Java in the Database

Rick Hillegas

Apache Derby

September 12, 2006

Plug It!

- Dock a free database in your code
- Dock your code to the database

Sample Function

```
public static int computeAge
( java.sql.Date date)
{
  long    interval =
    System.currentTimeMillis() - date.getTime();
  return (int)
    (interval / MILLISECONDS_IN_YEAR);
}
```

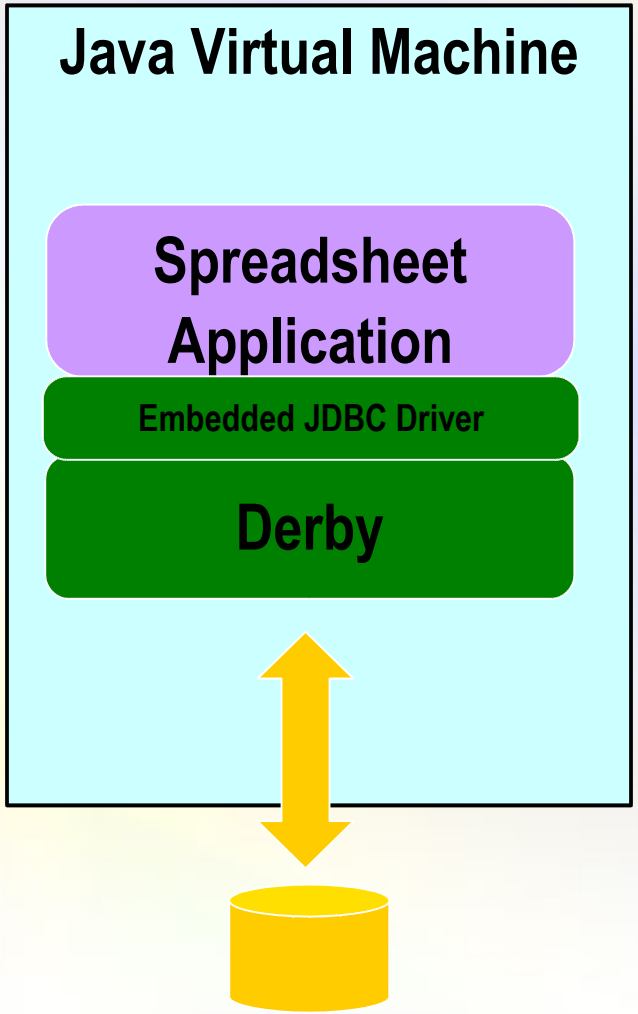
General Overview

- Derby Overview
- User Code in the Database
- Demo

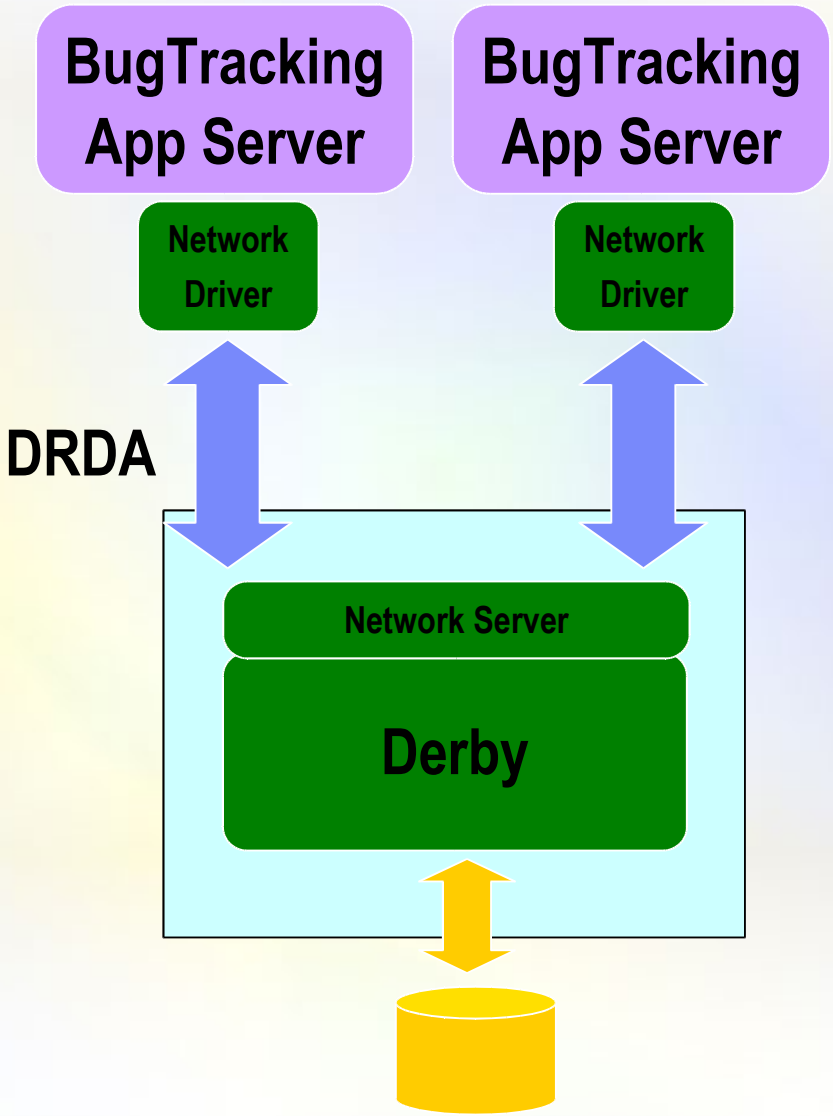
Derby Overview

- Lightweight, 0-admin, pure Java database
- Follows ANSI-SQL and JDBC standards
- Usage: embedded or client/server

Embedded


Java Virtual Machine
Spreadsheet Application
Embedded JDBC Driver
Derby

Client/Server



User Code

- Functions
  - Compute scalar result
  - Plug into query
- Procedures
