

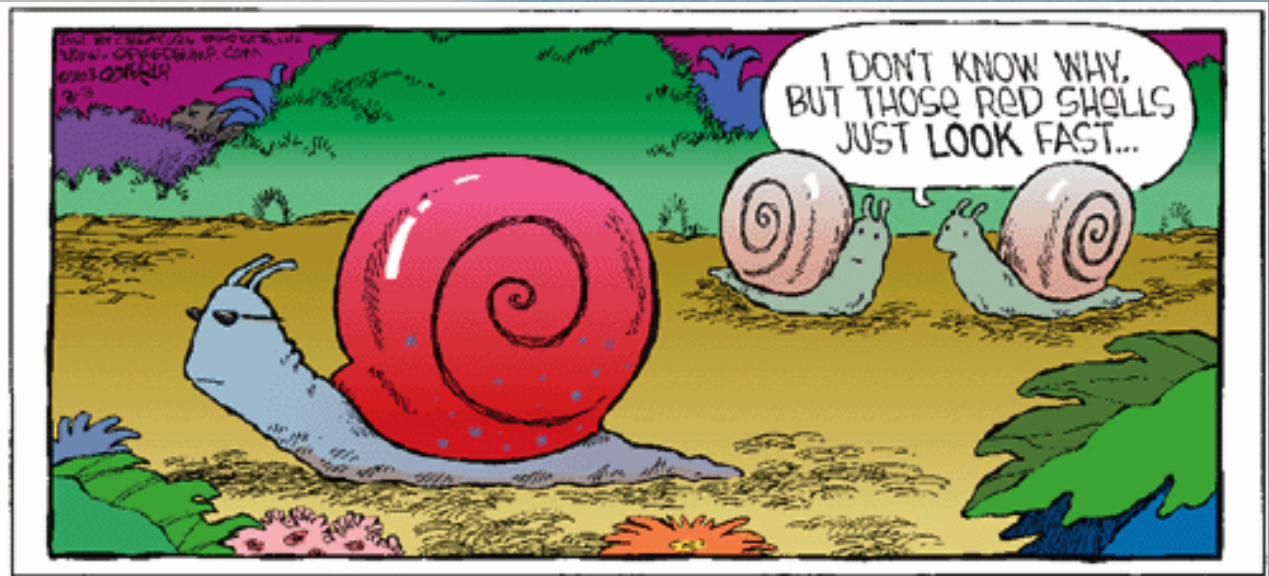
Nordic

Object Relational Database Design

Paul Nielsen

www.SQLServerBible.com

Object Databases – The Red Shell?



Copyright © 2003 Creators Syndicate, Inc.

O/R dbms –background

Attempted in '98 with SQL Server 6.5 – Material Spec System

Sprocs for inheritance were nearly impossible – reverted to Rdbms –
completed with diamond pattern

SQL Server 2K user-defined functions solve the inheritance Hierarchy
problem !

Nordic: 37,000 Feet

- **37,000 Feet**
- Class Management
- Object Management
- Association Management
- Next Steps

Comparing Data Stores Types

- Relational
 - requiring few supertype-subtypes, stable structure, simple interactions
 - for high transactions
- Data Warehouse
 - requiring aggregations
 - for reporting / analysis
- Object
 - requiring many subtypes, complex interactions
 - for inheritance / data mining

O/R dbms – the best from both worlds

Rdbms

data integrity

SQL set-based DML

relational algebra

vendor strength

performance

reporting tools

OO

OO interface

spidering associations

inheritance

business process analysis

app development

Classes, Objects, and Inheritance

Class: template or recipe for an object, similar to a relational table, with attributes

Object: an actual thing or instance of a class, similar to a relational dbms row

Inheritance: classes may inherit attributes from super-classes similar to a relational dbms supertype – subtype pattern

Classes

- General Classes
- Interface/Boundary Classes
- Process Classes
- **Entity Classes → Persistence**

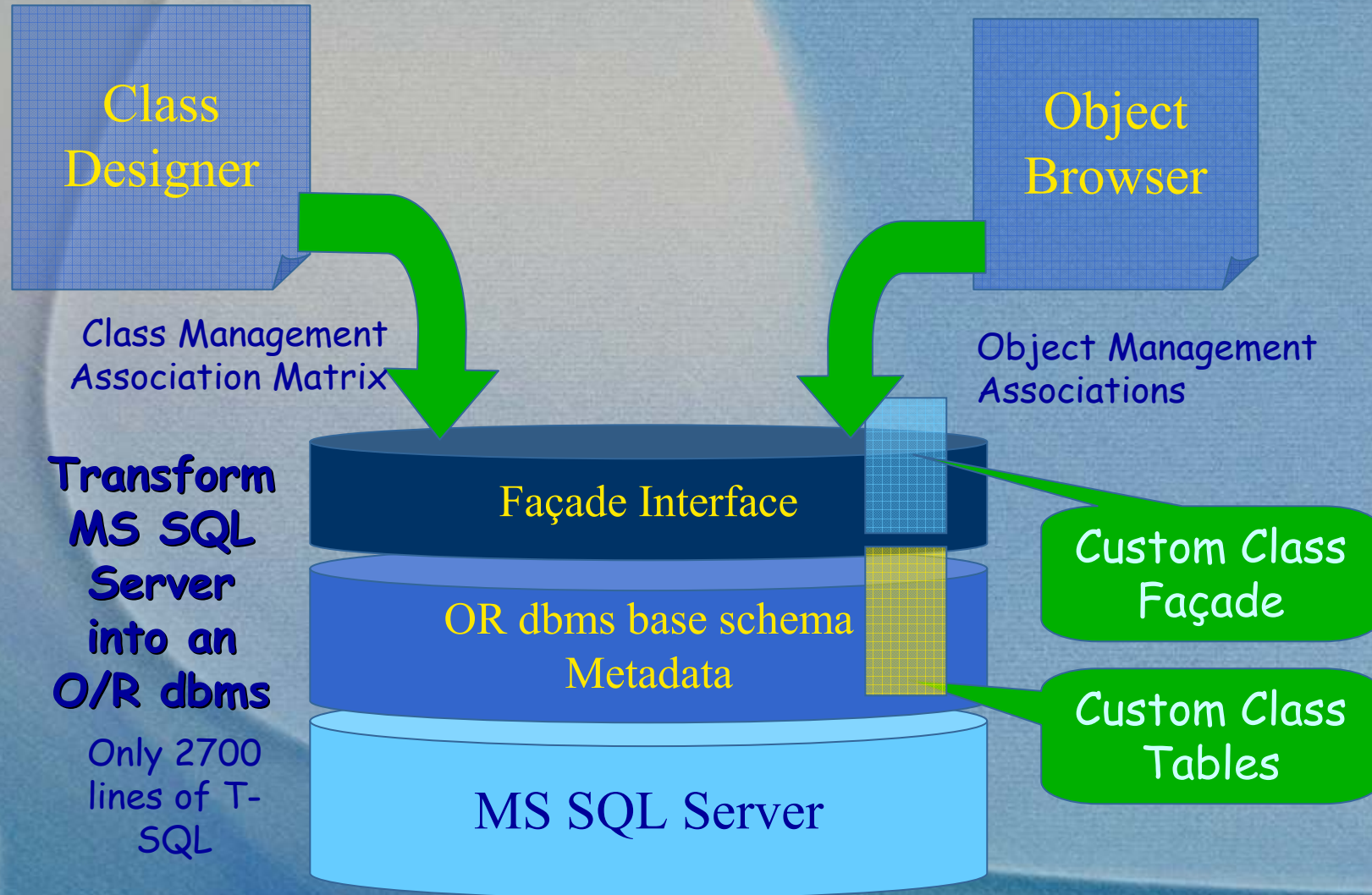
Comparing Database Terms

- Entity = Table = Class
- Tuple = Row = Object
- Attribute = Column = Property/Attribute
- Relationship = Association

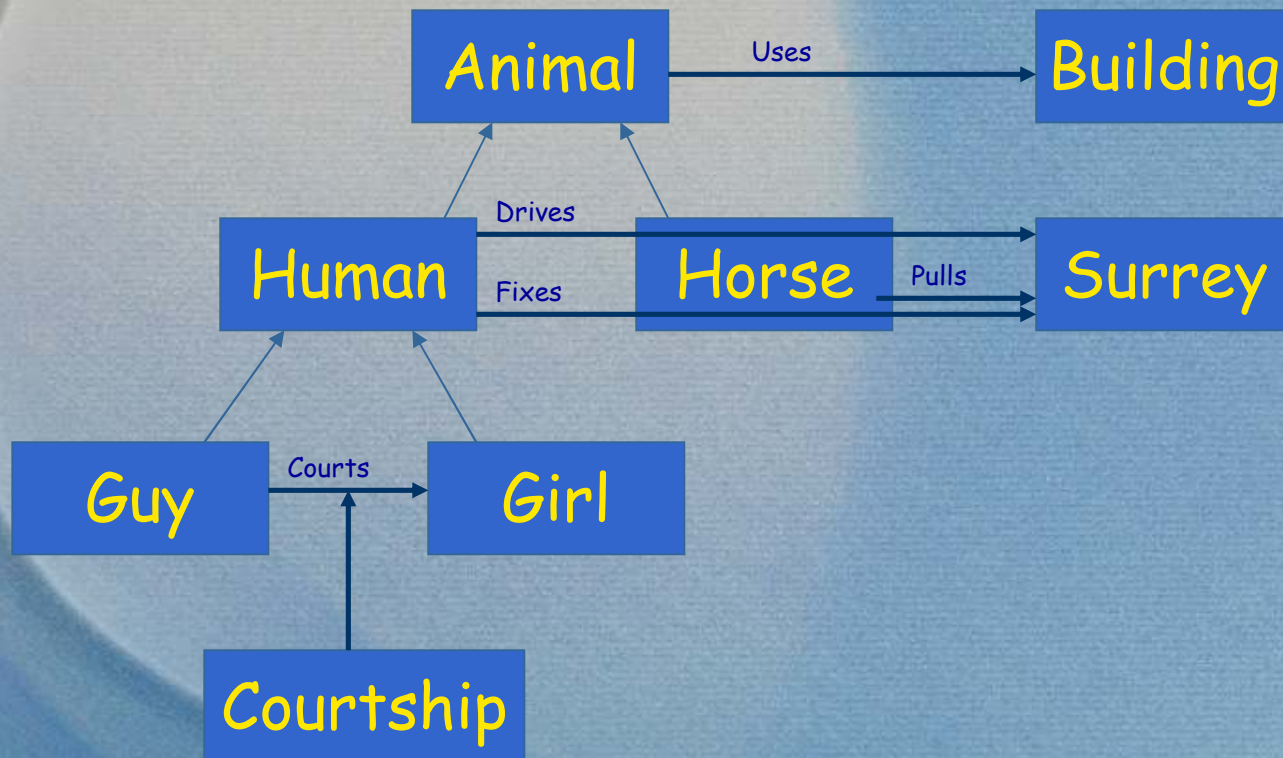
OOA/D Concepts Supported

- Pure OOdbms
 - Classes / Sub Classes
 - Domain Integrity
 - Inheritance
 - Polymorphism
 - Complex Objects
- Beyond OOdbms
 - Workflow State
 - Associations
 - Collections & Aggregations
 - Association Spidering
 - Façade Code Generation

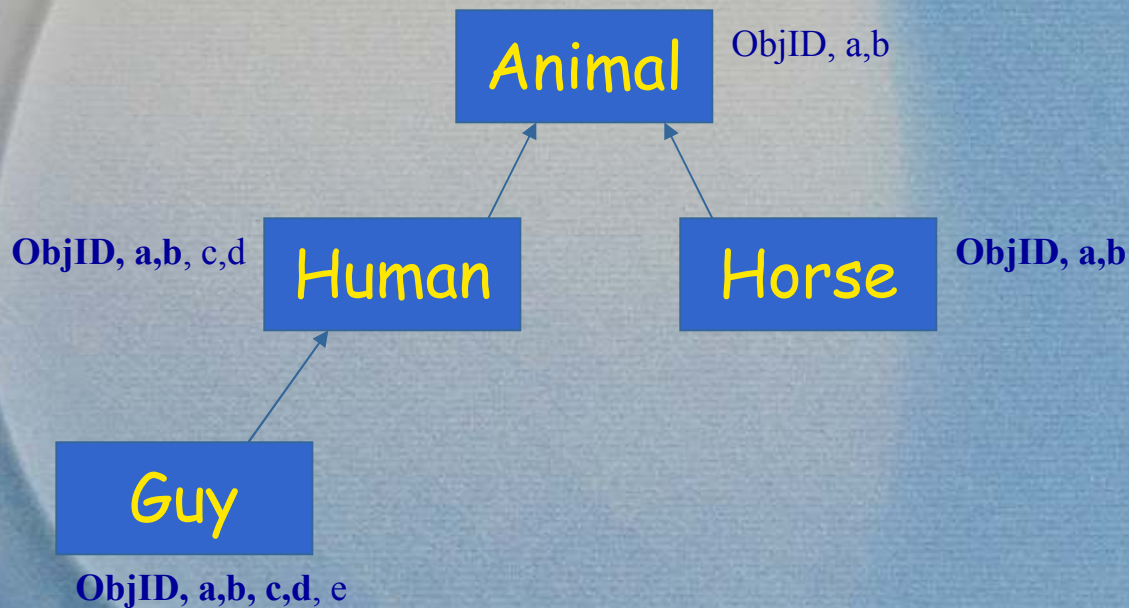
Nordic: a 37,000 feet view



Oklahoma - Sample Class Structure

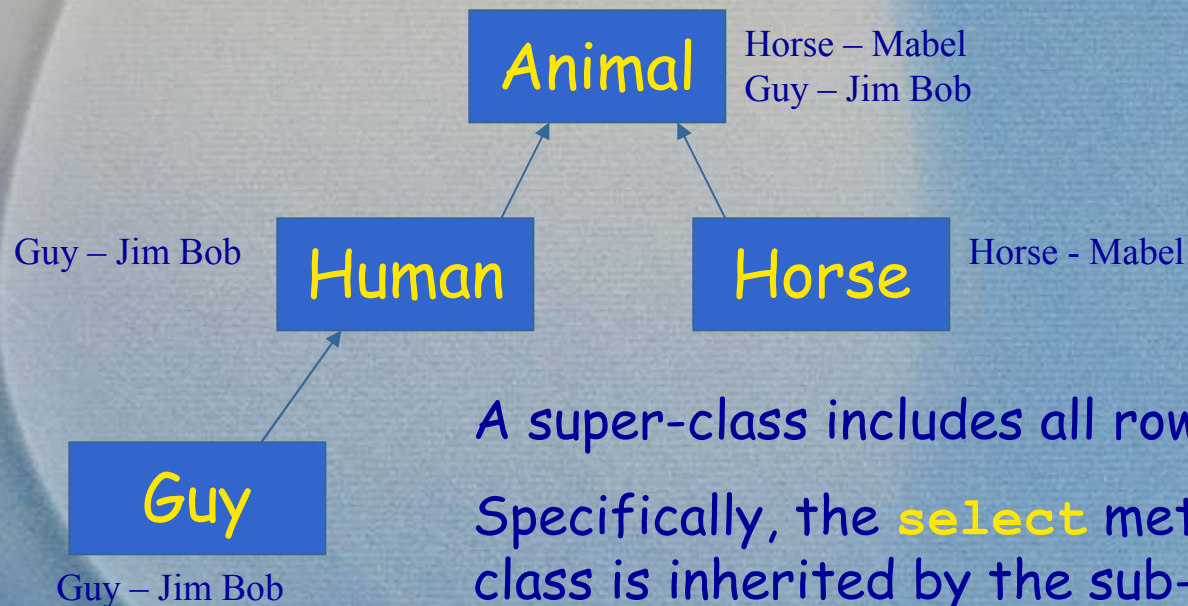


Attribute Inheritance



Subclasses **inherit** (or include) all (public) attributes from their super-class.

Object Inheritance - Polymorphism



A super-class includes all rows of sub-classes.

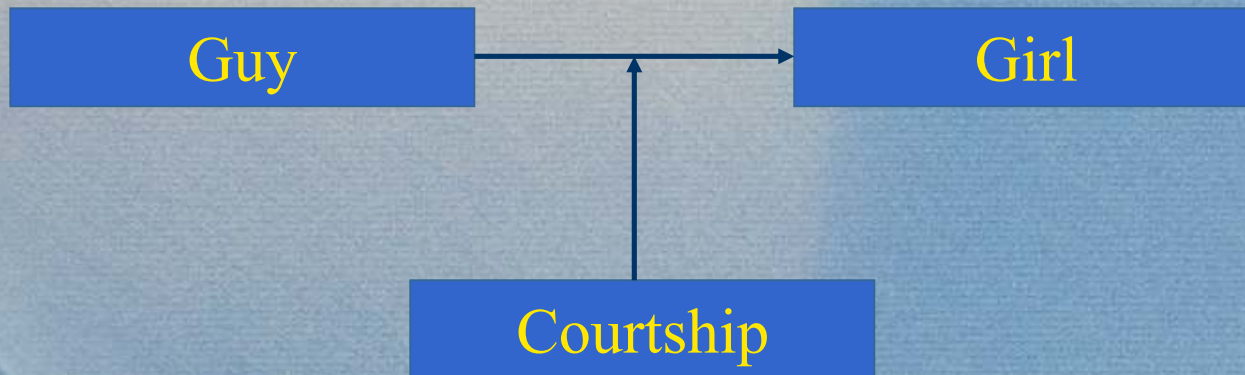
Specifically, the **select** method of a super-class is inherited by the sub-classes.

