CONESTOGA
Connect Life and Learning

WHAT YOU DO HERE... COUNTS OUT THERE

Active Record

(PROG1730 #1 (12F) Database Design)

Rich Hildred

rhildred@conestogac.on.ca

519-594-0900

2D04 (Waterloo Campus)

Objectives

- Introduce the idea of design patterns
  - Adaptor
  - Factory Method
  - Domain Model
  - Active Record
  - Approval
  - Oauth

Objectives (continued)

- Object to Relational Mapping
  - Semantics
  - ObjectID
- Security
  - 7 Functional Users

Design Patterns

- Christopher Alexander 1977-1979
- Ward Cunningham 1987
- Martin Fowler – 2002
- Structural
- Creational
- Behavioral

Adaptor

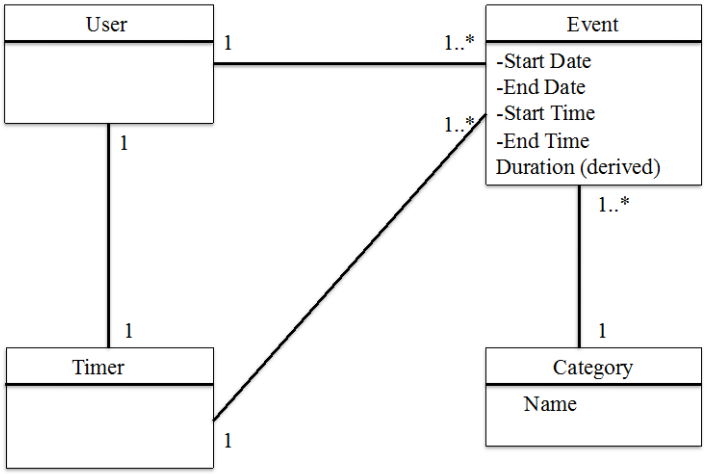
- Type: Structural
- What it is:
  - Convert the interface of a class into another interface that clients expect. Lets classes work together that otherwise couldn't because of incompatible interfaces.

Factory Method

- Type: Creational
- What it is:
  - Define an interface for creating an object, but let subclasses decide which class to instantiate. Lets a class defer instantiation to subclass.

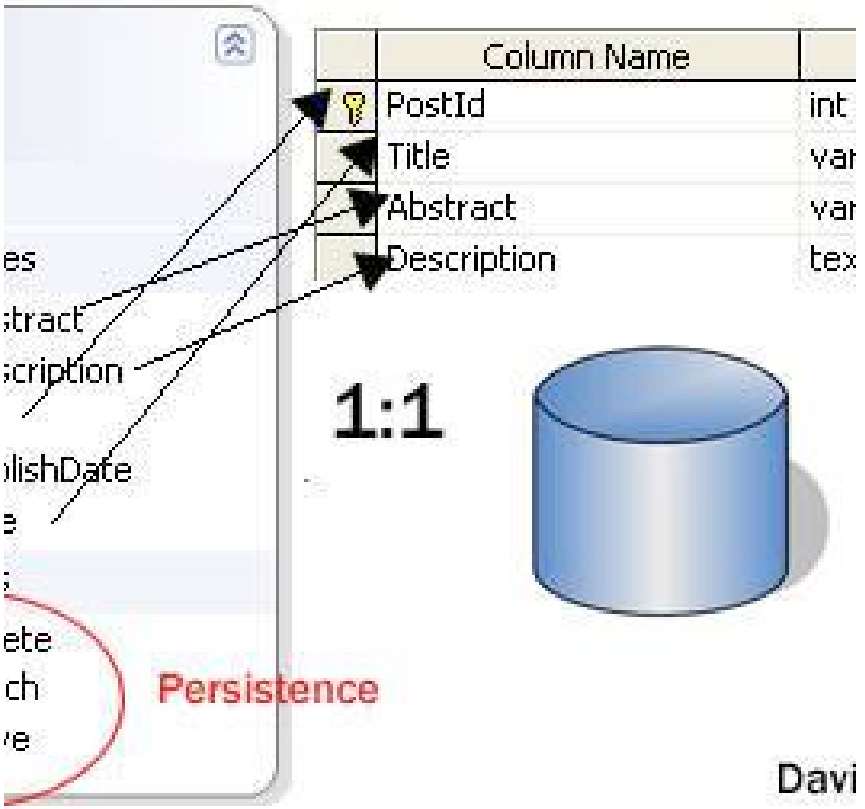
Domain Model

- Type: Structural
- What It is:
  - Define the pattern between entities in the customer's language. Lets programmers understand the customer's language using abstraction.