  - Perform business operations
  - Plug into trigger or call from application

Function Placement

```
select computeAge( birthday ), lastName
from Student
where computeAge( birthday ) > 5
order by computeAge( birthday ), lastName


select birthday, count(*)
from Student
group by birthday
having computeAge( birthday ) > 10
```

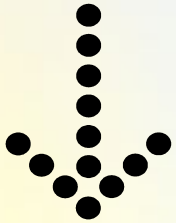
Procedure Placement

call **ScoreTestTaking**(takingID)

```
create trigger ScoreTestWhenDone
after update of takingDate
on TestTaking
referencing new as testTakingRow
for each row mode db2sql
call ScoreTestTaking( testTakingRow.takingID )
```

Triggered Procedure

Update table (client side)



Fires trigger (server-side)

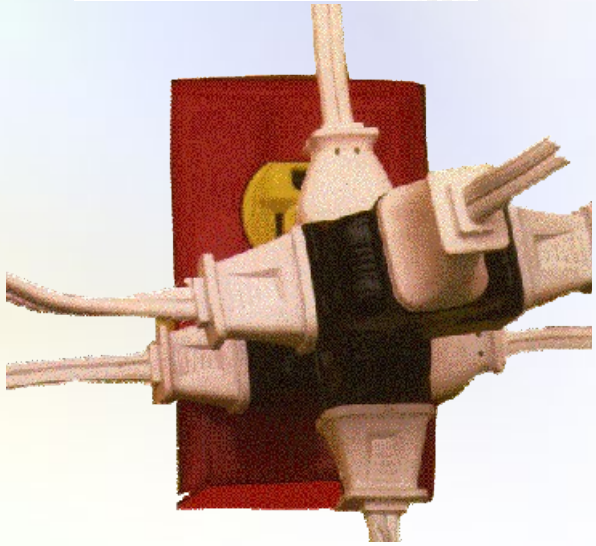
↳ Calls procedure (server-side)

↳ Loops through many rows (server-side)

↳ Updates table again (server-side)

Why?