For example,

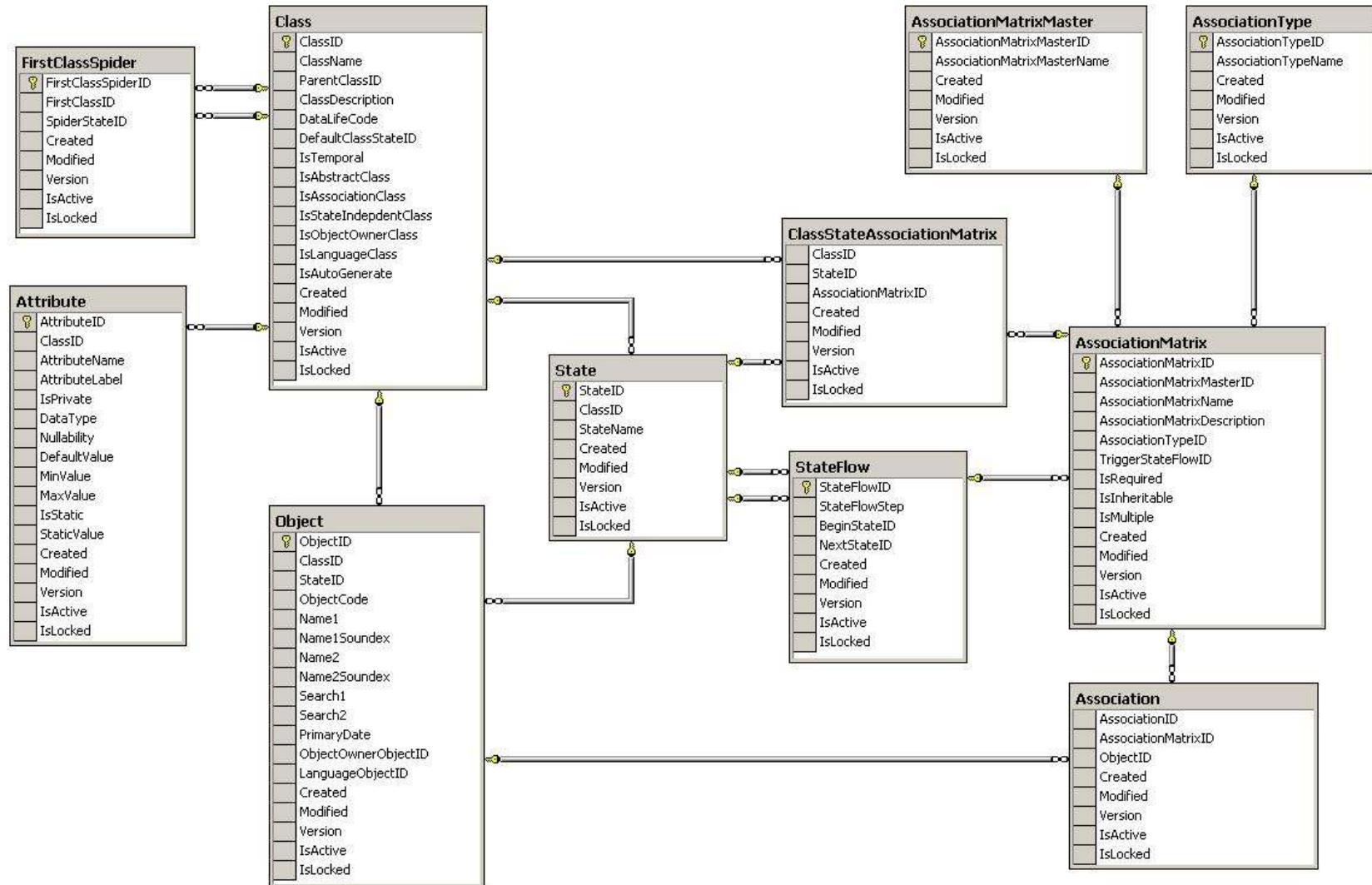
- a Human *is an* Animal
- a Guy *is a* Human

Association Classes

A Class may describe the association between two other classes.



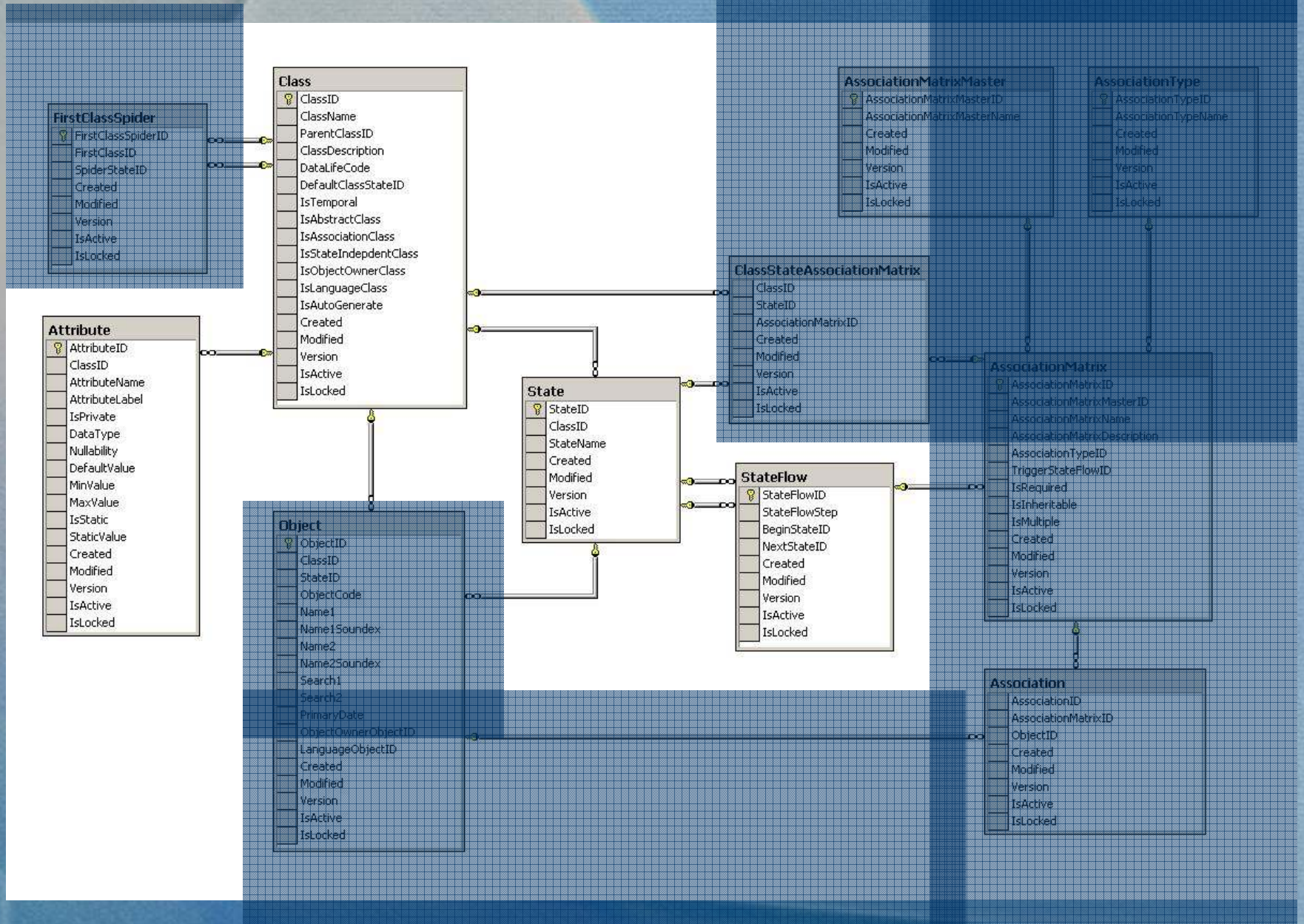
Nordic Metadata schema



Nordic: Class Management

- 37,000 Feet
- **Class Management**
- Object Management
- Association Management
- Next Steps

Class Management Schema



Class/Attribute MetaData

Classes

- Name / Description
- Temporal
- Abstract Class
- Association Class
- Language Class
- Static Attributes
- Default WorkflowState
- AutoGenerate Facade
- ObjectOwner Class
- Data Life

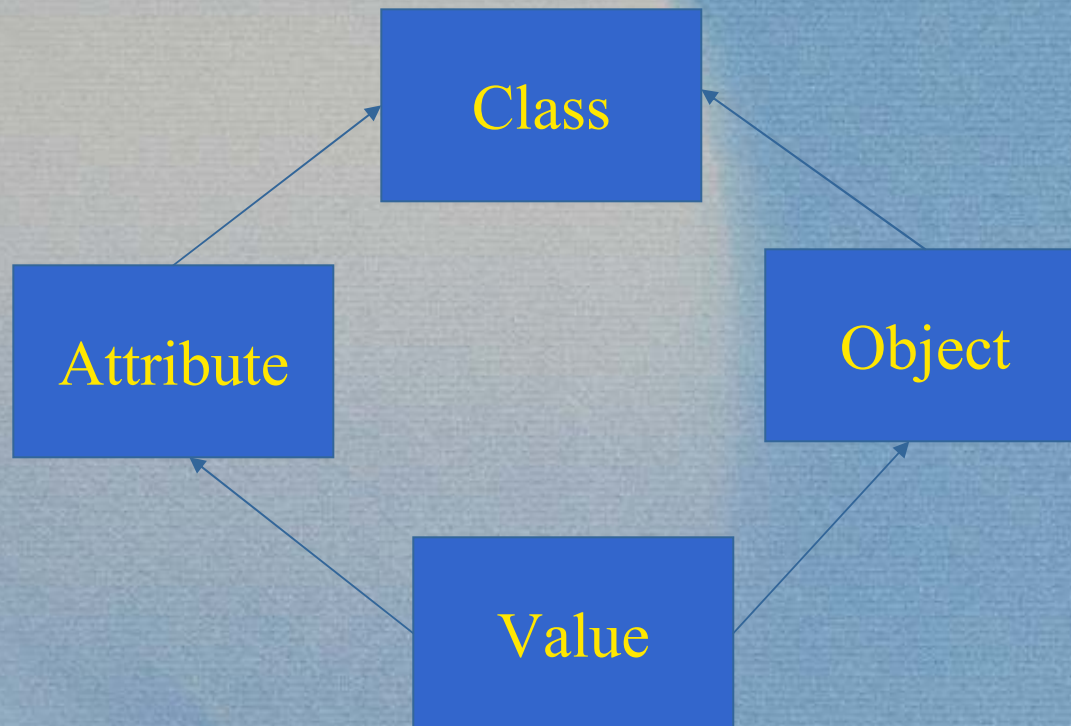
Attributes

- Name / Label
- Private
- T-SQL Data Types
- Required
- Min/Max
- Domain Integrity
- Abstract Class

Class Management Façade

- `CreateClass`
- `DropClass`
- `dbo.SuperClasses()`
- `dbo.SubClasses()`
- `CreateProperty`
- `CreateClassState`
- `DropClassState`

Option 1: Generic (Value-Pairs) Pattern



SQL Server 2000 Bible - Dynamic Relational Pattern
Con: difficult to construct data-set rows / columns
Con: normal SQL unusable

Option 2: Concrete Class Pattern

Table - Columns

ROWS

SuperClass ObjID, a, b

SubClass ObjID, a, b, c, d

SubClass ObjID, a, b, c, d, e, f

Pro: All columns for a specific class from single table
Con: Polymorphism requires faked-out Union

Option 3: Cascading Class Pattern

Table - Columns

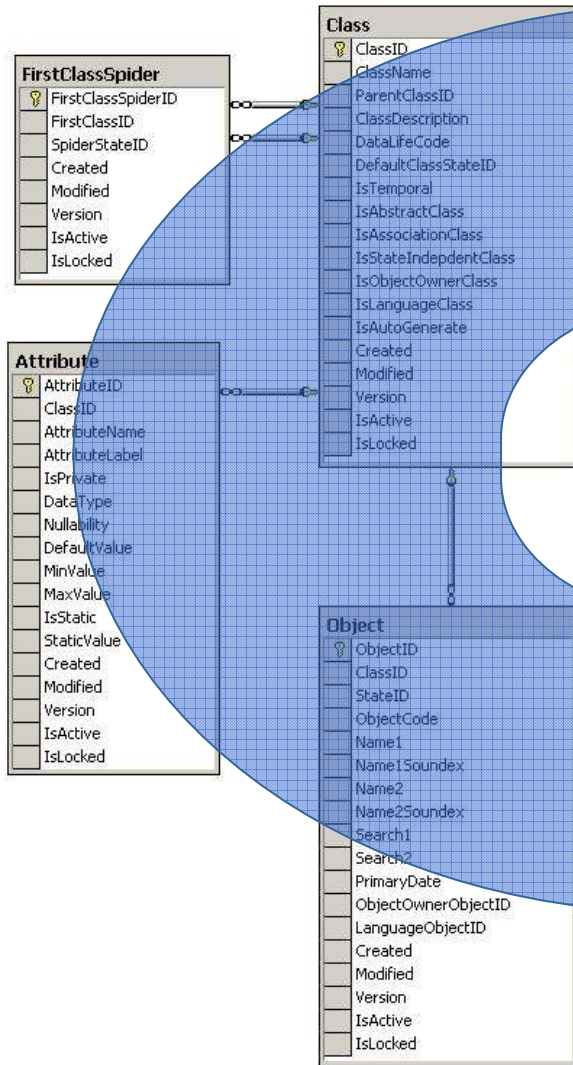
ROWS	Object ObjID, a, b	SuperClass ObjID, b,c	SubClass ObjID, e,f

