Database Fundamentals

presented to NYPHP
May 22, 2007

by Kenneth Downs
Secure Data Software, Inc.
ken@secdat.com
www.secdat.com
www.andromeda-project.org

Pre-Relational

In the bad old days, every program had to “know” the physical structure of every data file. No SQL! Every operation had to open files and scan record by record, or write libraries to do so.

The Modern Approach

```
<?php.
/*  MODERN   */.
$row=SQL_OneRow(.
   "SELECT code,name .
      FROM customers.
     WHERe customer=".SQLFC(gp('customer')).
);.
?>.
```

Old-Fashioned PHP

```
<?php.
/*   OLDEN DAYS */.
// open file.
$FILE=FOPEN('customers.dat','r');.
$record_length=128;.
// Look for customer.
while(true) {.
   $record=fread($FILE,$record_length);.
   $customer=substr($record,10,15);.
   // ... more code.
}.
?>.
```

The Relational Revolution

The “Relational Revolution” was based on using a mathematical model to design a system that would have predictable behavior.

Edsger Dykstra had a lot to say on “formal verification”

> *One starts with a mathematical specification of what a program is supposed to do and applies mathematical transformations to the specification until it is turned into a program that can be executed. The resulting program is then known to be correct by construction.*

Wikipedia Page: http://en.wikipedia.org/wiki/Edsger_Dijkstra

EF Codd (1 of 3) Logical Propositions

EF Codd is the father of the relational model. He was looking for a way to organize and store logical propositions:

Propositions:

- My car is blue
- Your car is red

A “Relation”

Owner	Color
Me	blue
You	red

EF Codd (2 of 3) Relational Math

Using either *Relational Algebra* or *Relational Calculus*, you can describe all possible combinations of relations (or as we call them, tables).

A “projection” is only some columns, a “selection” is only some rows, and several kinds of “joins” combine two or more relations into another relation.

Terms like “set division” and “theta-join” are all defined and expressible in SQL or some other query language.

Wikipedia topics: Relational Algebra, Relational Calculus, Relational Model

EF Codd (3 of 3) The 12 Rules

Ultimately Codd defined 12 rules, and later on many more. Overall they make the *implementation* of relational ideas very powerful.

DDL and DML:

```
Create table example (
   name char(20),address varchar(30)
)

INSERT INTO example (name,address)
VALUES ('joe','123 main')
```

Catalog:

```
SELECT
  table_name,column_name
  ,type_id
  ,character_maximum_length
FROM
 information_scheme.columns
```

Review: Model vs. File Format

The Relational Model can be called a model because it describes all aspects of managing data.

A comparison would be XML, which is by comparison *merely a file format*, albeit a well-supported one.

Review: Model vs. “Good Idea”

The operations in a relational data store are completely described by a branch of mathematics, and so the behaviors are predictable.

Compare to object orientation, which is *intuitively appealing,* and may even be a good idea, and may even be rigorously developed, but which cannot be described by a formal mathematics. You can never provide an *a priori* argument for one direction or another.

Tangent 1 of 2: Code Generation

Code generation is extremely easy for databases because the DDL, DML and system catalogs always reduce down to lists of columns.

Put another way, every operation on a relational database can be reduced itself to a collection of values in a database.

Tangent 2: People Like Tables

Ken's Law: "People work well with data organized into tables."

Tables are *sufficiently general*, you don't need extra mechanisms to store all of the customer's data.

Tables are not *overly general,* you do not need some other way to show it to mere mortals so that they will understand it.

Shocker: SQL is not Relational!

Big Difference is “bags” not sets. Bags allow duplicate values, nobody has produced a truly relational system.

Will the world end? What about all that theory? Not all that bad...

