lecture 2
lab 2
branches, databases
MVC
CodeIgniter

[Diagram showing the flow of operations: Index.php -> Routing -> Security -> Application Controller -> View, with additional components like Caching, Drivers, Models, Libraries, Helpers, Packages, Scripts connected to the Application Controller.]
controllers

http://example.com/[controller-class]/[controller-method]/[arguments]
database design
jharvard@appliance (~): mysql -u jharvard -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 40
Server version: 5.5.19 MySQL Community Server (GPL)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
SELECT Employees.Name, Orders.Product
FROM Employees, Orders
WHERE Employees.Employee_ID=Orders.Employee_ID
SELECT Employees.Name, Orders.Product
FROM Employees, Orders
WHERE Employees.Employee_ID=Orders.Employee_ID
SELECT Employees.Name, Orders.Product
FROM Employees
JOIN Orders ON Employees.Employee_ID=Orders.Employee_ID

http://www.w3schools.com/sql/sql_join.asp
indexes

- PRIMARY
- INDEX
- UNIQUE
- FULLTEXT
<table>
<thead>
<tr>
<th>Feature</th>
<th>MyISAM</th>
<th>Memory</th>
<th>InnoDB</th>
<th>Archive</th>
<th>NDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage limits</td>
<td>256TB</td>
<td>RAM</td>
<td>64TB</td>
<td>None</td>
<td>384EB</td>
</tr>
<tr>
<td>Transactions</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Locking granularity</td>
<td>Table</td>
<td>Table</td>
<td>Row</td>
<td>Table</td>
<td>Row</td>
</tr>
<tr>
<td>MVCC</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Geospatial data type support</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Geospatial indexing support</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>B-tree indexes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Hash indexes</td>
<td>No</td>
<td>Yes</td>
<td>No[a]</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Full-text search indexes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Clustered indexes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Data caches</td>
<td>No</td>
<td>N/A</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Index caches</td>
<td>Yes</td>
<td>N/A</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Compressed data</td>
<td>Yes[b]</td>
<td>No</td>
<td>Yes[c]</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Encrypted data [d]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Cluster database support</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Replication support [e]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Foreign key support</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Backup / point-in-time recovery [f]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Query cache support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Update statistics for data dictionary</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
race conditions
atomicity

INSERT INTO table (id, symbol, shares) VALUES(7, 'AFLB.OB', 10)
ON DUPLICATE KEY UPDATE shares = shares + VALUES(shares)
locks

LOCK TABLES account WRITE;
SELECT balance FROM account WHERE number = 2;
UPDATE account SET balance = 1500 WHERE number = 2;
UNLOCK TABLES;
transactions

START TRANSACTION;
UPDATE account SET balance = balance - 1000 WHERE number = 2;
UPDATE account SET balance = balance + 1000 WHERE number = 1;
COMMIT;
transactions

START TRANSACTION;
UPDATE account SET balance = balance - 1000 WHERE number = 2;
UPDATE account SET balance = balance + 1000 WHERE number = 1;
SELECT balance FROM account WHERE number = 2;
# suppose account #2 has a negative balance!
ROLLBACK;
PDO

$dsn = 'mysql:dbname=jharvard_lecture2;host=127.0.0.1';
$user = 'jharvard';
$password = 'crimson';

try{
    $dbh = new PDO($dsn, $user, $password);
}
catch (PDOException $e)
{
    echo 'Connection failed: ' . $e->getMessage();
}
to be continued...