Active Record

- Type: Behavioral
- What it is:
  - An object that wraps a row in a database table or view, encapsulates the database access, and adds domain logic on that data.



Approval

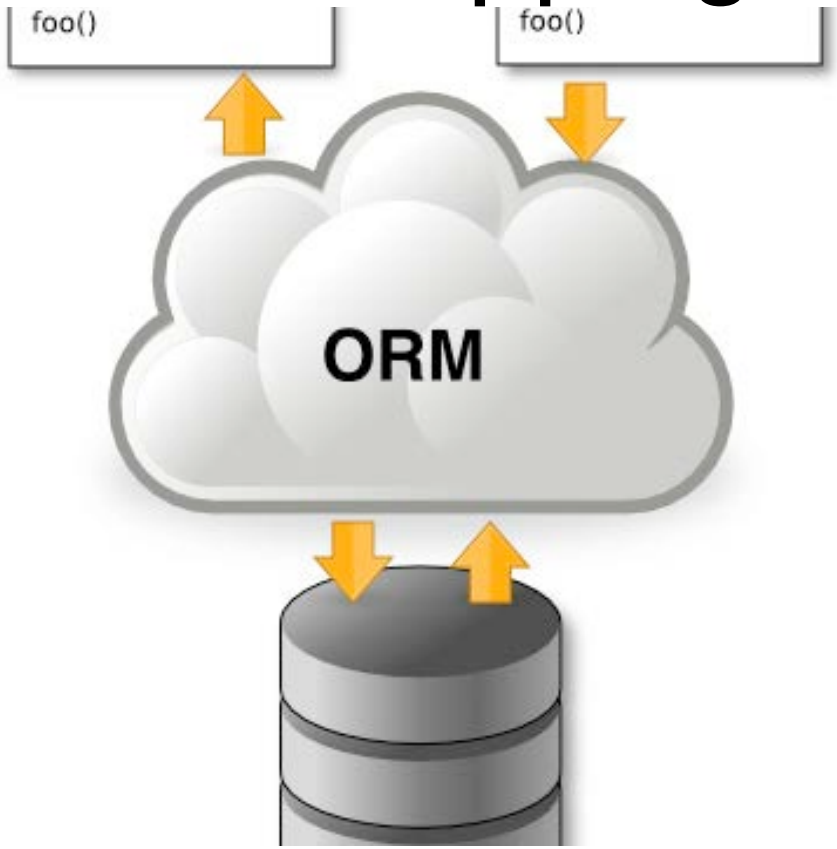
- Type: Behavioral
- What It is:
  - When someone adds you as a friend, Facebook sends you a notification prompting you to confirm or deny the request. If you approve the request, that person is added.

Oauth

- Type: Behavioral
- What It is:
  - Specifies a protocol for establishing system-to-system authorization. It allows you to approve access to your data on one system to another system acting on your behalf.

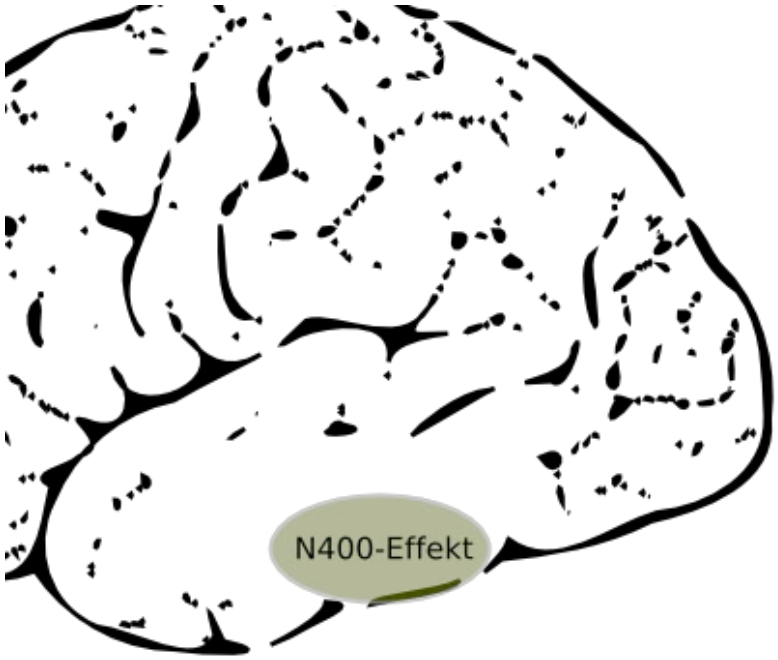
Object To Relational Mapping

- bind an object to its data in the database
- Productivity
- Application Design
- Code Reuse
- Application Maintainability