A SET

First_Name	Last_Name
John	Boone
Frank	Chalmers
Jackie	Boone

A BAG

First_Name	Last_Name
John	Boone
Frank	Chalmers
John	Boone

Later Concept: ACID Transaction

A transaction is one or more SQL statements which taken together have:

Atomicity: All or nothing.
Consistency: All rules must be followed at beginning and end.
Isolation: Nobody outside of transactions sees data in “half-way” state.
Durability: Once transaction is committed, no chance of loss.

The Primary Key

The primary key is one or more columns whose values must be unique within a table. *In a truly relational system there would be no primary key, all rows would be unique across all columns.*

In the beginning of table design, ask, “What am I keeping track of?” Each thing you are keeping track of is a candidate to be a table, and then “How is it identified?” points towards the primary key.

Some have algorithms, like SSN, some are user-entered, like a customer code of “JONES” and some are assigned solely to be unique such as an Order Number.

The Foreign Key

A foreign key is a column or columns in a table whose values must match the primary key of a “parent” table.

Customer	Address
ATREIDES	123 Main Street
CORRINO	456 2nd Selusa

Order	Date	Customer	Total
1	5/1/07	ATREIDES	50.00
7	5/2/07	ATREIDES	70.00
8	4/23/07	CORRINO	25.00
11	5/3/07	CORRINO	37.50

First Normal Form

No repeating groups. No lists, and no “item_1”, “item_2”.

Employee	Day	Sales
Sax	5/1/07	50,20,75
Ann	5/1/07	37,120

Employee	Day	Sale
Sax	5/1/07	50
Sax	5/1/07	20
Sax	5/1/07	75
Ann	5/1/07	37
Ann	5/1/07	120

Second Normal Form

All non-key attributes depend upon the entire key, not just part.

Student	Class	Professor
Maya	Aeroforming	Nirgal
Arkady	Aeroforming	Nirgal
Maya	Regolithology	Nadia
Arkady	Regolithology	Nadia

Student	Class
Maya	Aeroforming
Arkady	Aeroforming
Maya	Regolithology
Arkady	Regolithology

Class	Professor
Aeroforming	Nirgal
Regolithology	Nadia

Third Normal Form

No functional dependencies. No column may depend on a non-key value.

ORDER	Item	Price	Qty	Extended
1	WINDMILL	30.00	3	90.00
1	COIL	15.00	2	30.00
1	MICROCOLEUS	115.00	10	1150.00

Constraints

A constraint is some condition that must always apply. A primary key and a foreign key are both specialized constraints. Generalized constraints might be expressed with SQL:

```
CREATE TABLE CUSTOMERS (
   COLUMN CUSTOMER CHAR(20)
  ,COLUMN DISCOUNT_PCT NUMERIC (3,2)
     CONSTRAINT PCT_CHECK (
          DISCOUNT_PCT > 0 and DISCOUNT_PCT <= 1
     )
)
```

Security

Security is defined entirely in terms of tables, and who can:

- INSERT
- UPDATE
- DELETE
- SELECT

A system that uses real database accounts and real security is immune to SQL injection.

Triggers

A trigger is code that executes when something occurs to a table. A trigger can modify incoming data, read from other tables and execute further commands on other tables.

- Before / After INSERT For Row or Statement
- Before / After UPDATE For Row or Statement
- Before / After DELETE For Row or Statement

Simplest possible encapsulation.

Triggers are the basis of all automation.

Not Appearing in this Presentation

- Stored Procedures
- Indexes
- Higher forms of normalization. Try “Domain-Key Normalization”.
- Schemas
- Database Abstraction Layers, a different word for everything
- Why do DB Purists say bad things about mySQL?

Flamewar: The E-A-V Approach

Programmer thinks: “Hey, I've got a great idea, I'll save myself the trouble of those pesky table structure updates, it will be *easier*....”