- Integrity
- Performance
- Integration



Integrity

- Push business logic close to data
- Enforce business rules in one place

Integrity Example

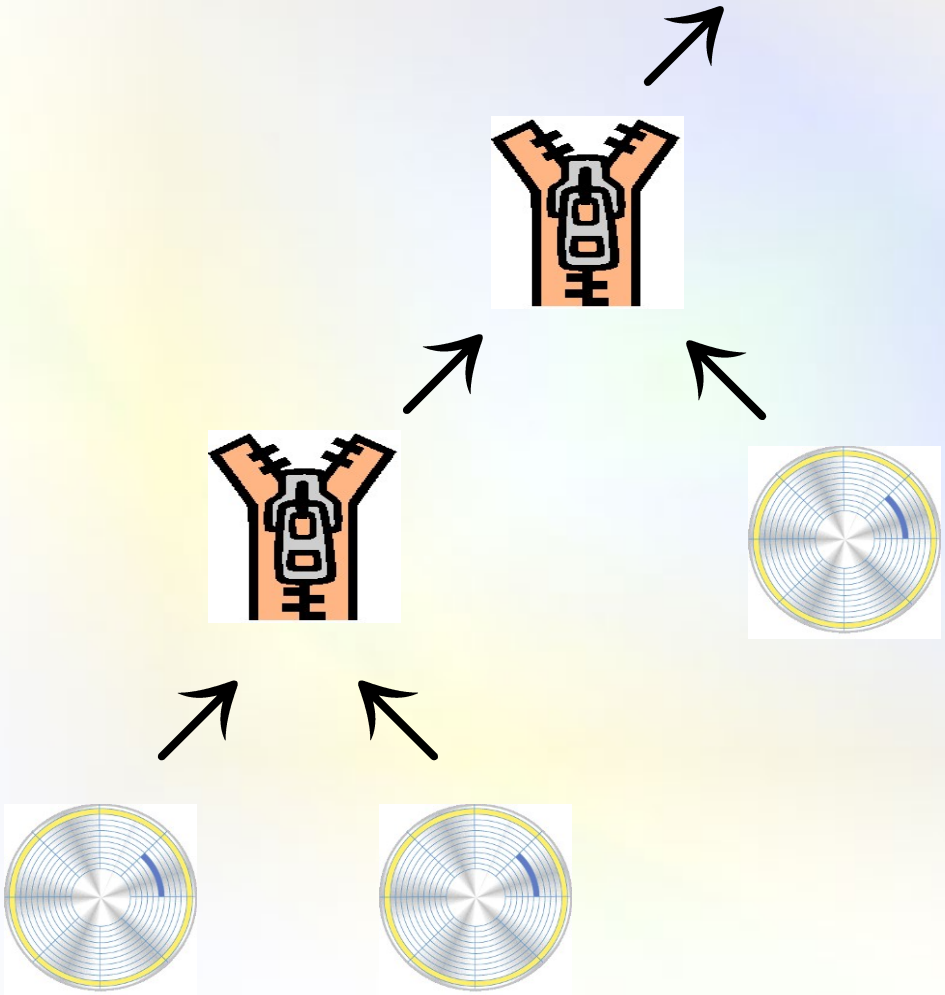
```
create table QuestionTaking
(
    questionID   int not null references Question( questionID ),
    takingID     int not null references TestTaking( takingID ),
    actualChoice int not null,

    unique( questionID, takingID ),
    check ( vetChoice( actualChoice, questionID ) > 0 )
)
```

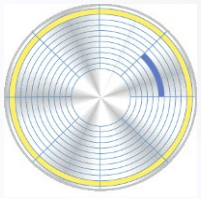
Performance

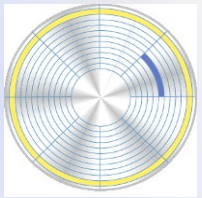
- Filter out noise early on
- Reduce query execution time
- Reduce network traffic