Con: All columns for most classes requires join

Pro: Partitioned inserts / updates

Pro: Inherited Objects very simple

Creating Cascading Class tables



Facade: **CreateClass ()**

Cascading Custom Class tables
code generated by the facade

Custom Class tables

Custom Class tables

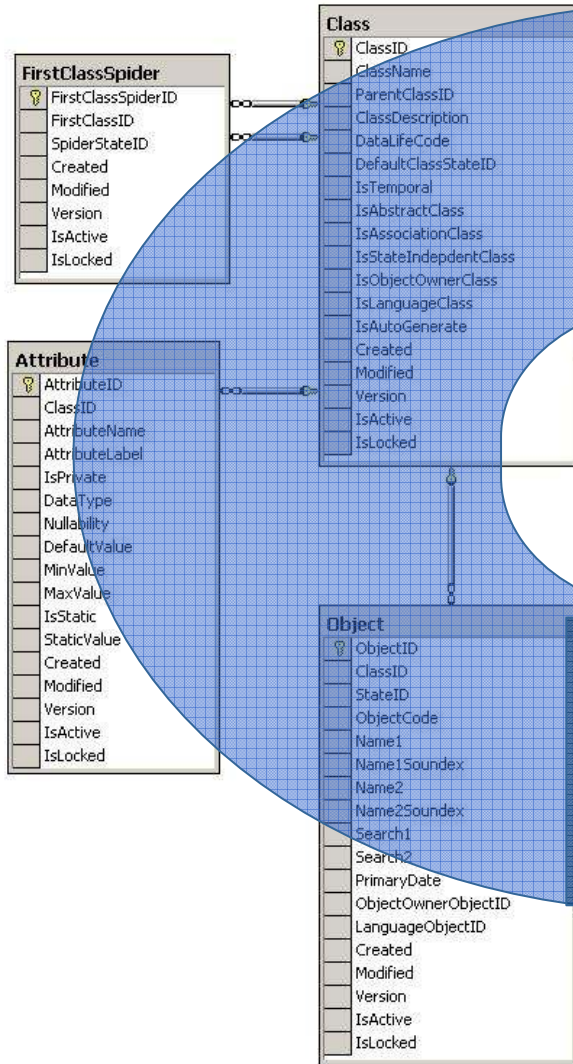
Custom Class tables

Custom Class tables

Custom Class tables

Creating Cascading Class tables

web service
vClass pClass fClass



Facade: `CreateClass()`

object access methods
code generated by the facade

Custom Class tables

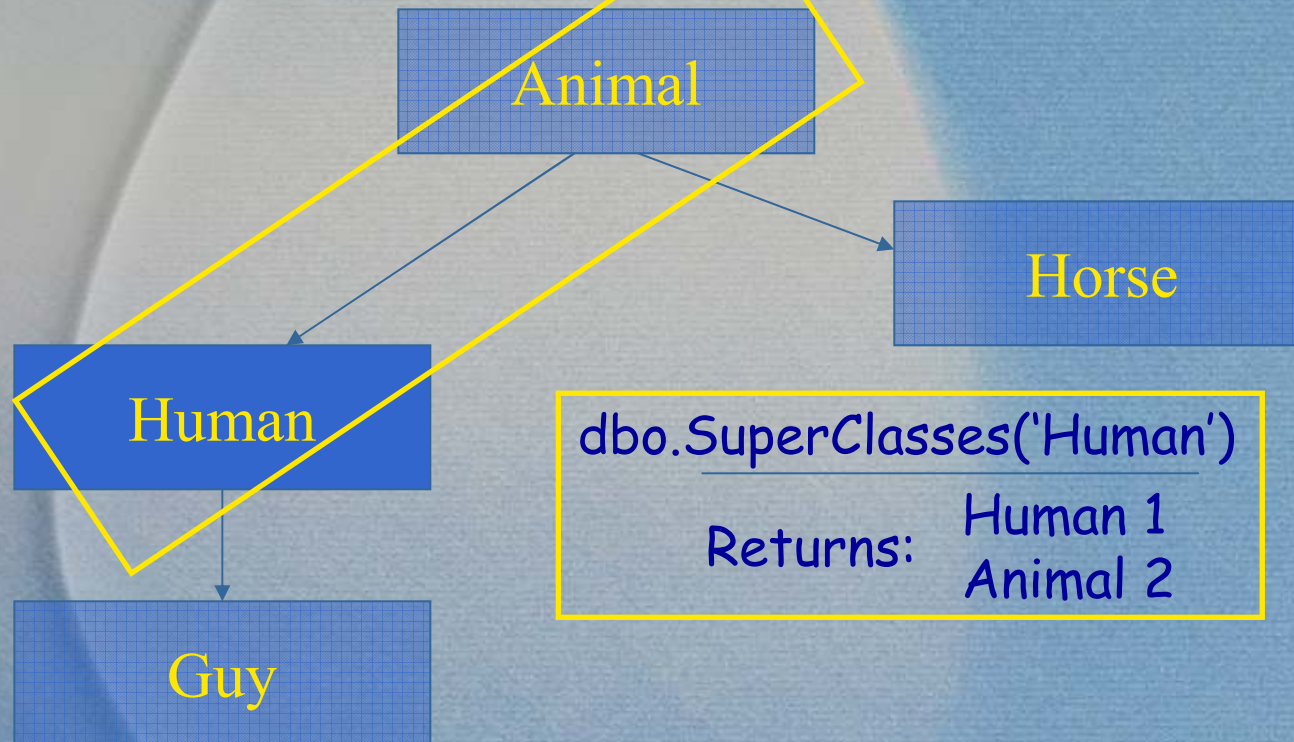
Custom Class tables

Custom Class tables

Custom Class tables

Custom Class tables

Navigate Class Hierarchy

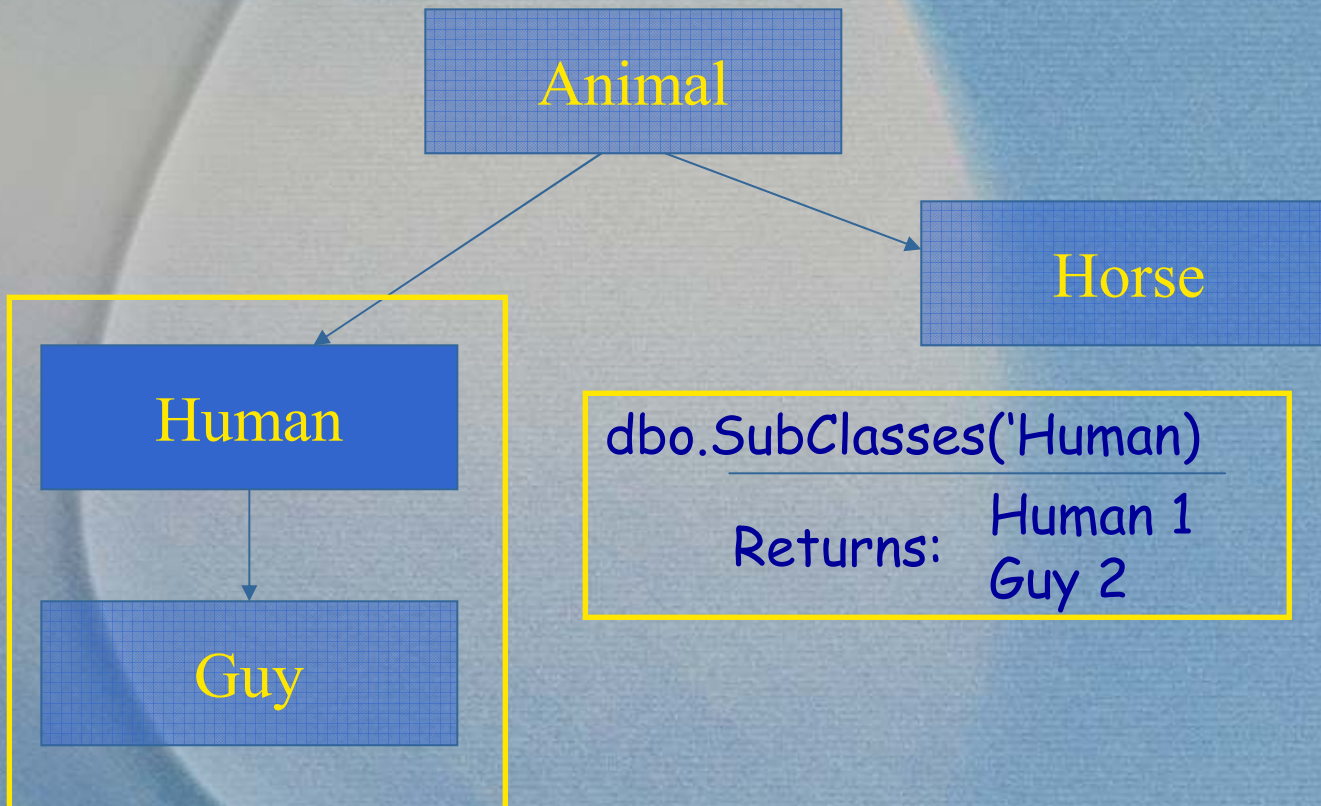


```
dbo.SuperClasses('Human')
```

```
Returns: Human 1  
Animal 2
```

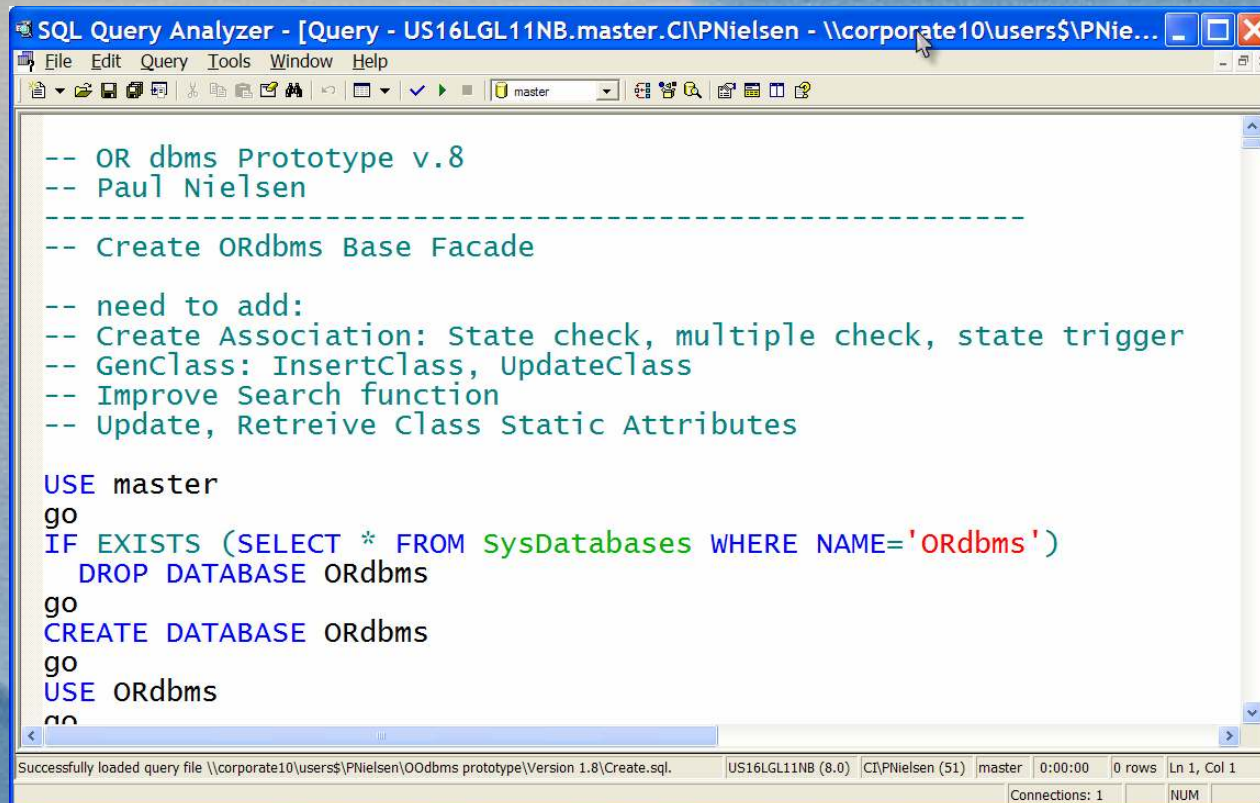
Functions return a record set so queries can apply information from all applicable classes.

Navigate Class Hierarchy



Functions return a record set so queries can apply information from all applicable classes.

Live Code #1 - Classes



The screenshot shows a window titled "SQL Query Analyzer - [Query - US16LGL11NB.master.CI\PNielsen - \\corporate10\users\$\PNie...". The window contains the following T-SQL code:

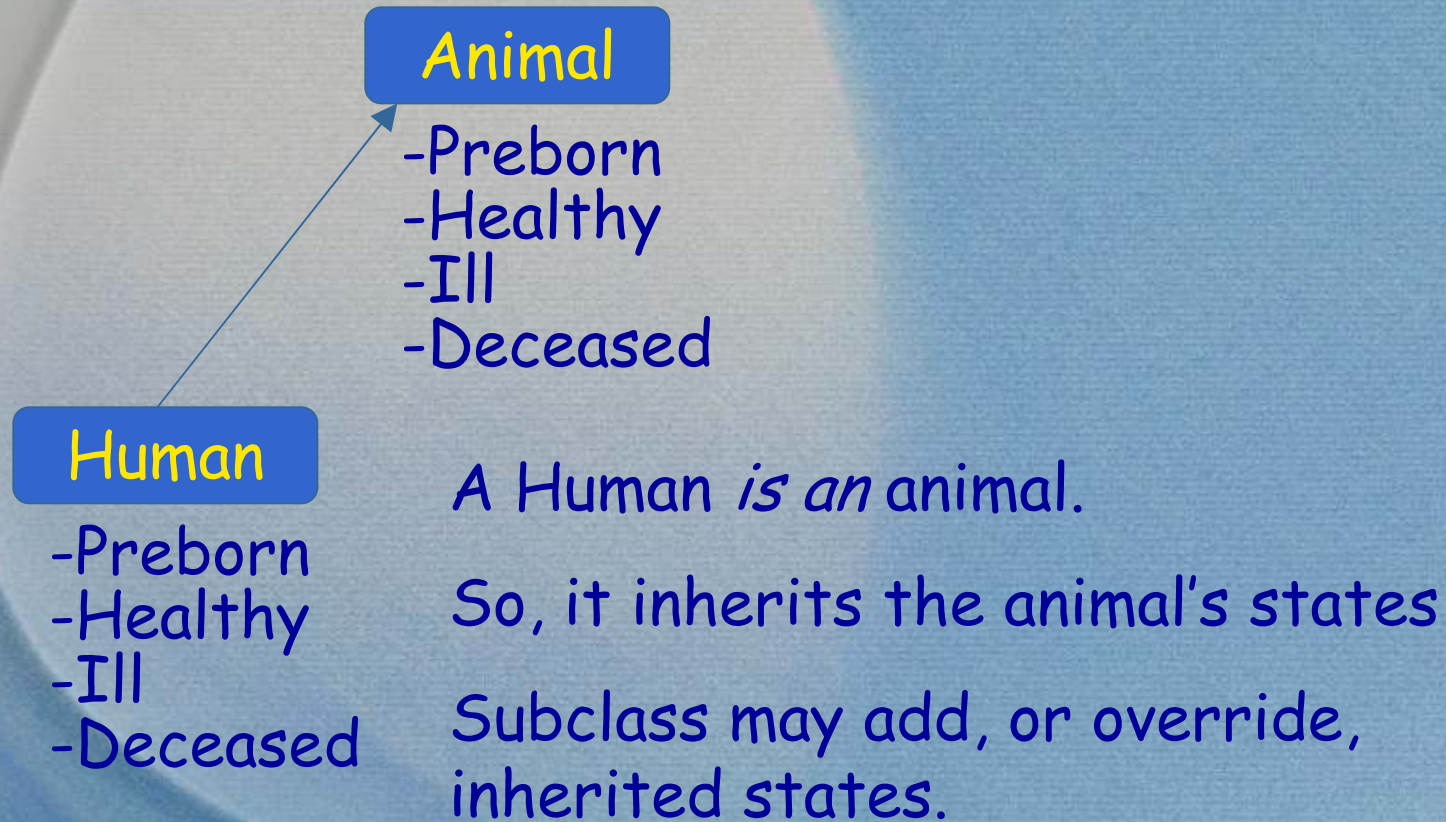
```
-- OR dbms Prototype v.8
-- Paul Nielsen
-----
-- Create ORdbms Base Facade

-- need to add:
-- Create Association: State check, multiple check, state trigger
-- GenClass: InsertClass, UpdateClass
-- Improve Search function
-- Update, Retrieve Class Static Attributes

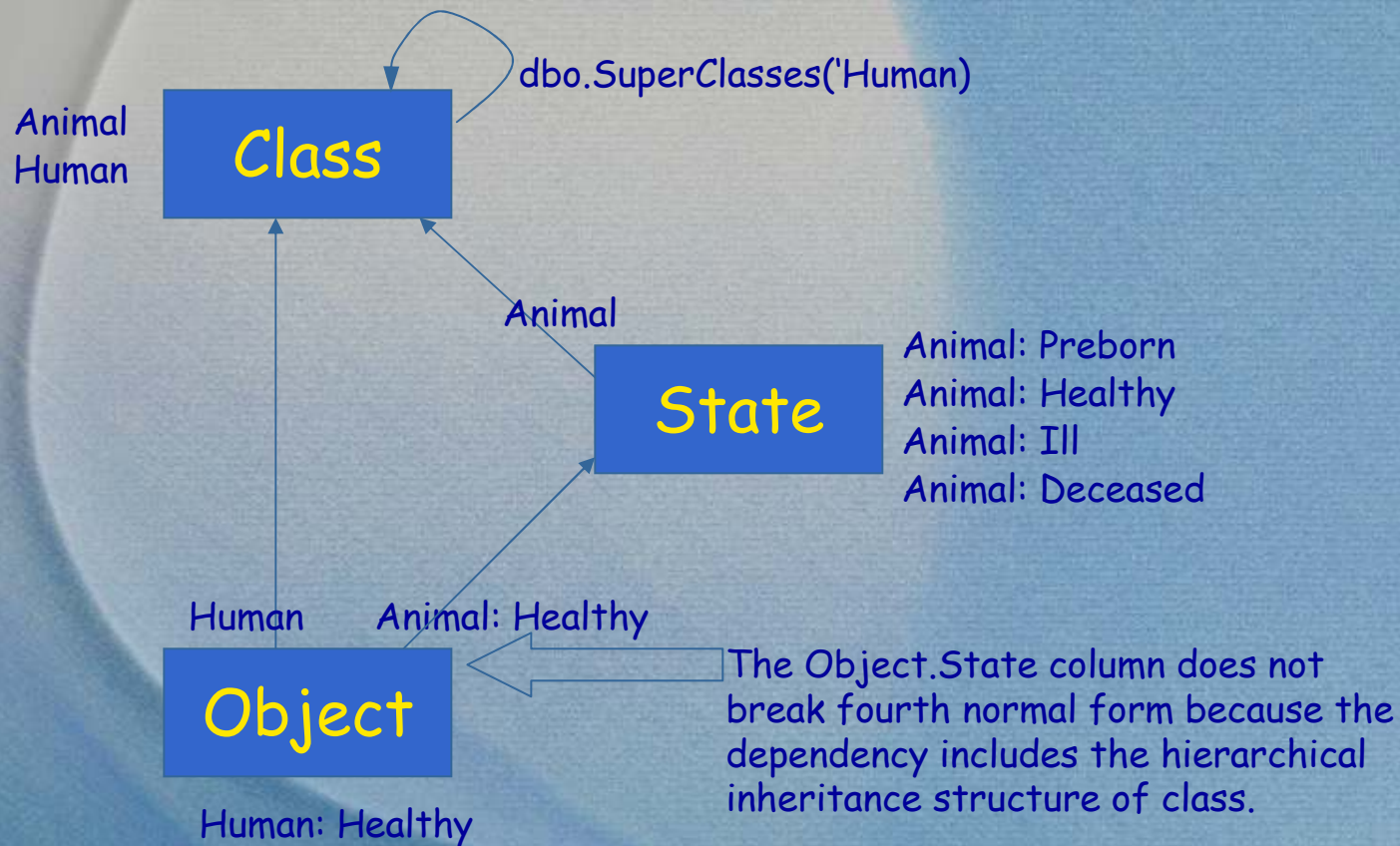
USE master
go
IF EXISTS (SELECT * FROM SysDatabases WHERE NAME='ORdbms')
    DROP DATABASE ORdbms
go
CREATE DATABASE ORdbms
go
USE ORdbms
go
```

The status bar at the bottom of the window displays: "Successfully loaded query file \\corporate10\users\$\PNielsen\Oodbms prototype\Version 1.8\Create.sql. US16LGL11NB (8.0) CI\PNielsen (51) master | 0:00:00 | 0 rows | Ln 1, Col 1 | Connections: 1 | NUM".

