Introduction to MySQL

Database Systems
Presented by Rubi Boim
Agenda

- Bureaucracy…
- Database architecture overview
- Buzzwords
- SSH Tunneling
- Intro to MySQL
- Comments on homework
Homework #1

- Submission date is on the website.. (No late arrivals will be accepted)

- Work should be done in pairs

- Please, please, please, names and ID on the submittals.

- Submit Hardcopies to Rubi’s mailbox

- USE THE FORMAT DESCRIBED IN THE ASSIGNMENT
Project

- Hard work, but real.
- Work in groups of 4
- Project goal: to tackle and resolve real-life DB related development issues
- One Two stages.
- Use JAVA (SWT)

- Thinking out of the box will be rewarded
Agenda

- Bureaucracy...
- Database architecture overview
- Buzzwords
- SSH Tunneling
- Intro to MySQL
- Comments on homework
DB System from lecture #1

“Two tier database system”

Data files

Database server (someone else’s C program)

connection
(ODBC, JDBC)

Applications
1,2,3 tiers

Client Applications -> JDBC -> Data Source

Client Tier
Application Server Tier
Infrastructure Tier
Abstractly (DB) system **layers** may include:

- Application
- DB infrastructure
- DB driver
- Transport
- DB engine
- Storage
Why?

- DB programmer
- App programmer
- Gui designer
- Tester
- DBA
- DB programmer
Why should it actually use database?
+ Persistence layer
+ Access data storage
+ Interfacing between systems
+ Large volumes
+ Scalability
+ Redundancy
Infrastructure layer

- Goals:
  + Database “hiding”
  + Schema abstraction
  + Encapsulation of db mechanisms

- How: (In two words)
DB driver / bridge

- Used for:
  - API for database connectivity
  - Protocol converter
  - Performance improvements
  - Transaction management

- Examples:
  - In a minute…
Transport

- Mainly TCP but not only
- Secure
- Efficient
- Fast but not fast enough
DB engine

- Total management of the DB environment including
  - Security
  - Scalability
  - Fault tolerant (disaster management)
  - Monitoring
  - Services

- Large DB engines include Microsoft SQL Server, Oracle, SyBase, MySQL, etc.
DB engine (2)

DB engine management includes:

+ Databases/Tables/Fields
  Creation/removal/modification/optimization
+ Connections/Users/Roles
  Security/monitoring/logging
+ Jobs/Processes/Threads
  Scheduling/balancing/managing
NAS/SAN, Raid and other stuff…
(sorry… not in this course)
Agenda

- Bureaucracy…
- Database architecture overview
- Buzzwords
- SSH Tunneling
- Intro to MySQL
- Comments on homework
Terms...

- ODBC
- ADO
- OLE-DB
- MDAC/UDA
- JDBC
- ORM
Various standards have been developed for accessing database servers.

Some of the important standards are

- **ODBC** (Open Database Connectivity) is the early standard for relational databases.
- **OLE DB** is Microsoft’s object-oriented interface for relational and other databases.
- **ADO** (Active Data Objects) is Microsoft’s standard providing easier access to OLE DB data for the non-object-oriented programmer.
Open Database Connectivity (ODBC) is a standard software API method for using database management systems (DBMS)

- Maximum interoperability
Examples of common tasks:

+ Selecting a data source and connecting to it.
+ Submitting an SQL statement for execution.
+ Retrieving results (if any).
+ Processing errors.
+ Committing or rolling back the transaction enclosing the SQL statement.
+ Disconnecting from the data source.
UDA (Universal Data Access) and/or MDAC (Microsoft Data Access Components) include (ADO), OLE DB, and (ODBC).
JDBC

- Java DB connectivity API
- Similar to ODBC
- Why do you need it:
  + Pure Java
  + Simple API
  + Well.....Multi-platform
API includes:
+ DriverManager, Connection, Statement, PreparedStatement, CallableStatement, ResultSet, SQLException, DataSource

JDBC Type Driver:
+ Type 1 - (JDBC-ODBC Bridge) drivers.
+ Type 2 - native API for data access which provide Java wrapper classes
+ Type 3 - 100% Java, makes use of a middle-tier between the calling program and the database.
+ Type 4 - They are also written in 100% Java and are the most efficient among all driver types. Calls directly into the vendor-specific database protocol.
JDBC Types

Type 1

Type 2

Type 3

Type 4
Object-Relational mapping is a programming technique for converting data between incompatible type systems in relational databases and object-oriented programming languages.

For example: Hibernate
Agenda

- Bureaucracy…
- Database architecture overview
- Buzzwords
- SSH Tunneling
- Intro to MySQL
- Comments on homework
Connecting...

You need:

- **IP**
- **Port**

- **Home install:** IP=localhost
- **TAU’s server:** IP=mysqlsrv.cs.tau.ac.il

- **MySQL default port is 3306**

*is it really that easy??*
Welcome to

The travels of a query
**Standard way**

- Application
- DB infrastructure
- DB bridge/driver
- Transport (TCP)
- DB engine

**Using Tunnel**

- Application
- DB infrastructure
- DB bridge/driver
- Proxy machine
- Tunnel machine (SSH server)
- DB engine
- Server machine

**Transport (TCP)**
SSH in TAU

Application
DB infrastructure
Db bridge/driver
proxy
Tunnel machine (SSH server)
DB engine

YOUR MACHINE
define DB at localhost, port 3305
Putty connects to nova and forward local port 3305 to mysqlsrv.cs.tau.ac.il port 3306
Nova.cs.tau.ac.il
SSH in TAU

- Putty

![PuTTY Configuration](image_url)
Don’t forget to

- CHECK THE CONNECTION GUIDE!!
  (course website)
Agenda

- Bureaucracy…
- Database architecture overview
- Buzzwords
- SSH Tunneling
- Intro to MySQL
- Comments on homework
Products we will be using

- MySQL (Community Server – Home)
- MySQL (Enterprise Edition – TAU)
- MySQL Workbench (GUI Tool..)
- MySQL Connector (J) – In two weeks…

Free to download on [www.mysql.com](http://www.mysql.com)
TAU Server settings..

