Transactions and Protection

Cascading

- Foreign key relationships
- What if you want to delete a parent?

What Could Possibly Go Wrong?

- Enforced referential integrity
- Triggers
- Cascades
- Stored procedures
- Calamity
- Knuckleheadedness

Basic MySQL Syntax

- START TRANSACTION or BEGIN begins a new transaction.
- COMMIT follows through on the current transaction, making its changes permanent.
- ROLLBACK rolls back the current transaction, canceling its changes.

Sample ATM Transactions

- START TRANSACTION
  - Deduct $200 from my savings account
  - Add $200 to my checking account
  - No problems reported from the MySQL server
- COMMIT the current transaction

- START TRANSACTION
  - Deduct $100 from checking for tonight's date
  - ATM whirrs and clicks. Produces nothing. ERROR
- ROLLBACK cancels the MySQL transaction.

```
-- start a new transaction
start transaction;

-- remove $200 from savings account
UPDATE Accounts
SET Savings = (Savings - 200)
WHERE CustomerNumber = 23456;

-- add $200 to checking account
UPDATE Accounts
SET Checking = (Checking + 200)
WHERE CustomerNumber = 23456;

-- commit changes
commit;
```

```
try {
// Begin a transaction
    $db->beginTransaction();

// Run queries; if one fails, an exception should be thrown

$db->query('UPDATE Accounts SET Savings = (Savings - 200) WHERE
CustomerNumber = 23456;');

$db->query(UPDATE Accounts SET Checking = (Checking + 200) WHERE
CustomerNumber = 23456;');

// If no “exception was thrown” (no query failed), we commit the transaction
    $db->commit();
} catch (Exception $e) {

// Something went wrong.  Must rollback the transaction
    $db->rollback();
}
```

Concurrent Users

Concurrency

- Lock Granularity
  - Record and field locking
  - File, block, and database locking
- Read and write locks
- Try to lock, wait, lock, unlock.
- Timestamp… Grab a number…

Table Locks

- Some transactions require exclusive access to the entire database
  - Global queries
  - Global updates
  - Data dumps
  - Synchronizations
  - Backups

Database Logs and Backups

- The server can be configured to keep a log of every transaction
- Resets logs after a successful backup
- The DBMS sets checkpoints
- More common: backup
  - Both the database and logs
  - Multiple backups in multiple sites
- LOCKSS