadb/>
- For more information see <http://blogs.sun.com/roller/page/FrancoisOr>
- Francois is working on a next version of the demo
 - > Better Ajax integration (LAJAX)
 - > Make Java DB a Firefox extension

Java DB – Community

Apache Derby Community

- Apache Software License v2 
- Anyone can contribute
- Active contributors become committers through community vote.
- Apache Derby community is
 - > growing at fast pace
 - > a very active one
 - > a great place to learn more about database internals

Recent Work by the Community

- Additional Network Security mechanisms
- Scrollable updatable result sets
- JDBC 4.0
- GRANT/REVOKE (privileges)
- Online backup (non-blocking)
- Performance improvements
- Optimizer hints
- More data types (e.g., national characters, boolean)
- `Statement.setQueryTimeout`

Participate!

- <http://db.apache.org/derby>
 - > Download, read docs
- JIRA
<http://issues.apache.org/jira/browse/DERB>
 - > Report bugs, submit patches
- derby-user@apache.org
 - > Discuss experience, get help
- derby-dev@apache.org
 - > Discuss developer issues

Derby Integration

- ActiveMQ
- AntHill Pro
- Apache Cocoon
- Apache Geronimo
- Apache Xalan
- Daffodil Replicator
- Data Direct Sequelink
- DB Visual Architect
- Drone IRC Bot
- Eclipse
- Glassfish
- Hibernate
- IBM Cloudscape
- IBM DB2 Everyplace
- IBM DB2 JDBC Universal Driver
- IBM WebSphere App Server
- ISQL-Viewer
- Java DB
- JBoss
- JPOX
- Jython
- Kodo 3.3.3
- Maven
- Netbeans 5.0
- RIFE
- Red Hat Application Server
- Roller 2.0
- Sequoia (C-JDBC)
- SQuirreL SQL
- Sun Java ES
- Sun Java Portal Server
- Sun Java Studio
- Sun JavaEE
- Sun Service Registry
- SUSE Linux 9.3
- Zend core for IBM
- Tomcat