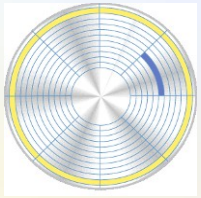
Query Tree



Filtered Tree

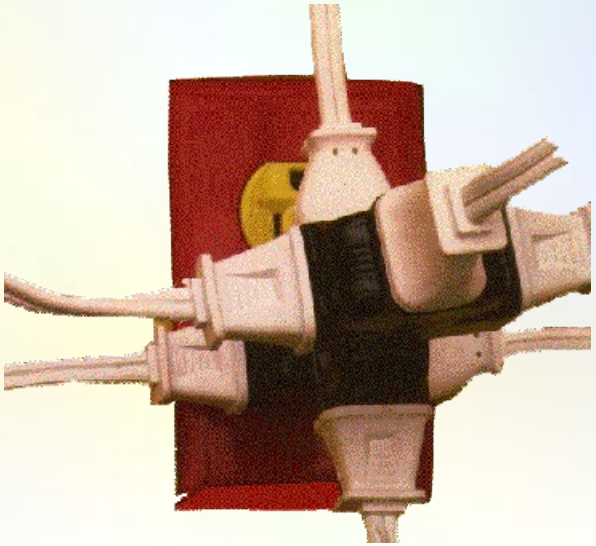






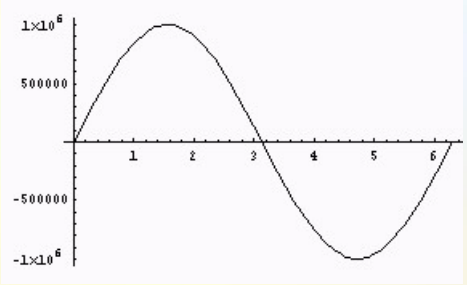
Integration

- Re-use existing freeware libraries
- Join with external data



ANSI SQL

- User-written functions
- User-written procedures



Why Java?

- Expressive
- Good exception handling

Why Java?

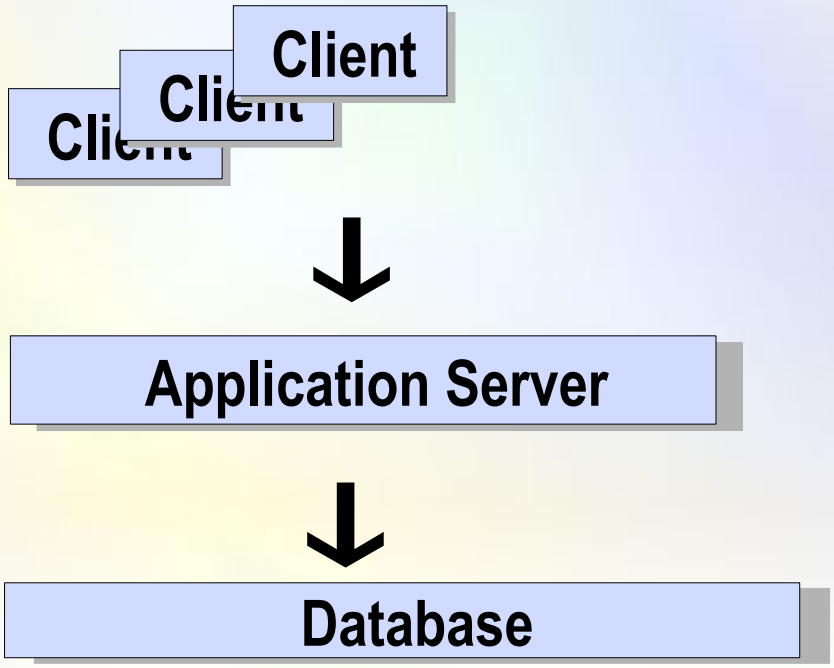
- Portable datatypes

Why Java?