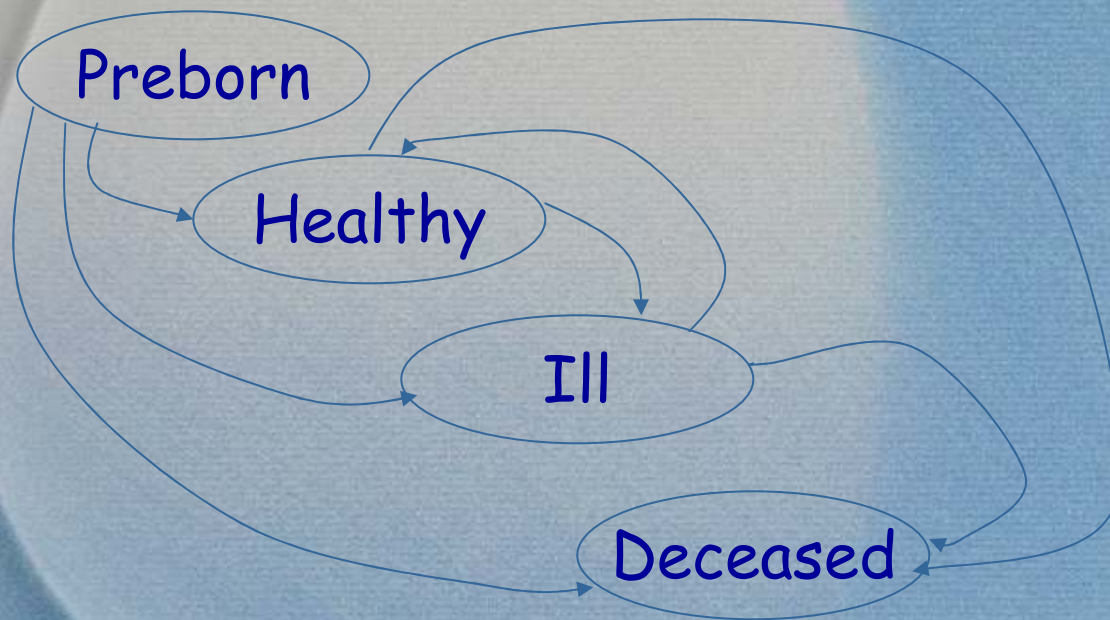
Class State Inheritance



Class State Inheritance

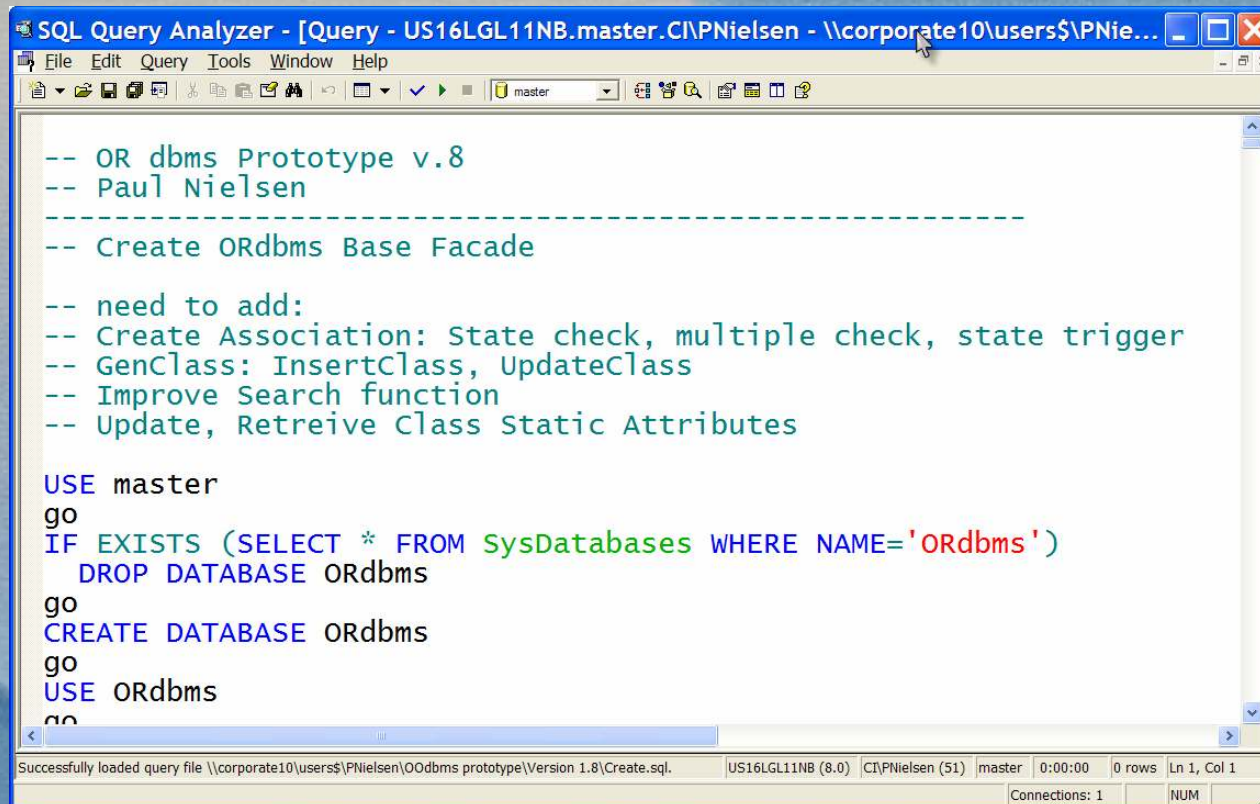


Animal Class State Flow Diagram



```
dbo.ObjectValidStates('Human', 'H101')  
ClassStateFlowSteps('Animal')
```

Live Code #2 - States



```
SQL Query Analyzer - [Query - US16LGL11NB.master.C:\PNielsen - \\corporate10\users$\PNie...
File Edit Query Tools Window Help
master
-- OR dbms Prototype v.8
-- Paul Nielsen
-----
-- Create ORdbms Base Facade

-- need to add:
-- Create Association: State check, multiple check, state trigger
-- GenClass: InsertClass, UpdateClass
-- Improve Search function
-- Update, Retrieve Class Static Attributes

USE master
go
IF EXISTS (SELECT * FROM SysDatabases WHERE NAME='ORdbms')
    DROP DATABASE ORdbms
go
CREATE DATABASE ORdbms
go
USE ORdbms
go

Successfully loaded query file \\corporate10\users$\PNielsen\Oodbms prototype\Version 1.8\Create.sql.
US16LGL11NB (8.0) C:\PNielsen (51) master 0:00:00 0 rows Ln 1, Col 1
Connections: 1 NUM
```

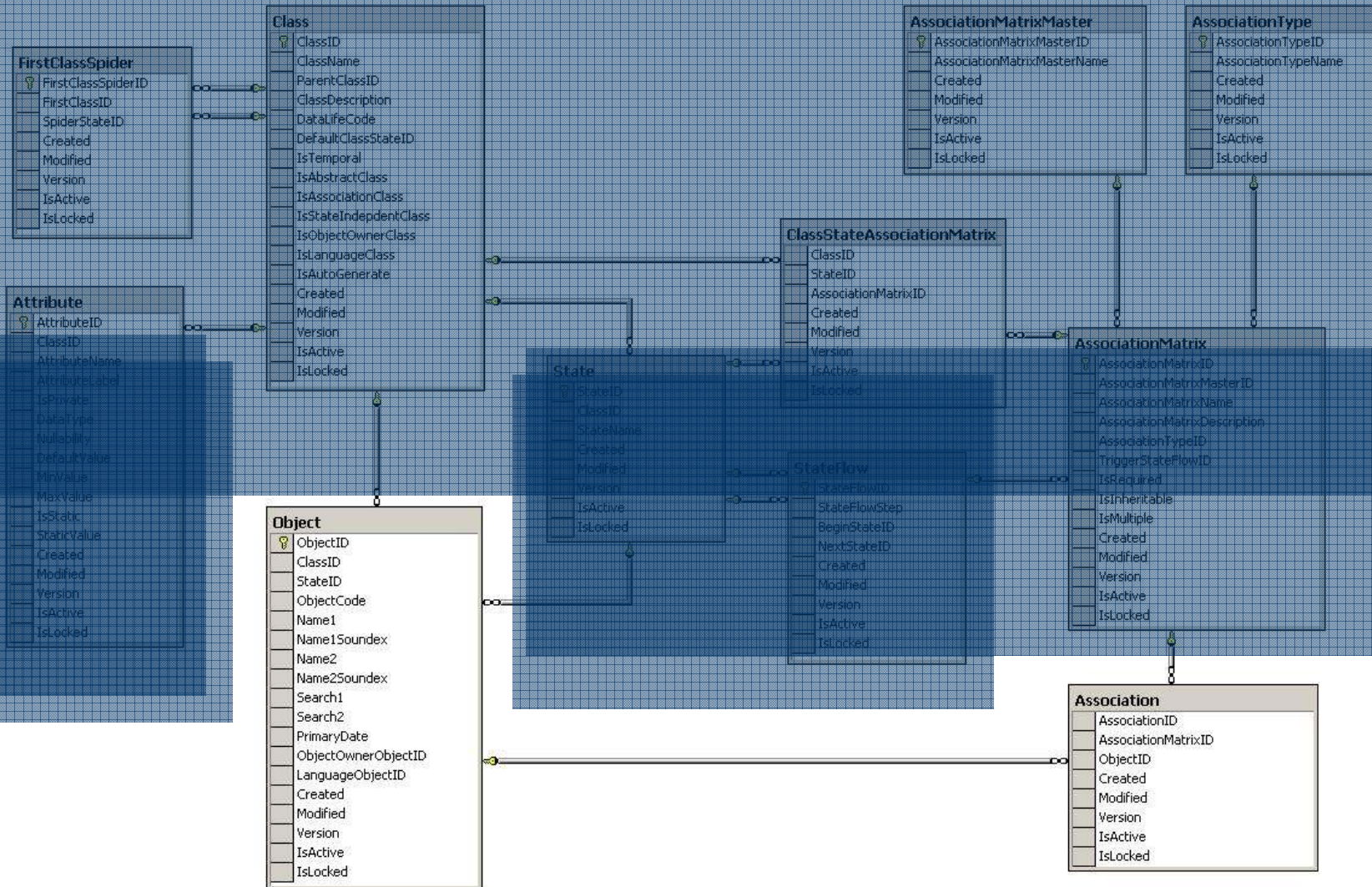
Nordic: Object Management

- 37,000 Feet
- Class Management
- **Object Management**
- Association Management
- Next Steps

Object Data

- ObjectID (GUID)
- ObjectCode
- Name1 (w/soundex)
- Name2 (w/soundex)
- Workflow State
- Primary Date
- Owner Object
- Language Object
- Version
- DateCreated
- DateModified
- Active
- Locked (Frozen)

Object Schema



Object Façade

Object are identified by ClassName & ObjectCode

- `InsertObject`
- `DeleteObject`
- `MorphObject`
- `MorphClass`
- `UpdateObject`
- `AssociateObject`
- `DeleteAssociation`

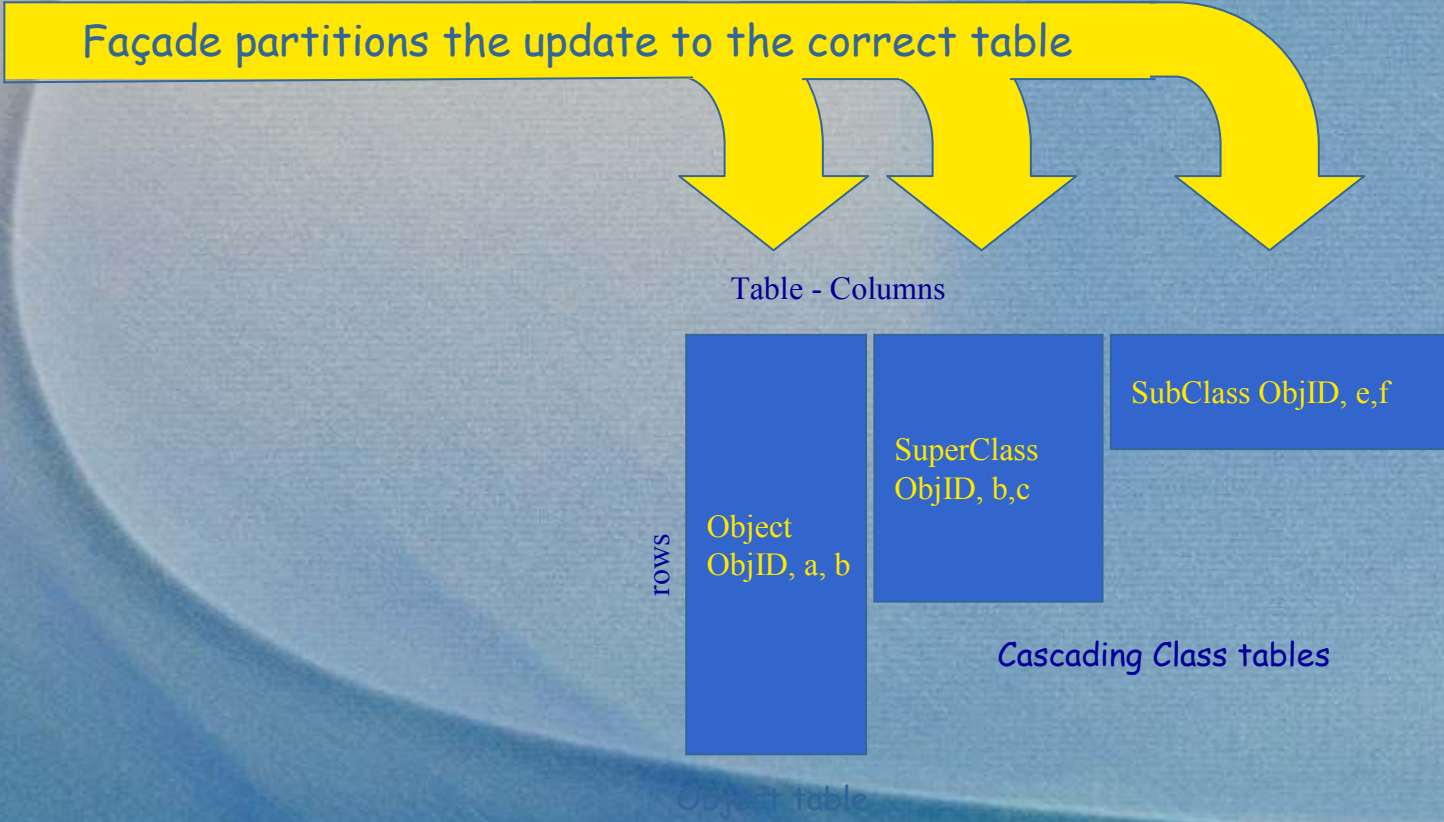
Object Façade: Searching

- `dbo.FetchObjectID(class, objectCode)`
- `ObjectSearch`

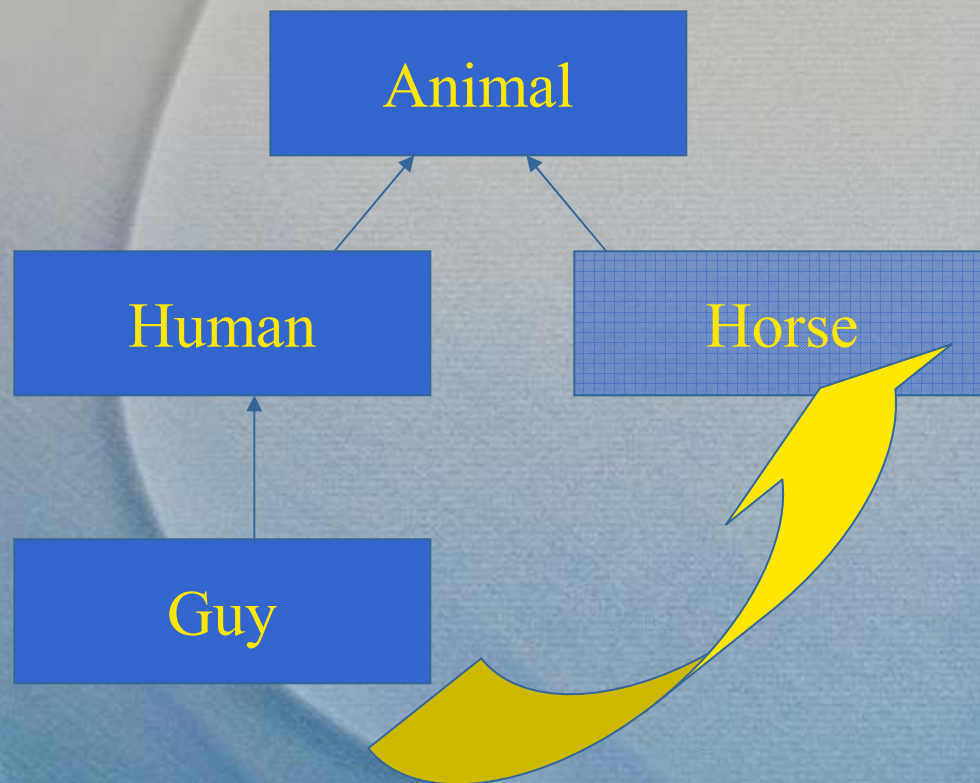
Object Façade: Cascade Class Retrieval

- Retrieval: sproc, function, & view
- Generated by Class Management Façade
- Insert object with attributes
- Update multiple attributes
- Partitions updates to correct cascaded class table or search table

Object Façade: UpdateObject



Object Façade: Object Morphing

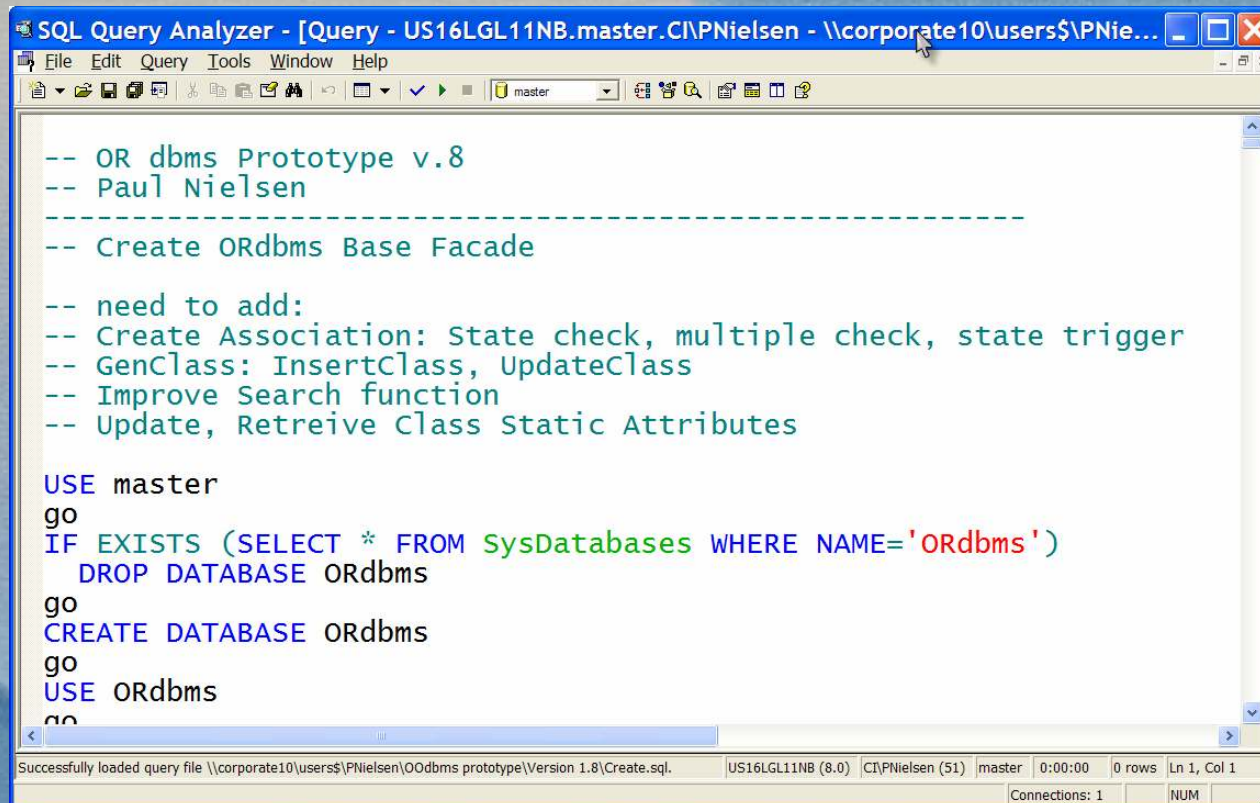


An existing object may have several cascading class rows and associations.