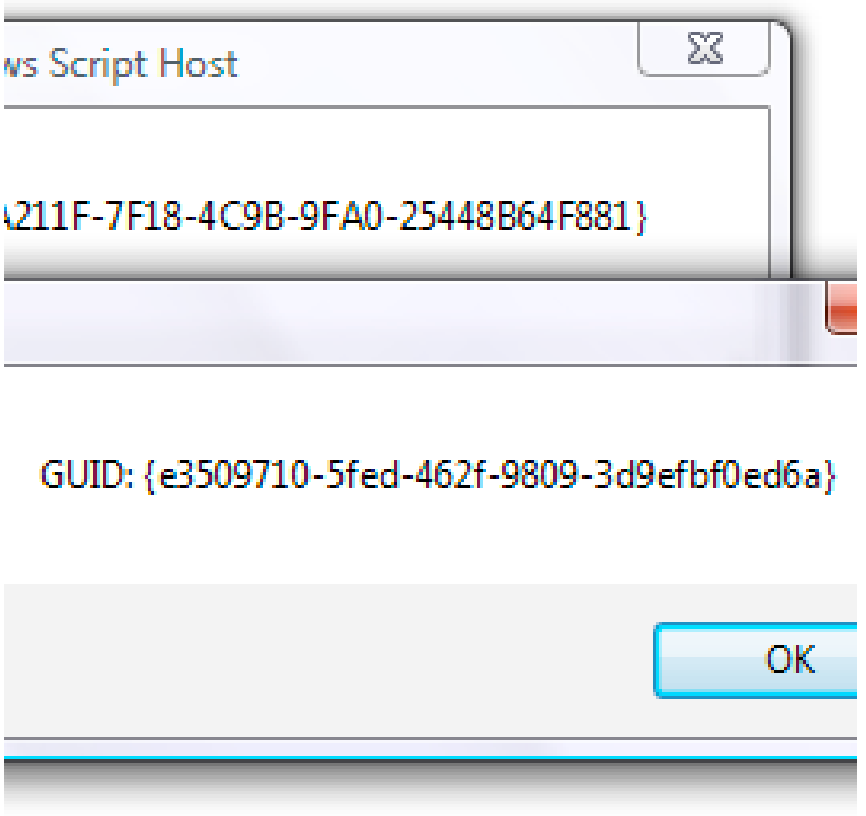
Semantics

- A class is named Singularly – Person
- A collection of objects of a single class is named in the plural – People
- A table is viewed as a collection of objects of the same class - People



ObjectId

- Characteristic of an Object Oriented Database
- Can even be a GUID
- Often implemented as a primary key auto incrementing
- By Convention “id”



In Addition to Domain Methods

	var oPerson = ActiveRecord::create('person');
C	oPerson.fname = "Richard"; oPerson.lname = "Hildred"; oPerson.save(fComplete); // fComplete is a function that gets called on completion

R	oPerson.find('lname = ?', array('Hildred'), fComplete); //fComplete is a function that gets called on completion
	oPerson.load('lname = ?', array('Hildred'), fComplete); // like find but only loads 1 object

U	oPerson.fname = "Rich"; oPerson.save(fComplete); // fComplete is a function that gets called on completion
D	oPerson.delete(fComplete); //fComplete is a function that gets called on completion

Security

- All security has to come from database
- Attackers can make your javascript do anything