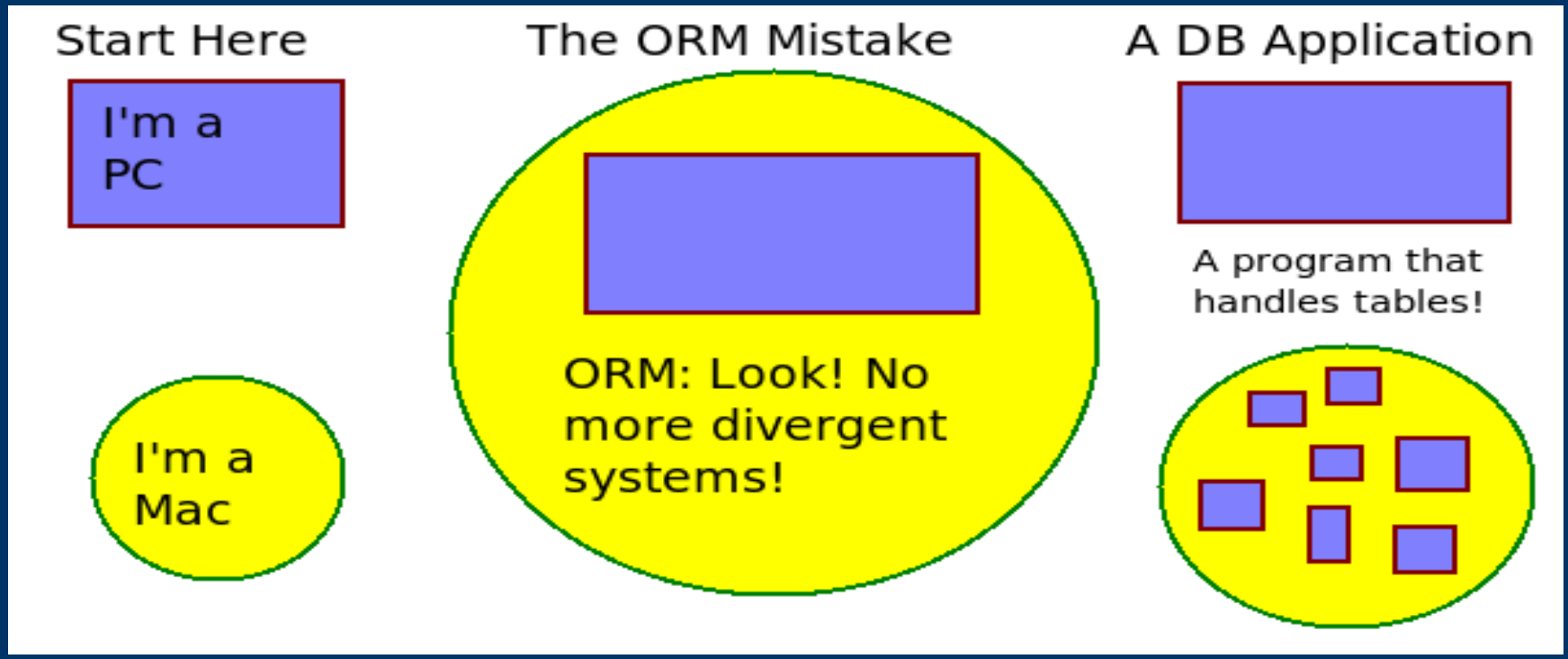
But you throw away SQL, back to slide #1.

Ken's encounters in 1998 and 2003.

Entity	Att	Value
Customer	First_Name	Arkady
Customer	Last_Name	Bogdonov
Customer	Nationality	Russian

Object Relational Mapping

We should see by now that databases and programs have two very distinct natures. The fundamental goal of ORM is to resolve them to a similar nature, which cannot be done.



Where to Put Biz Rules?

A review of the schools of thought, all of which are completely and utterly correct and true and all of which contradict each other.

1. DB Purist: Data must be fully normalized, database contains only constraints, and never contains derived data or calculations.

2. Programmer Purist: All logic belongs in my code, the database takes what I give it, normalized or not. No constraints or code in database because they don't “belong” there.

3. Maverick Positions: Databases can contain all logic, making them completely self-contained. “Lowest energy state encapsulation.”

THE END