Morphing keeps those attributes and associations by:

- reassigning the class in the object table
- creating or removing rows in the affected cascading class tables using a set difference query.

Live Code #3 - Objects



```
SQL Query Analyzer - [Query - US16LGL11NB.master.C:\PNielsen - \\corporate10\users$\PNie...
File Edit Query Tools Window Help
master
-- OR dbms Prototype v.8
-- Paul Nielsen
-----
-- Create ORdbms Base Facade

-- need to add:
-- Create Association: State check, multiple check, state trigger
-- GenClass: InsertClass, UpdateClass
-- Improve Search function
-- Update, Retrieve Class Static Attributes

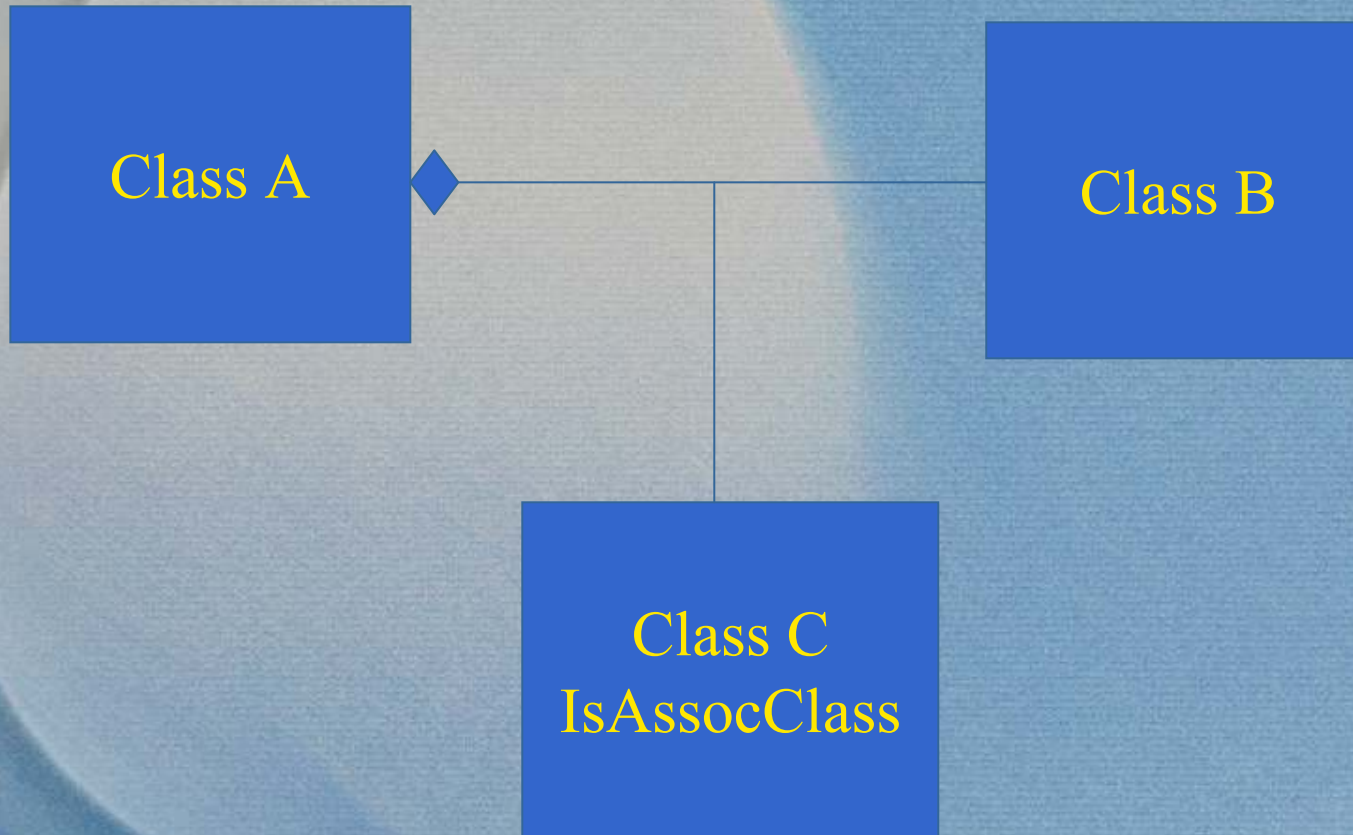
USE master
go
IF EXISTS (SELECT * FROM SysDatabases WHERE NAME='ORdbms')
    DROP DATABASE ORdbms
go
CREATE DATABASE ORdbms
go
USE ORdbms
go

Successfully loaded query file \\corporate10\users$\PNielsen\Oodbms prototype\Version 1.8\Create.sql.
US16LGL11NB (8.0) C:\PNielsen (51) master 0:00:00 0 rows Ln 1, Col 1
Connections: 1 NUM
```

Nordic: Associations

- 37,000 Feet
- Class Management
- Object Management
- **Association Management**
- Next Steps

Association Models



open: AssociationMatrixModes.xls

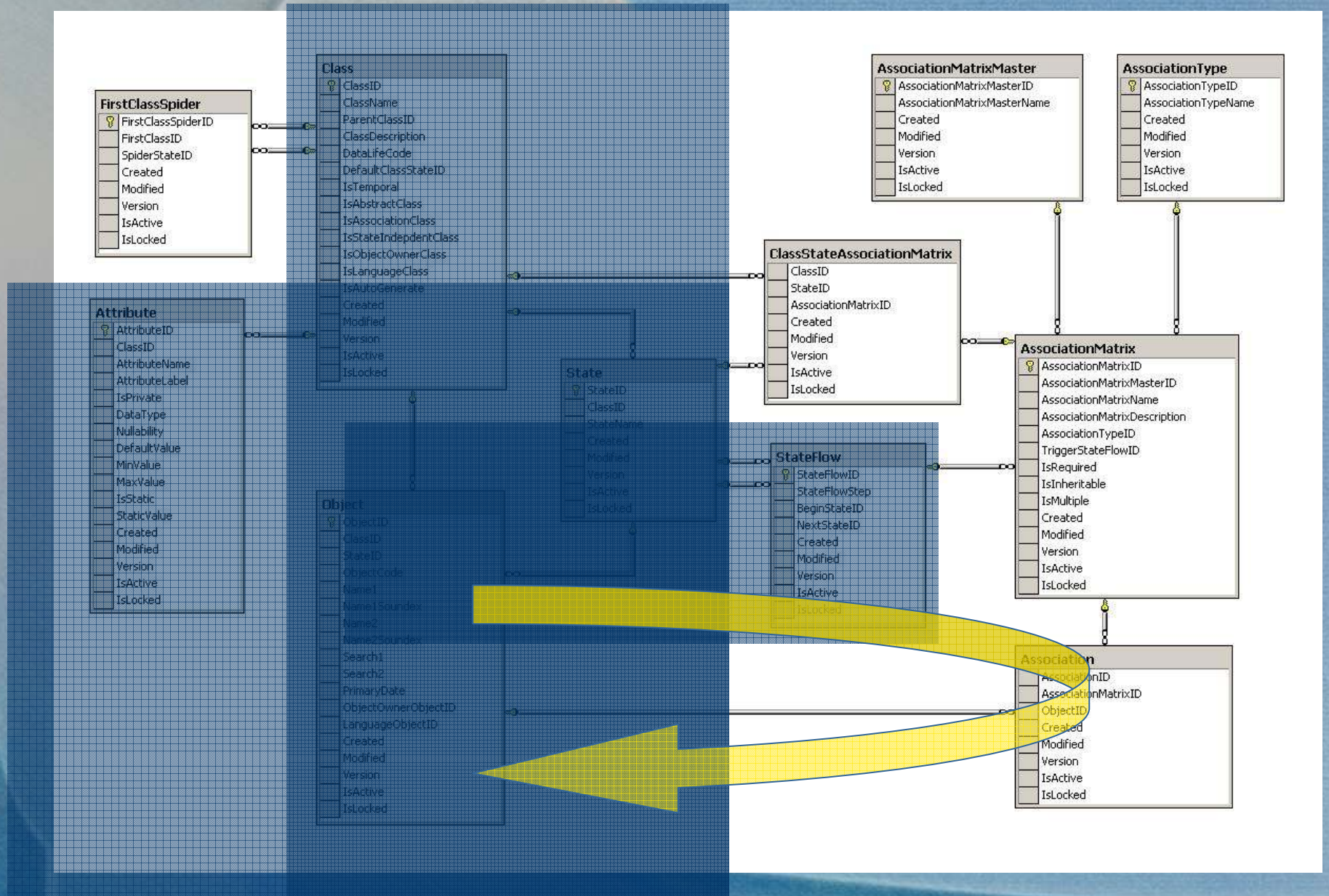
Associations MetaData

- Master Name
- Detail Name
- Detail Class/State(s)
IsRequired
- IsCollection
- IsStrong
- IsInheritable
- Version
- DateCreated
- DateModified
- Active
- Locked (Frozen)

Association Data

- ObjectID
- DateBegin
- DateEnd
- Version
- DateCreated
- Active
- Locked (Frozen)

Association Matrix Schema



Association Matrix Facade

- `CreateAssociationMatrixMaster`
- `DropAssociationMatrixMaster`
- `CreateAssociationType`
- `DropAssociationType`
- `CreateAssociationMatrix`
- `DropAssociationMatrix`
- `CreateClassStateAssociationMatrix`
- `DropClassStateAssociationMatrix`

Association Matrix

Association is Master/Detail model

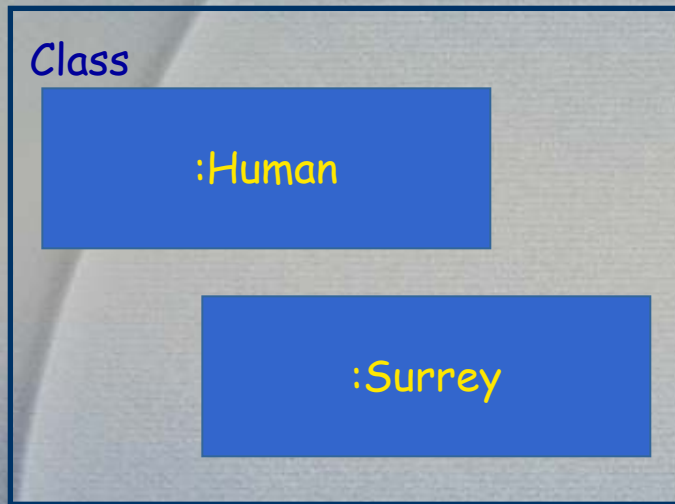
- More flexible than binary association pattern
- Tertiary+ associations
- Association Classes

Rules for valid associations based on

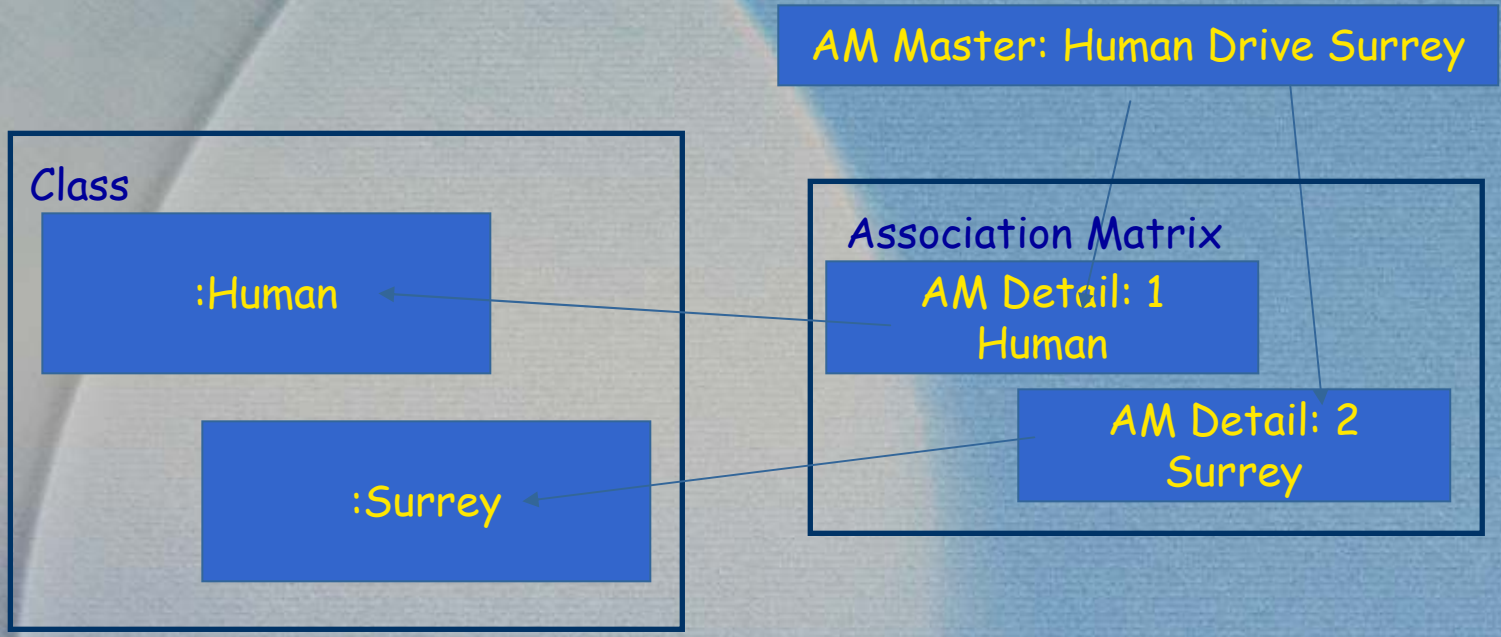
- Class / state
- Inheritance
- Collections
- Required