7 functional users

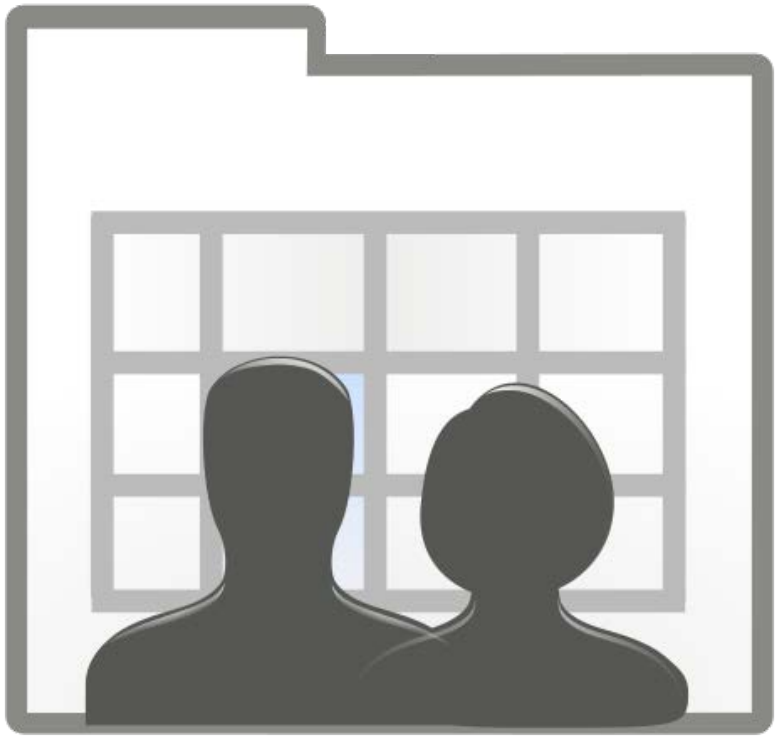
CREATE USER	GRANT
ReadOnly	GRANT SELECT on tables that anyone should be able to see
LoggedInReadOnly	GRANT SELECT on tables that a logged in user can see
AdminReadOnly	GRANT SELECT on tables that an admin user can see
LoggedInUpdate	GRANT INSERT and UPDATE to tables that anyone that is logged in can update
AdminUpdate	GRANT INSERT and UPDATE to tables that only an admin can update

7 functional users (continued)

CREATE USER	GRANT
LoggedInDelete	GRANT DELETE on tables that a logged in user can delete from
AdminDelete	GRANT DELETE on tables that only an admin user can delete from

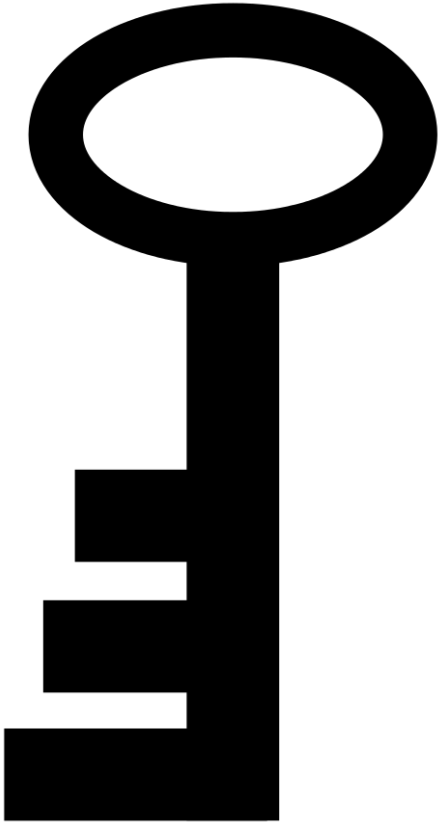
Views

- Views create an object in the database that a USER can be GRANTed privileges to.



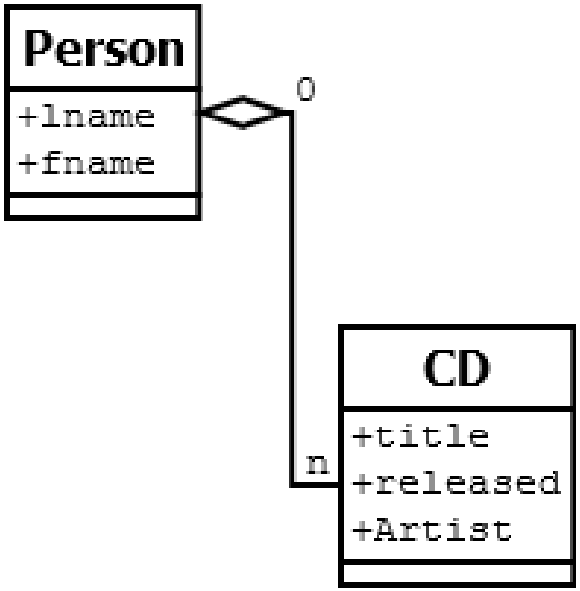
Entity Integrity

- Extended by Active Record pattern to say that each object must have a unique id that is also the PRIMARY KEY in the database
- By convention called “id”



Relational Integrity

- Pretty much guaranteed that bad data will be attempted
- Relational Integrity is an important protection against a cd belonging to someone who doesn't exist



Legal Values Integrity

- SQL
  - **CHECK** clause enforces legal-values integrity
  - Example:

```
CHECK (CreditLimit IN
(5000, 7500, 10000,
15000))
```