- You can create your own user (schema) by following the connection guide link (course website..)

- For the project, each group will get a `special` user (schema)
“Sakila” Schema (For hw1)

- We will use the “Sakila” schema

- Install and download from
  [http://dev.mysql.com/doc/index-other.html](http://dev.mysql.com/doc/index-other.html)

- Already installed on TAU’s server:
  - username: sakila
  - password: sakila
  - schema: sakila
MySQL Command

How to run:
http://www.cs.tau.ac.il/system/faq/development/databases/mysql2

→ mysql -u sakila -h mysqlsrv.cs.tau.ac.il sakila –p

Common commands:
- “show databases;”
- “show tables;”
- “select.. ;”

→ Don’t forget the ;
Install MySQL at Home

- MySQL Community Server
  http://www.mysql.com/downloads/mysql/

- MySQL Workbench
  http://www.mysql.com/downloads/workbench/

- (You might need to download Microsoft Visual C++ 2010 Redistributable Package)
MySQL Workbench

Installation only at home…
Demo Time 😊

- Startup the Server..
Demo Time 😊

- Server Administration
  - run the local instance
  - create users
  - export/import
Demo Time 😊

- SQL Development
  - browse the schema
  - create/alter tables
  - run queries
  - export results
Demo Time 🎉

- Install the “sakila” schema
Demo Time 😊

- Data Modeling
  → browse / alter the schema
phpMyAdmin
Another tool for managing MySQL
 Installed on tau, and reachable from home without a tunnel!

https://www.cs.tau.ac.il/phpmyadmin/index.php
(note the https)

To install at home, download from:
http://www.phpmyadmin.net/
(requires php server so its not recommended unless you are familiar with these stuff...)
<table>
<thead>
<tr>
<th>Table</th>
<th>Action</th>
<th>Records</th>
<th>Type</th>
<th>Collation</th>
<th>Size</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>actor</td>
<td></td>
<td>200</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>32.0 K</td>
<td></td>
</tr>
<tr>
<td>actor_info</td>
<td></td>
<td>~0°</td>
<td>View</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>address</td>
<td></td>
<td>602</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>96.0 K</td>
<td></td>
</tr>
<tr>
<td>category</td>
<td></td>
<td>16</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>16.0 K</td>
<td></td>
</tr>
<tr>
<td>city</td>
<td></td>
<td>600</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>64.0 K</td>
<td></td>
</tr>
<tr>
<td>country</td>
<td></td>
<td>105</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>16.0 K</td>
<td></td>
</tr>
<tr>
<td>customer</td>
<td></td>
<td>590</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>120.0 K</td>
<td></td>
</tr>
<tr>
<td>customer_list</td>
<td></td>
<td>~0°</td>
<td>View</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>film</td>
<td></td>
<td>1,002</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>272.0 K</td>
<td></td>
</tr>
<tr>
<td>film_actor</td>
<td></td>
<td>6,462</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>272.0 K</td>
<td></td>
</tr>
<tr>
<td>film_category</td>
<td></td>
<td>1,000</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>80.0 K</td>
<td></td>
</tr>
<tr>
<td>film_list</td>
<td></td>
<td>~0°</td>
<td>View</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>film_text</td>
<td></td>
<td>1,000</td>
<td>MyISAM</td>
<td>utf8_general_ci</td>
<td>317.8 K</td>
<td></td>
</tr>
<tr>
<td>inventory</td>
<td></td>
<td>4,991</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>284.0 K</td>
<td></td>
</tr>
<tr>
<td>language</td>
<td></td>
<td>6</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>16.0 K</td>
<td></td>
</tr>
<tr>
<td>nicer_but_slower_film_list</td>
<td></td>
<td>~0°</td>
<td>View</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>payment</td>
<td></td>
<td>16,049</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>2.1 MiB</td>
<td></td>
</tr>
<tr>
<td>rental</td>
<td></td>
<td>16,044</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>2.7 MiB</td>
<td></td>
</tr>
<tr>
<td>sales_by_film_category</td>
<td></td>
<td>~0°</td>
<td>View</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>sales_by_store</td>
<td></td>
<td>~0°</td>
<td>View</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>staff</td>
<td></td>
<td>2</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>98.0 K</td>
<td></td>
</tr>
<tr>
<td>staff_list</td>
<td></td>
<td>~0°</td>
<td>View</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>store</td>
<td></td>
<td>2</td>
<td>InnoDB</td>
<td>utf8_general_ci</td>
<td>40.0 K</td>
<td></td>
</tr>
</tbody>
</table>
Agenda

- Bureaucracy…
- Database architecture overview
- Buzzwords
- SSH Tunneling
- Intro to MySQL
- Comments on Homework
“Sakila” Schema

- We will use the “Sakila” schema

- Install and download from
  http://dev.mysql.com/doc/index-other.html

- Already installed on TAU’s server:
  username: sakila
  password: sakila
  schema: sakila
Homework Notes

- SQL functions and arithmetic conditions.
- ‘strings’
- LIKE (%), LOWER
- Use the Syntax help in Query browser
- MAX, MIN
- IN
Thank you 😊