Association can trigger an object's class/state flow step

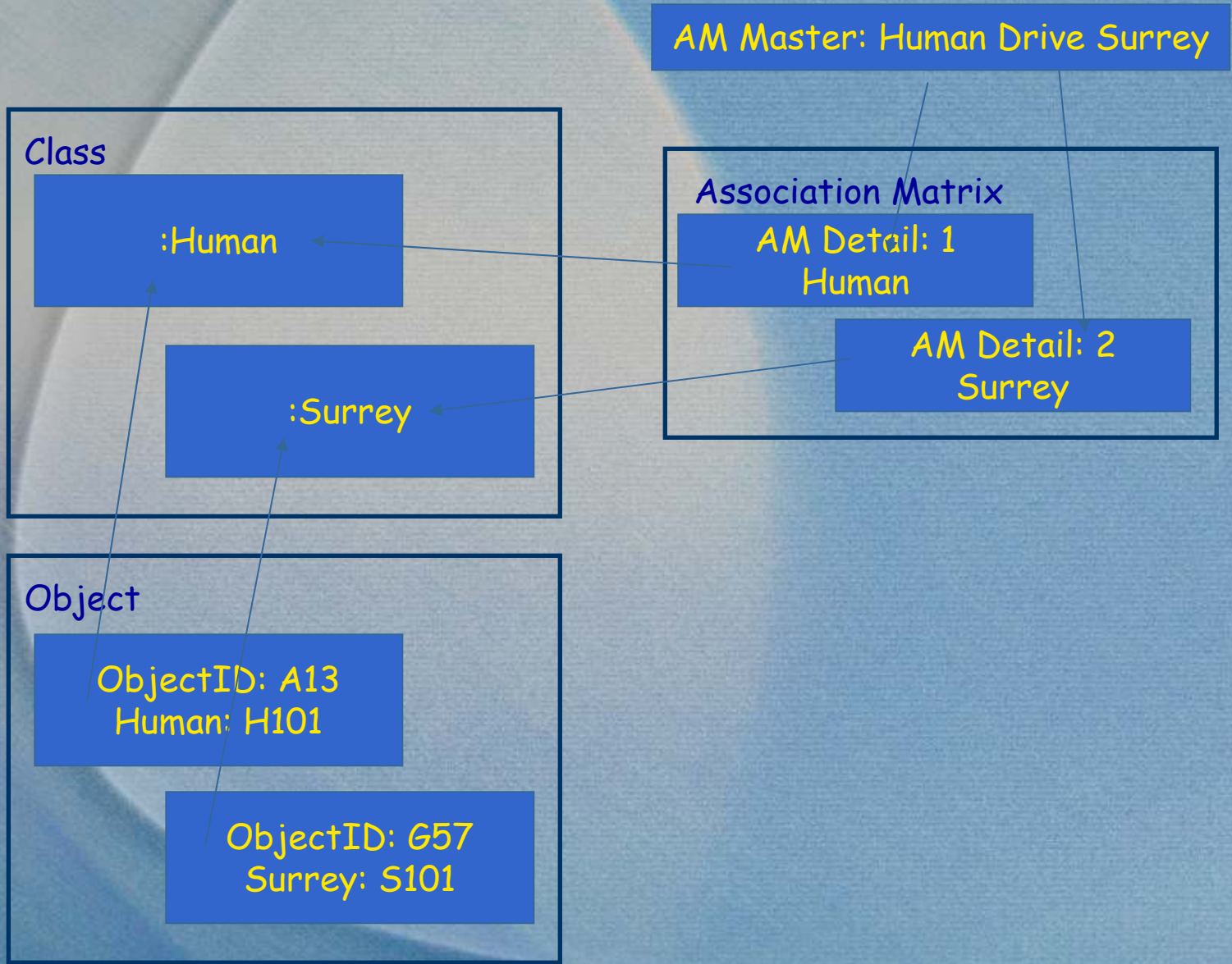
Human Drives Surrey Example Association



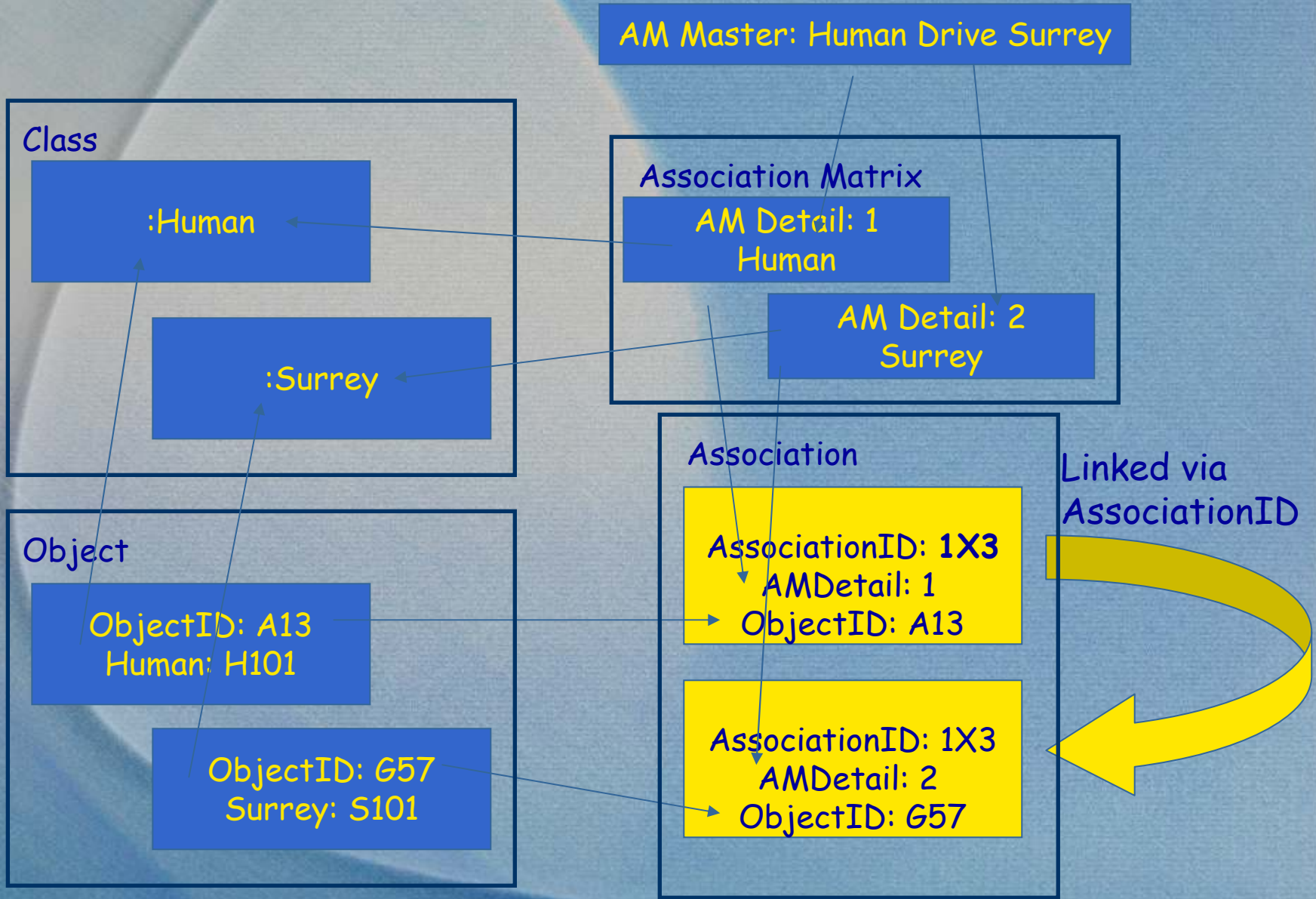
Human Drives Surrey Example Association



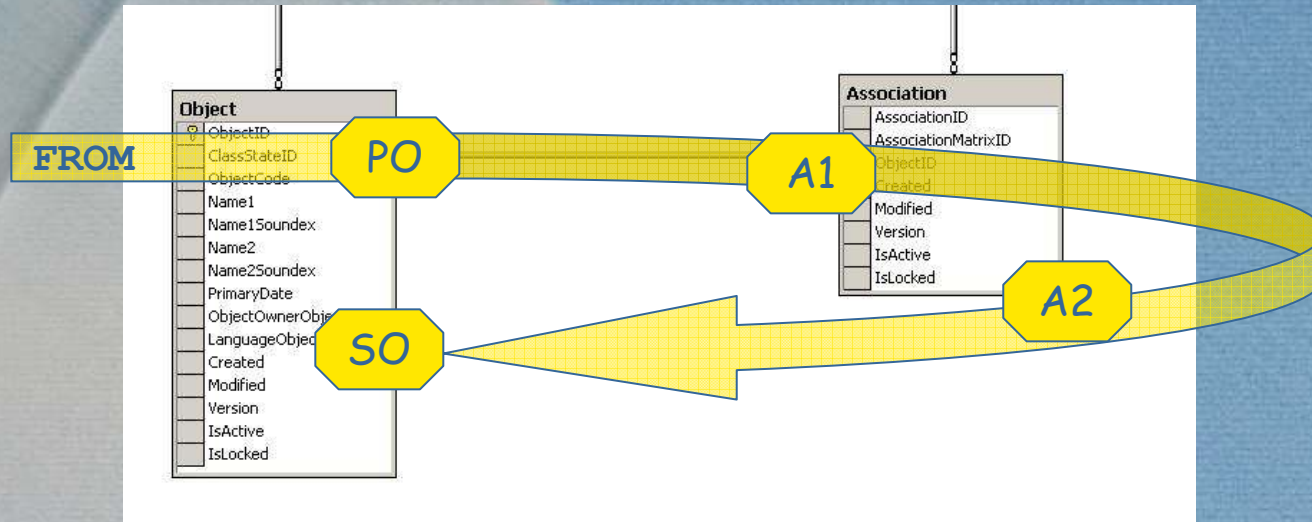
Human Drives Surrey Example Association



Human Drives Surrey Example Association



Association Query



```
SELECT PO.ObjectCode, SO.ObjectCode
```

```
FROM Object PO
```

```
JOIN Association A1
```

```
ON PO.ObjectID = A1.ObjectID
```

```
JOIN Association A2
```

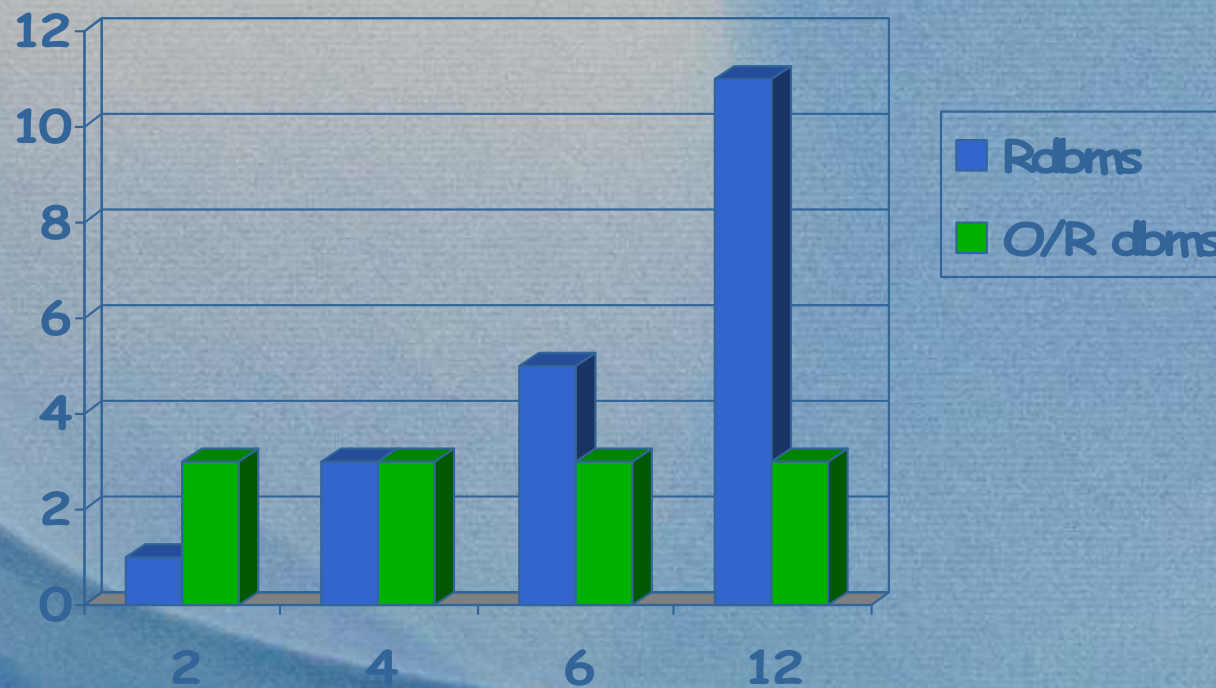
```
ON A1.AssociationID = A2.AssociationID
```

```
JOIN Object SO
```

```
ON SO.ObjectID = A2.ObjectID
```

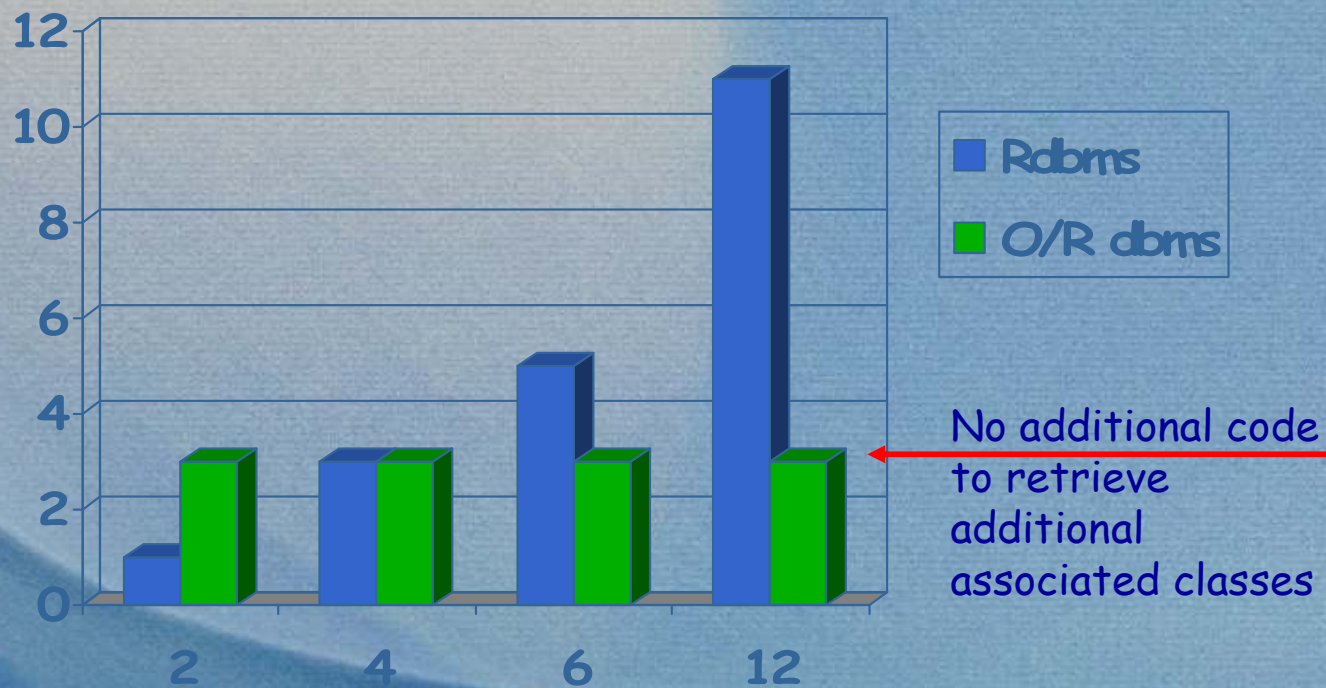
Association List scalable in complex interactions

Number of joins to retrieve an
object plus associated objects

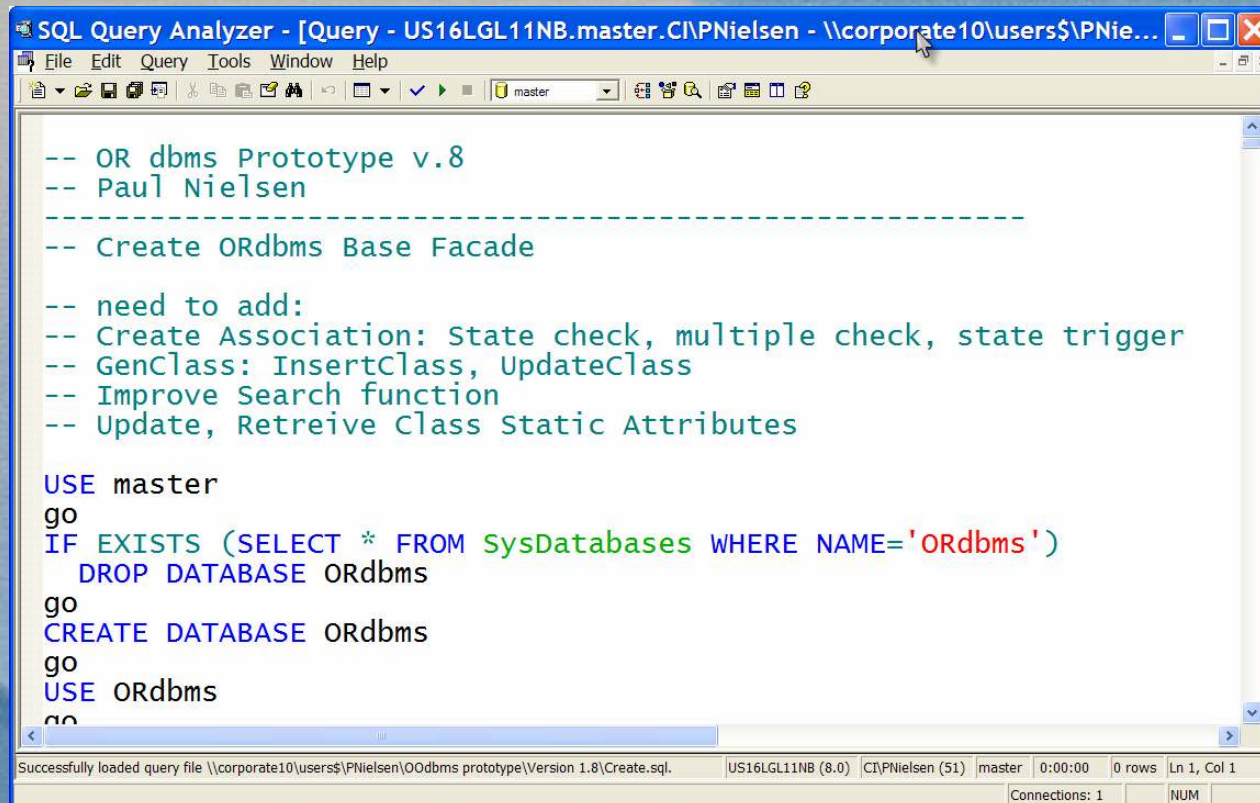


Association List scalable in complex interactions

Number of joins to retrieve an
object plus associated objects



Live Code #4 - Associations



```
SQL Query Analyzer - [Query - US16LGL11NB.master.C:\PNielsen - \\corporate10\users$\PNie...
File Edit Query Tools Window Help
master
-- OR dbms Prototype v.8
-- Paul Nielsen
-----
-- Create ORdbms Base Facade

-- need to add:
-- Create Association: State check, multiple check, state trigger
-- GenClass: InsertClass, UpdateClass
-- Improve Search function
-- Update, Retrieve Class Static Attributes

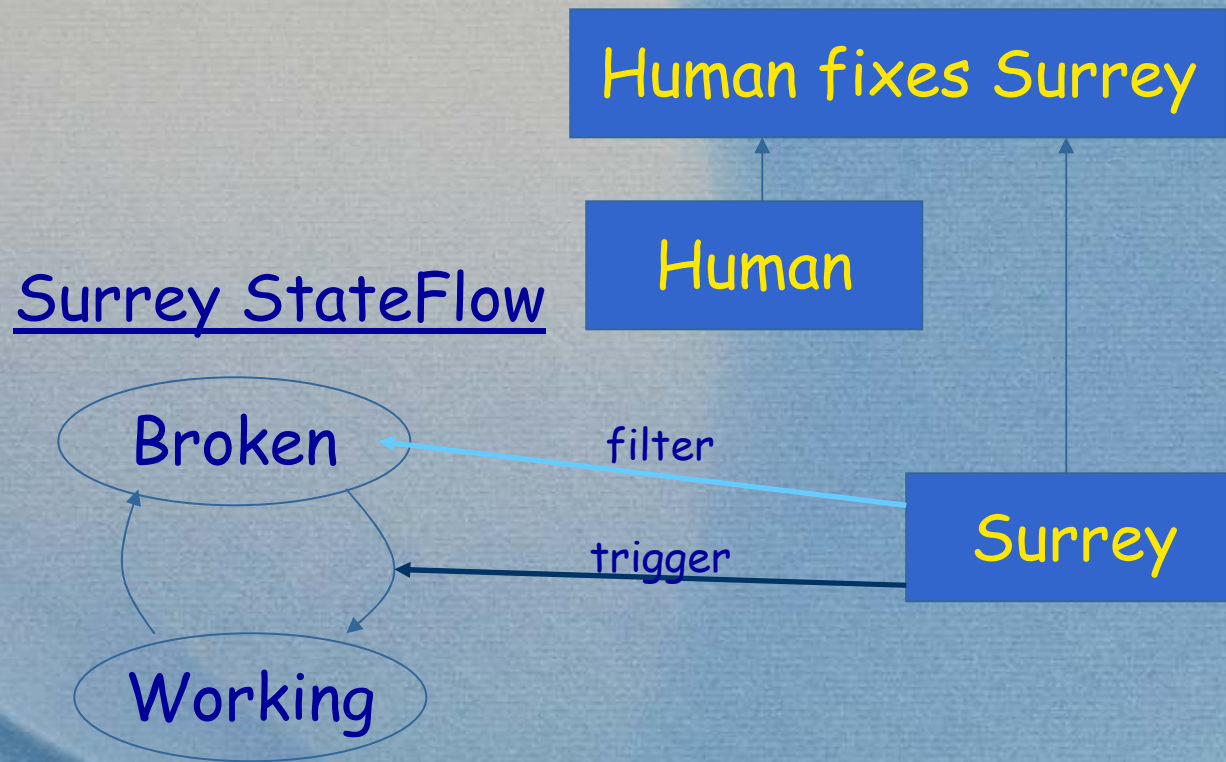
USE master
go
IF EXISTS (SELECT * FROM SysDatabases WHERE NAME='ORdbms')
    DROP DATABASE ORdbms
go
CREATE DATABASE ORdbms
go
USE ORdbms
go

Successfully loaded query file \\corporate10\users$\PNielsen\Oodbms prototype\Version 1.8\Create.sql.
US16LGL11NB (8.0) C:\PNielsen (51) master 0:00:00 0 rows Ln 1, Col 1
Connections: 1 NUM
```