- Large body of available freeware
- Off-the shelf tool support, including debuggers

Java Everywhere

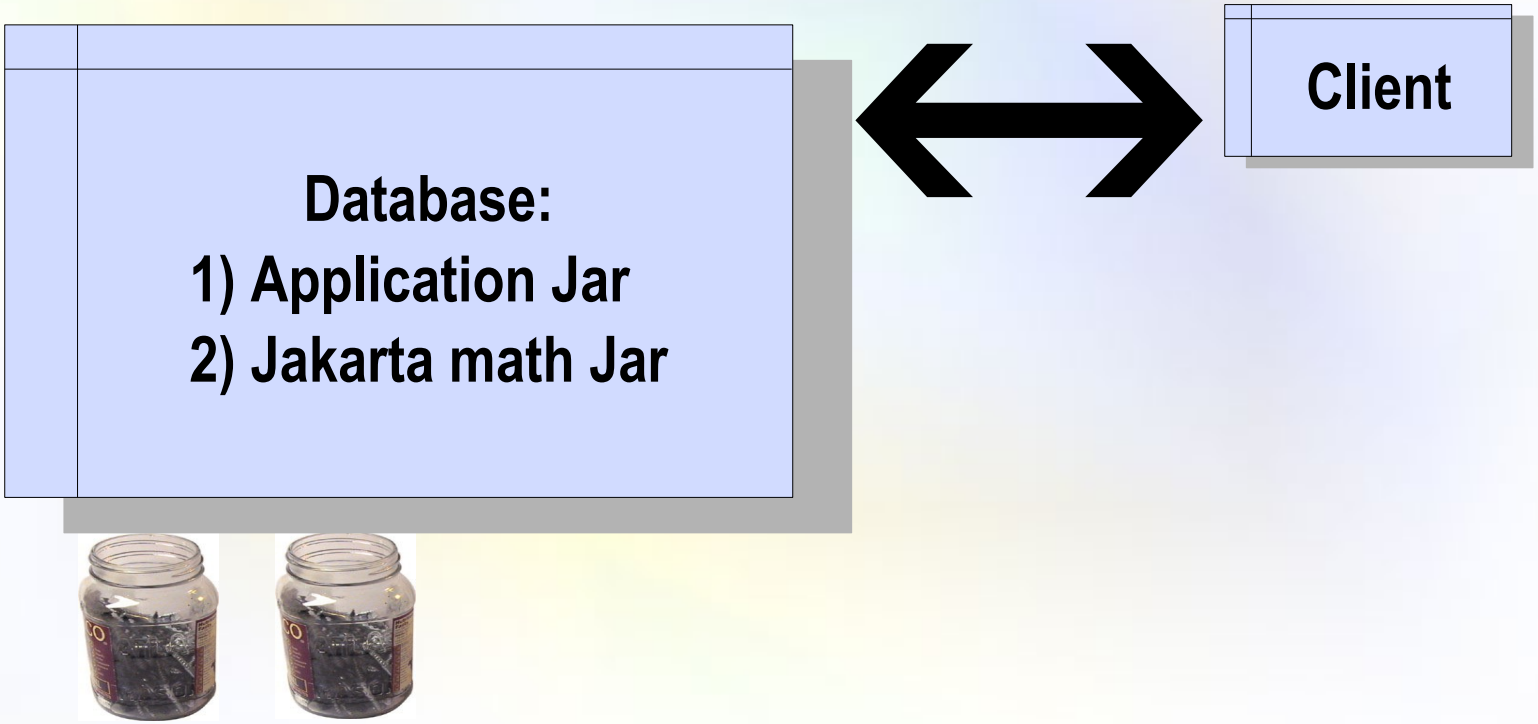
- Same code can run in client, middle tier, and server



Supporting Databases

- Derby
- Postgres
- Oracle
- DB2
- Sybase

Jars in Database



Derby References

- Developer's Guide
  - “Loading classes from a database”
  - “Derby server-side programming”

Demo Concepts

- Educational testing application
- Students, Schools, Tests, Questions, Takings