Associations vs. Domain Integrity

- Model...
 - Entities
 - Associations (entity to entity)
 - Domain Integrity (attribute lookups)

Associations Matrix & StateFlow

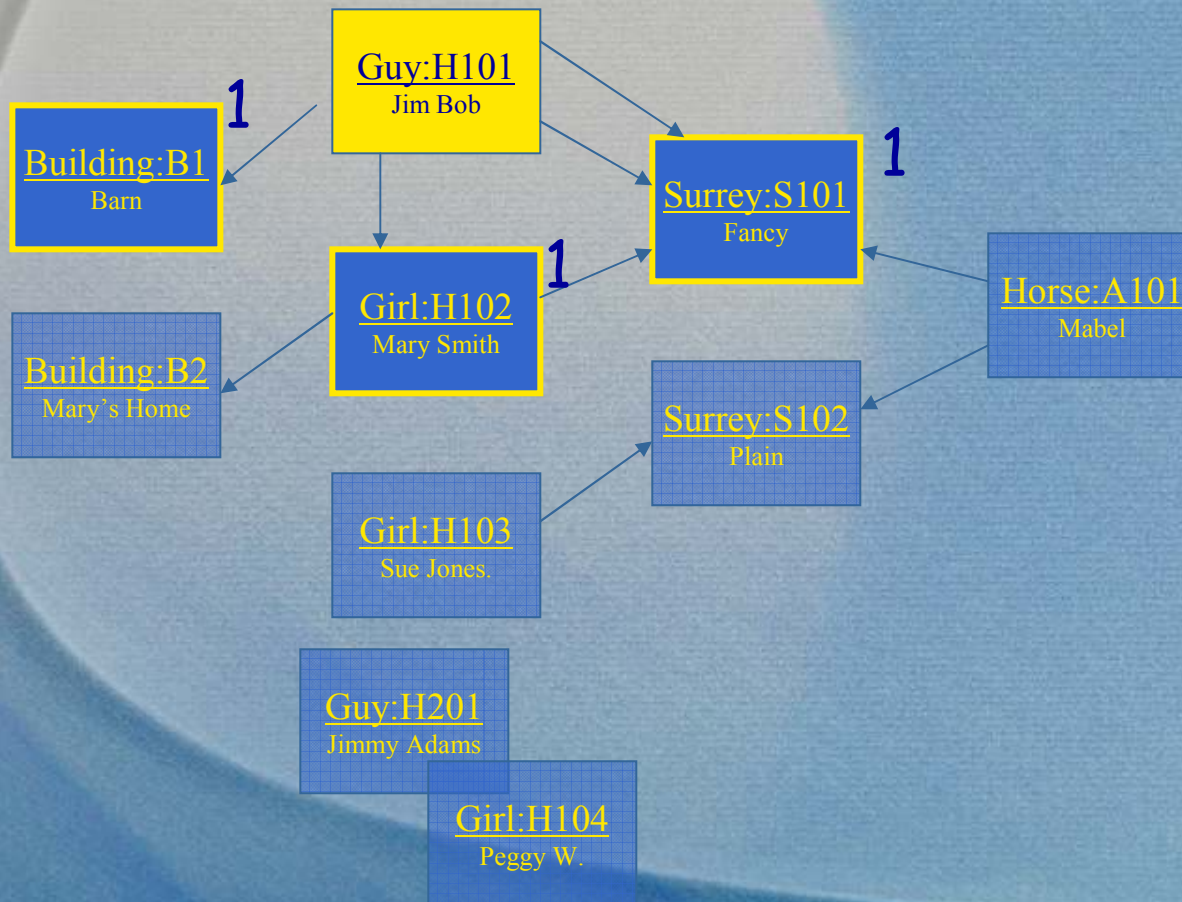


Associations Retrieval Façade

- `ObjectAssociations (Class,ObjCode)`
- `ObjectFreeAssociations (Class,ObjCode)`
- `ClassAssociatons (Class1, Class2)`
- `ObjectNotAssociated (class,AMM)`

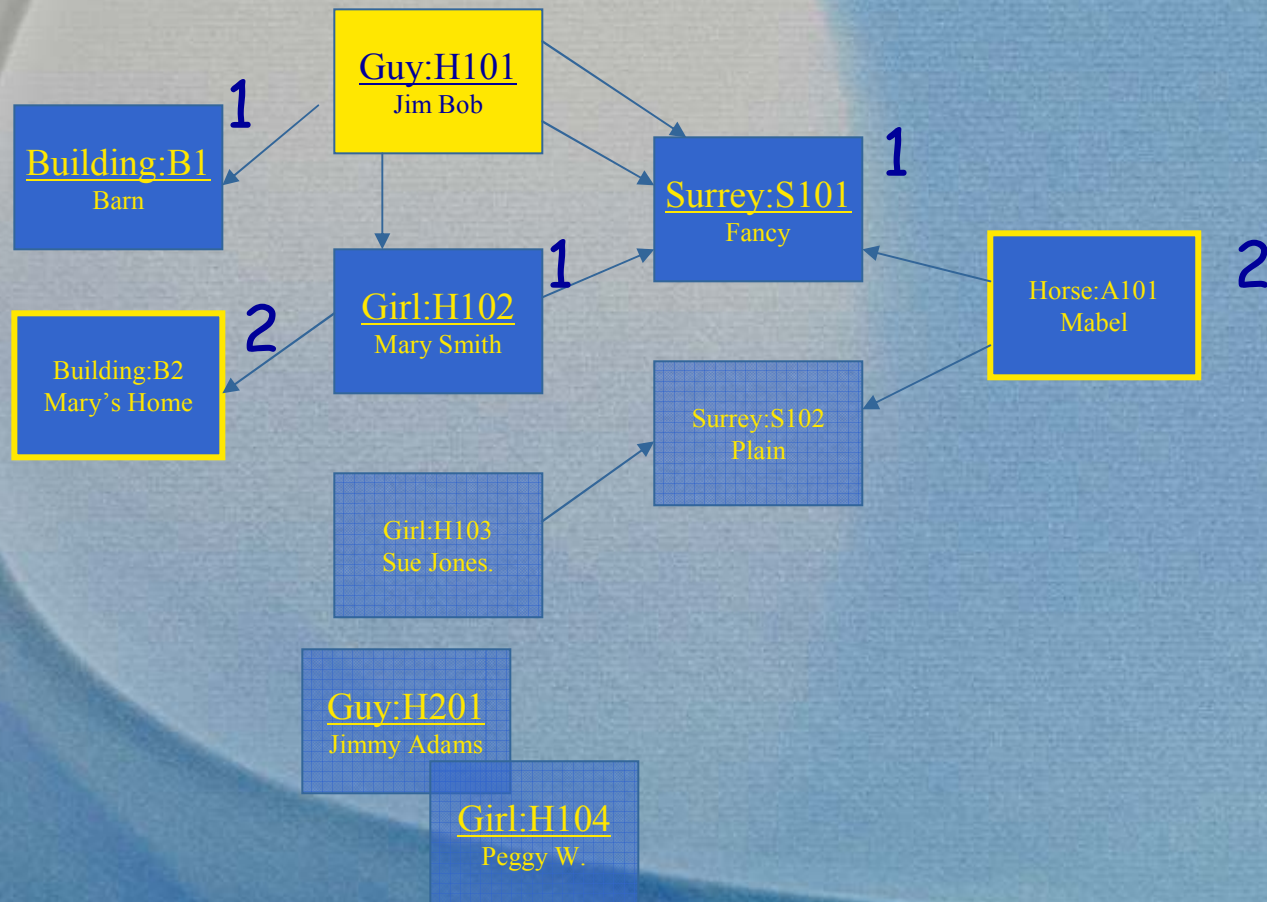
Association Mapping – 1 level

Similar to spidering all web pages in a web site



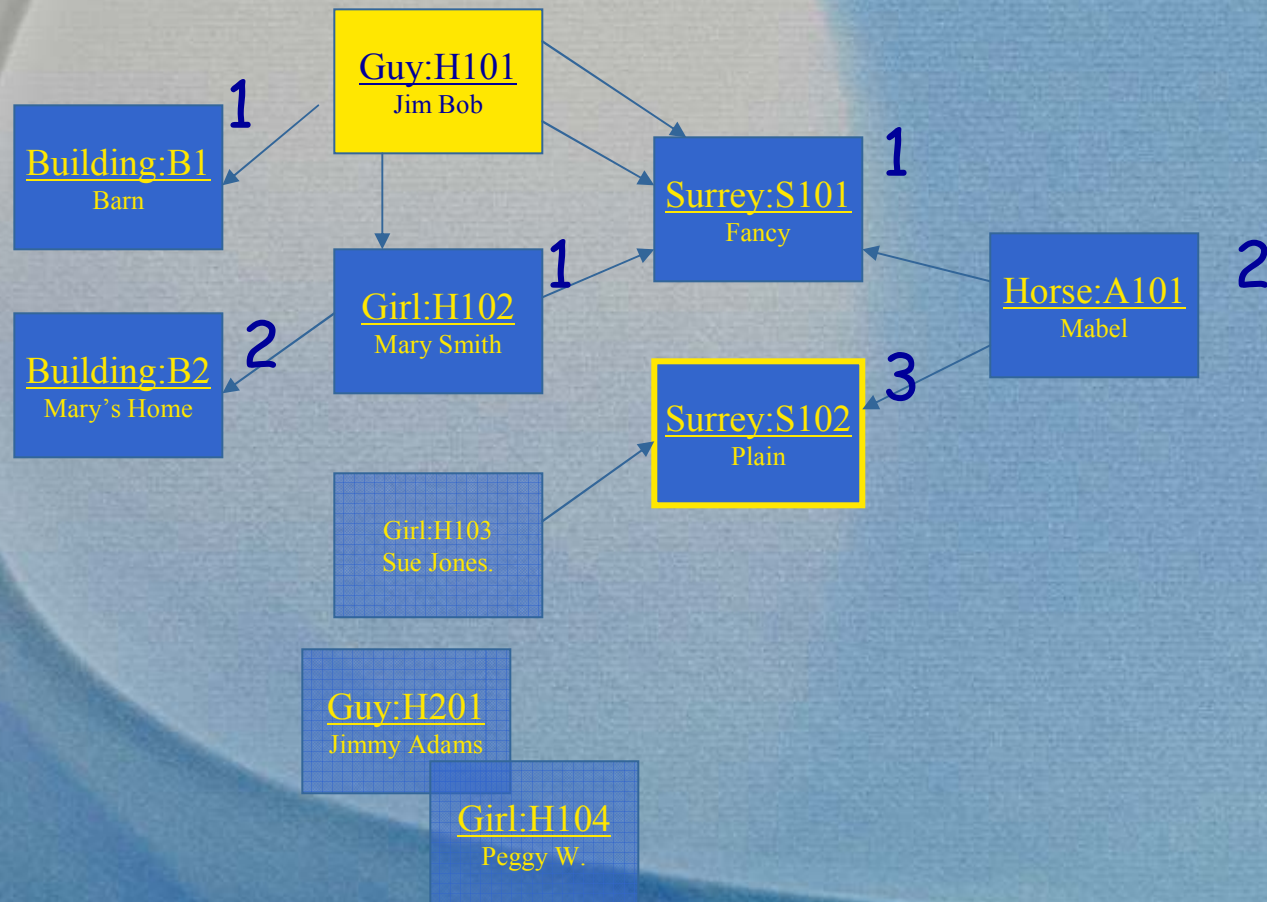
Association Mapping – 2 levels

Similar to spidering all web pages in a web site



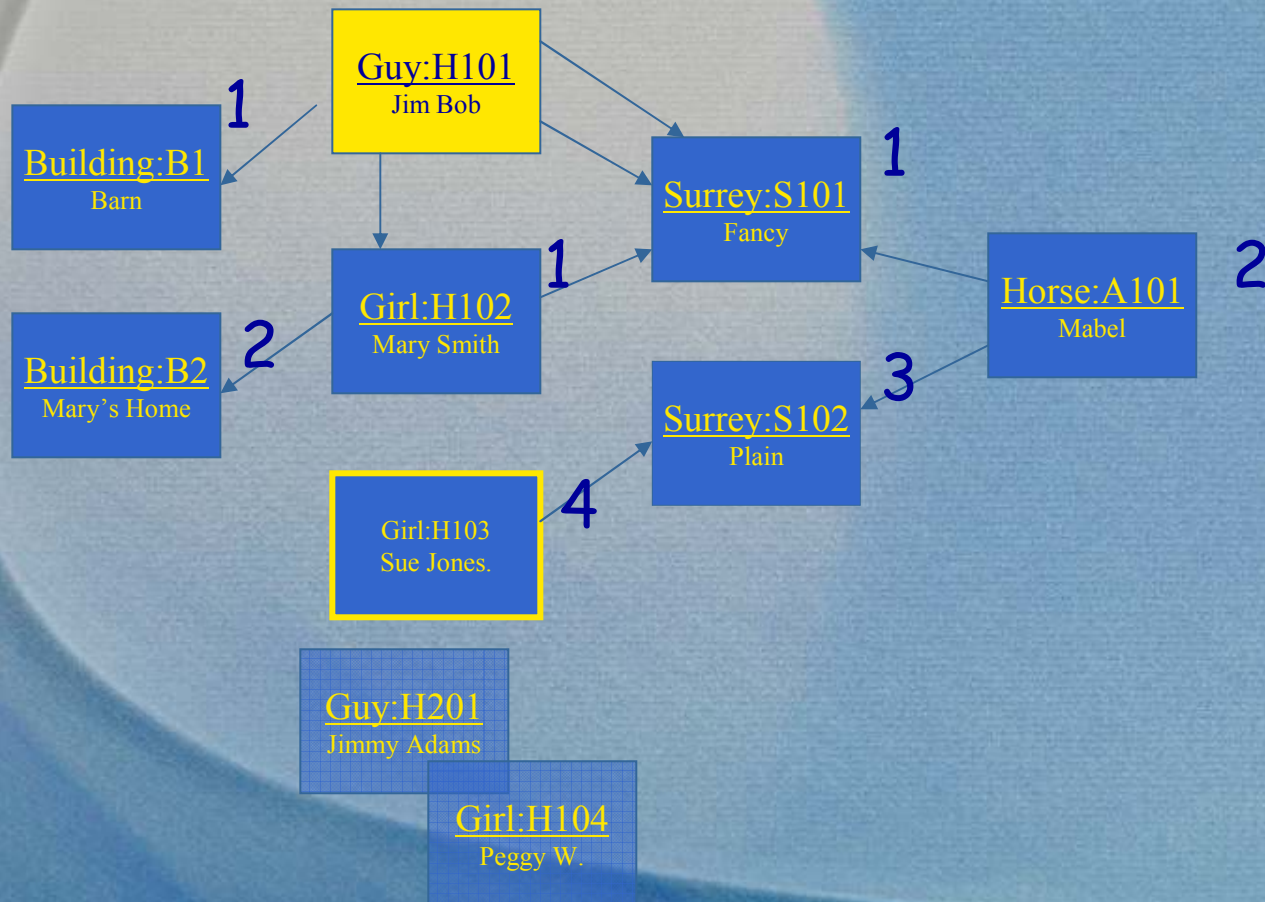
Association Mapping – 3 levels

Similar to spidering all web pages in a web site



Association Mapping – 4 levels

Similar to spidering all web pages in a web site



Controlling Association Mapping

- **Depth of Levels scanned**
- **Limiting Class Backtracking**
 - (once one object of a class is found don't find any more objects of that class)
 - (optionally configured by class)
- **First Class Objects**
 - (when an object of this class is mapped always only find these associated class objects)

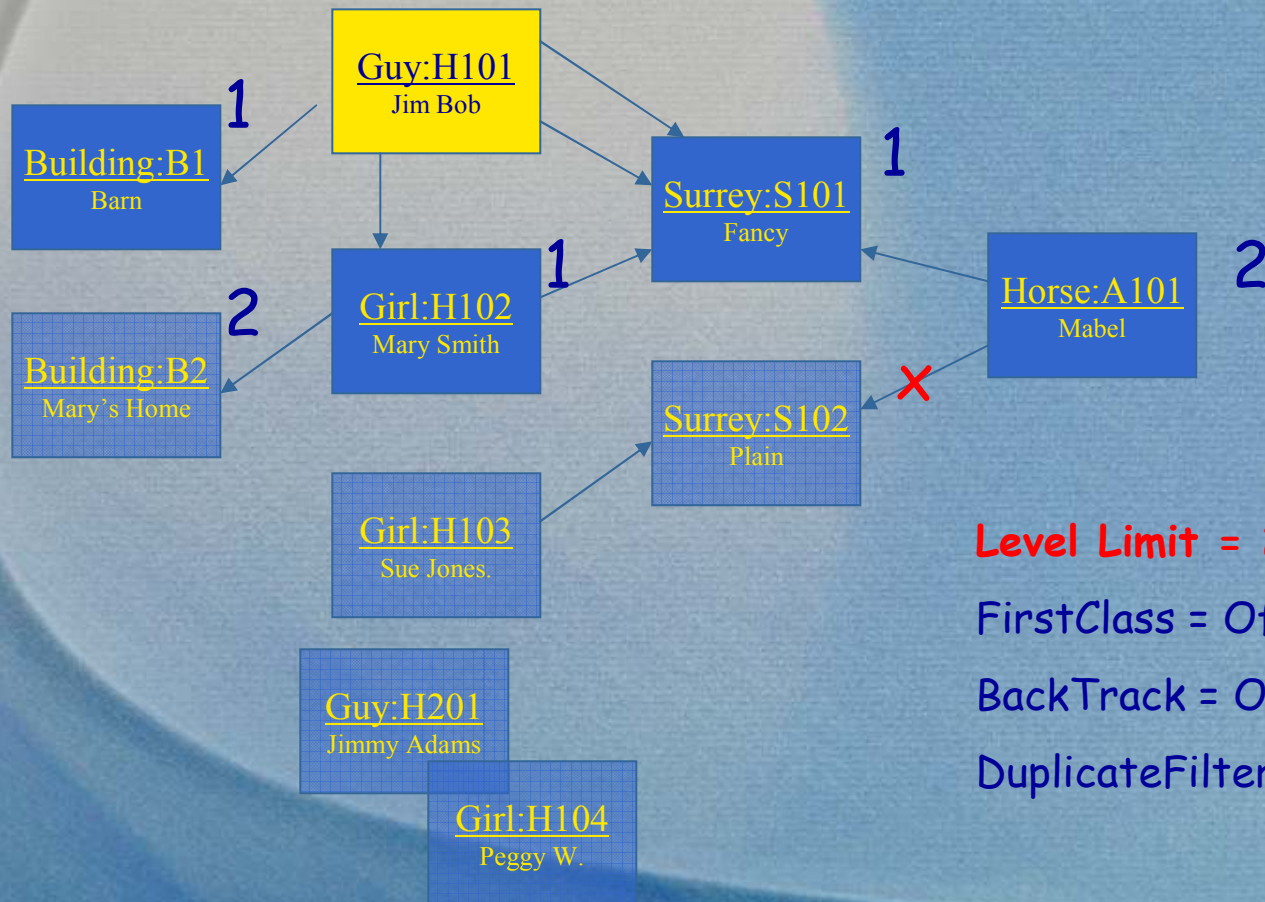
Association Mapping

ObjectFreeAssociations(level, 1st class, backtrack, duplicates)

- Level Limit (0 = infinite)
- First Class Filter (1 = on, 0 = off)
- BackTrack Filter (
 - 0 = off,
 - 1 = Filter all class backtrack,
 - 2 = ClassConfiguredFilter)
- DuplicateObject Filter (1 = on, 0= off)

Association Mapping – Level Limit

ObjectFreeAssociations('Guy','H101',2,0,0,0)



Level Limit = 2

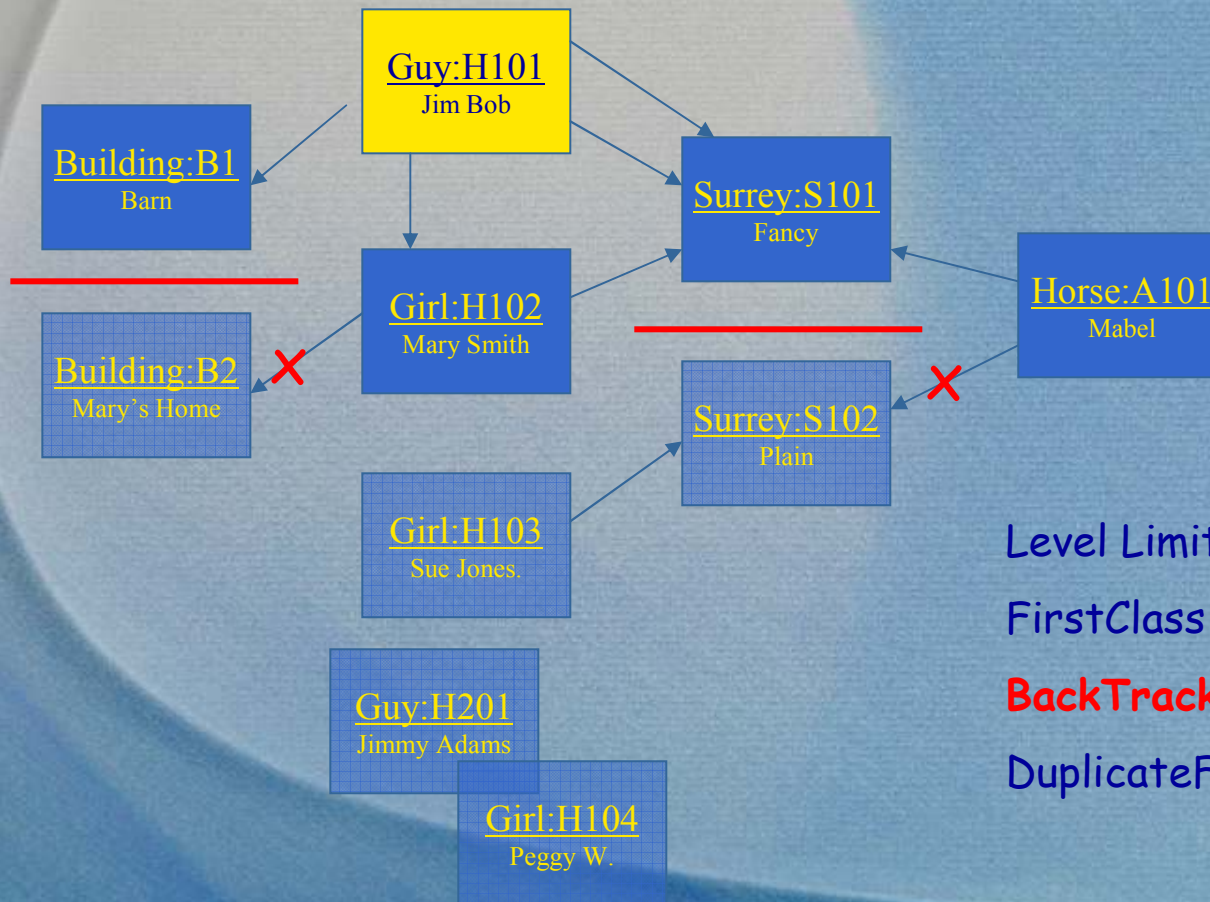
FirstClass = Off

BackTrack = Off

DuplicateFilter = On

Association Mapping – no backtracking

ObjectFreeAssociations('Guy','H101',5,0,1,1)



Level Limit = 5

FirstClass = Off

BackTrack = ClassBlocking

DuplicateFilter = On

Association Mapping – class blocking

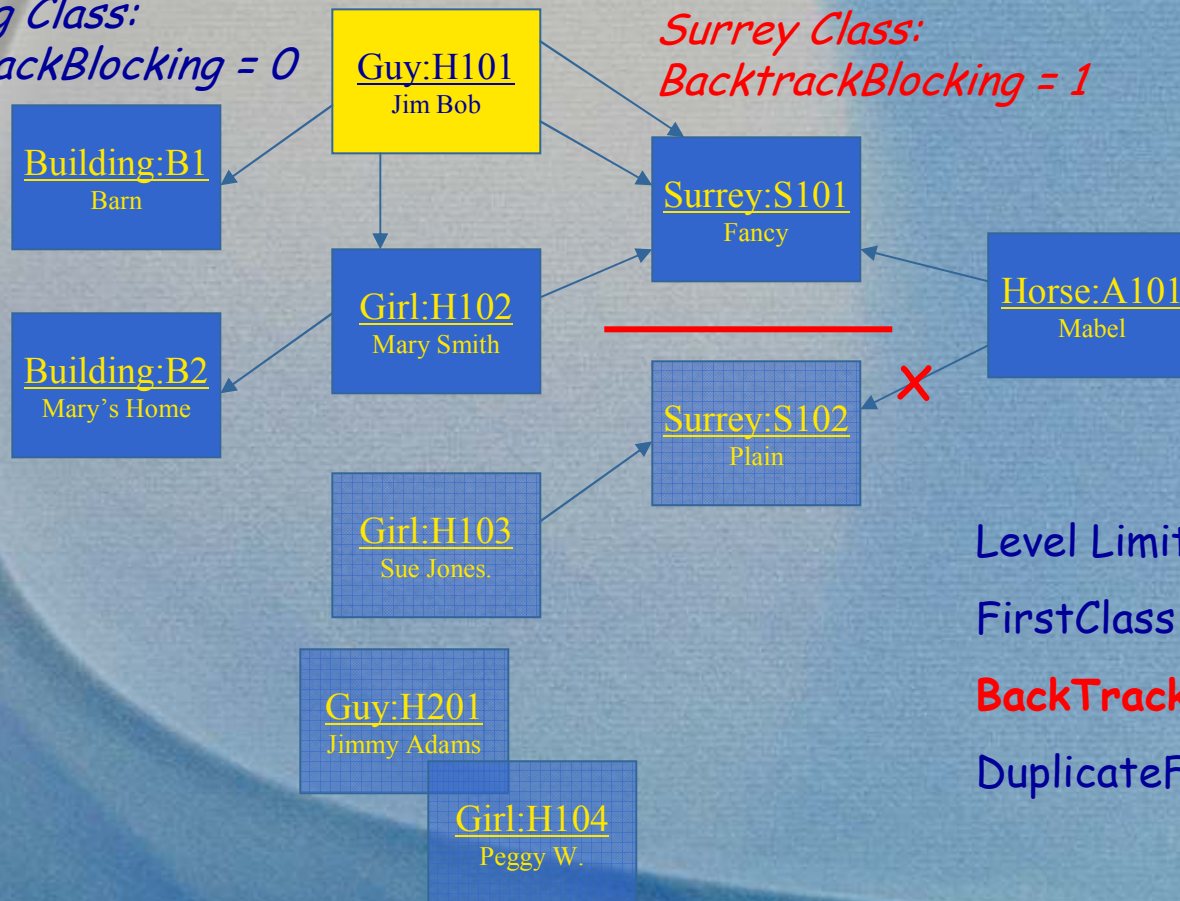
ObjectFreeAssociations('Guy','H101',5,0,2,1)

Building Class:

BacktrackBlocking = 0

Surrey Class:

BacktrackBlocking = 1



Level Limit = 5

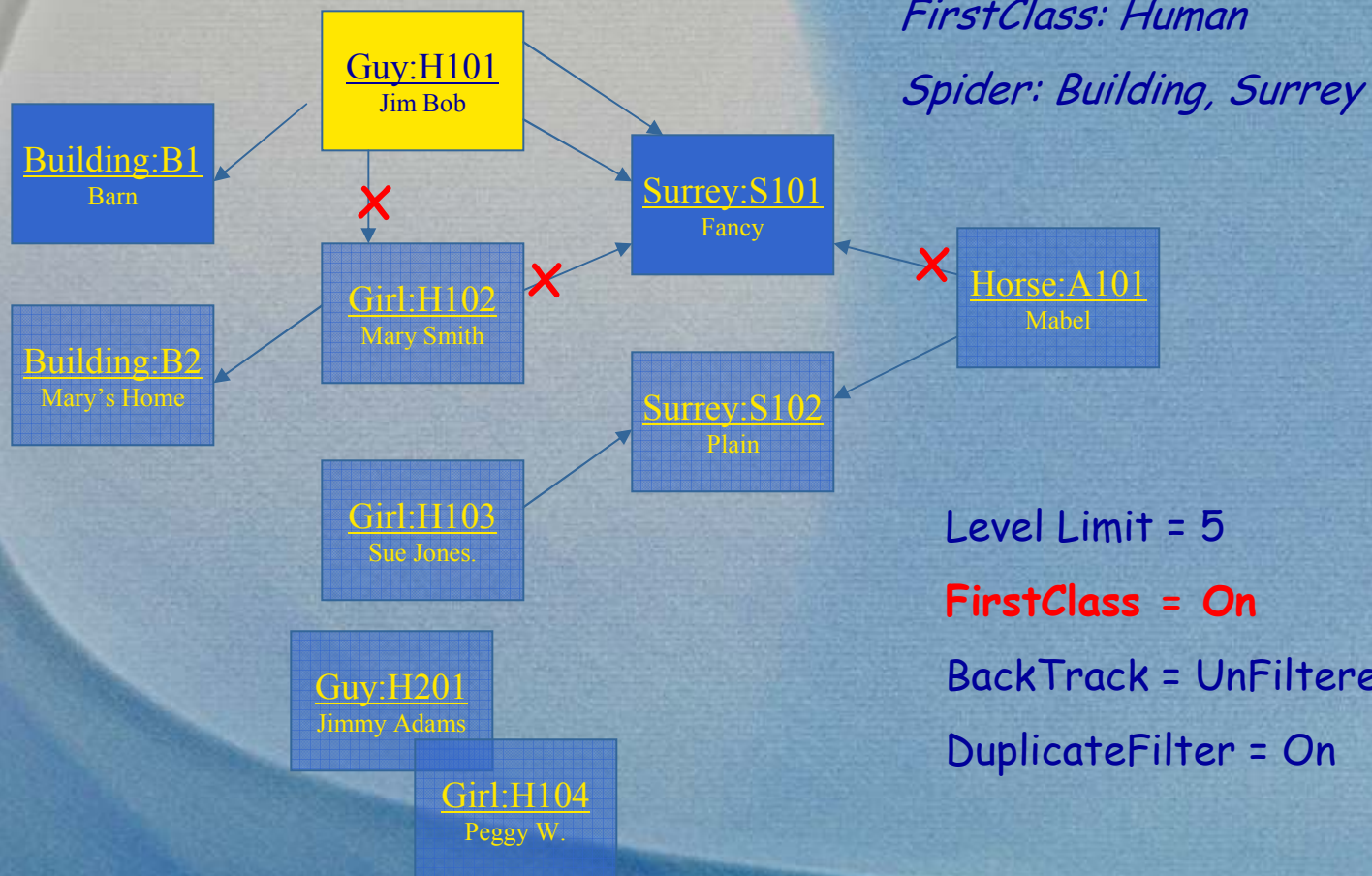
FirstClass = Off

BackTrack = ClassBlock

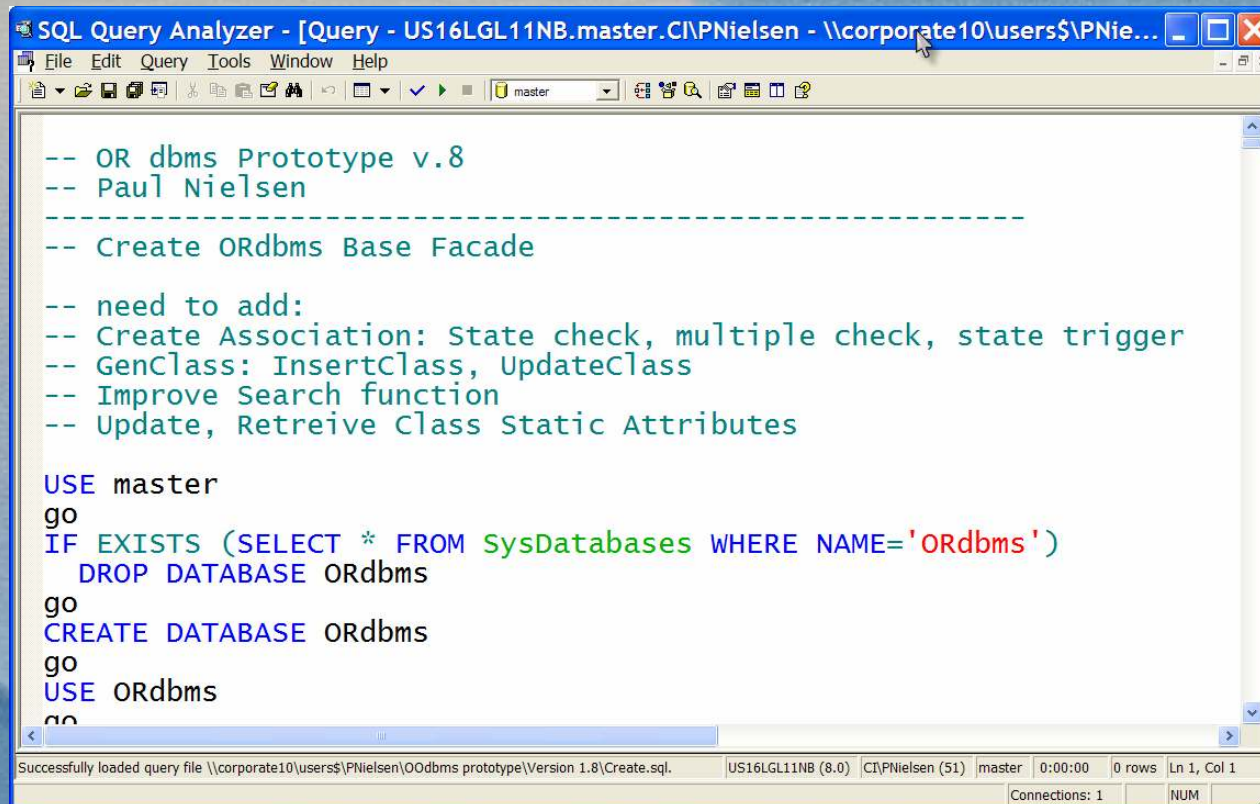
DuplicateFilter = On

Association Mapping – First Class

ObjectFreeAssociations('Guy','H101',5,1,0,1)



Live Code #5 - Spidering



```
SQL Query Analyzer - [Query - US16LGL11NB.master.C:\PNielsen - \\corporate10\users$\PNie...
File Edit Query Tools Window Help
master
-- OR dbms Prototype v.8
-- Paul Nielsen
-----
-- Create ORdbms Base Facade

-- need to add:
-- Create Association: State check, multiple check, state trigger
-- GenClass: InsertClass, UpdateClass
-- Improve Search function
-- Update, Retrieve Class Static Attributes

USE master
go
IF EXISTS (SELECT * FROM SysDatabases WHERE NAME='ORdbms')
    DROP DATABASE ORdbms
go
CREATE DATABASE ORdbms
go
USE ORdbms
go

Successfully loaded query file \\corporate10\users$\PNielsen\Oodbms prototype\Version 1.8\Create.sql.
US16LGL11NB (8.0) C:\PNielsen (51) master 0:00:00 0 rows Ln 1, Col 1
Connections: 1 NUM
```

Aggregation Associations (Collections)

- Multiple Participating Classes
- Rule Driven

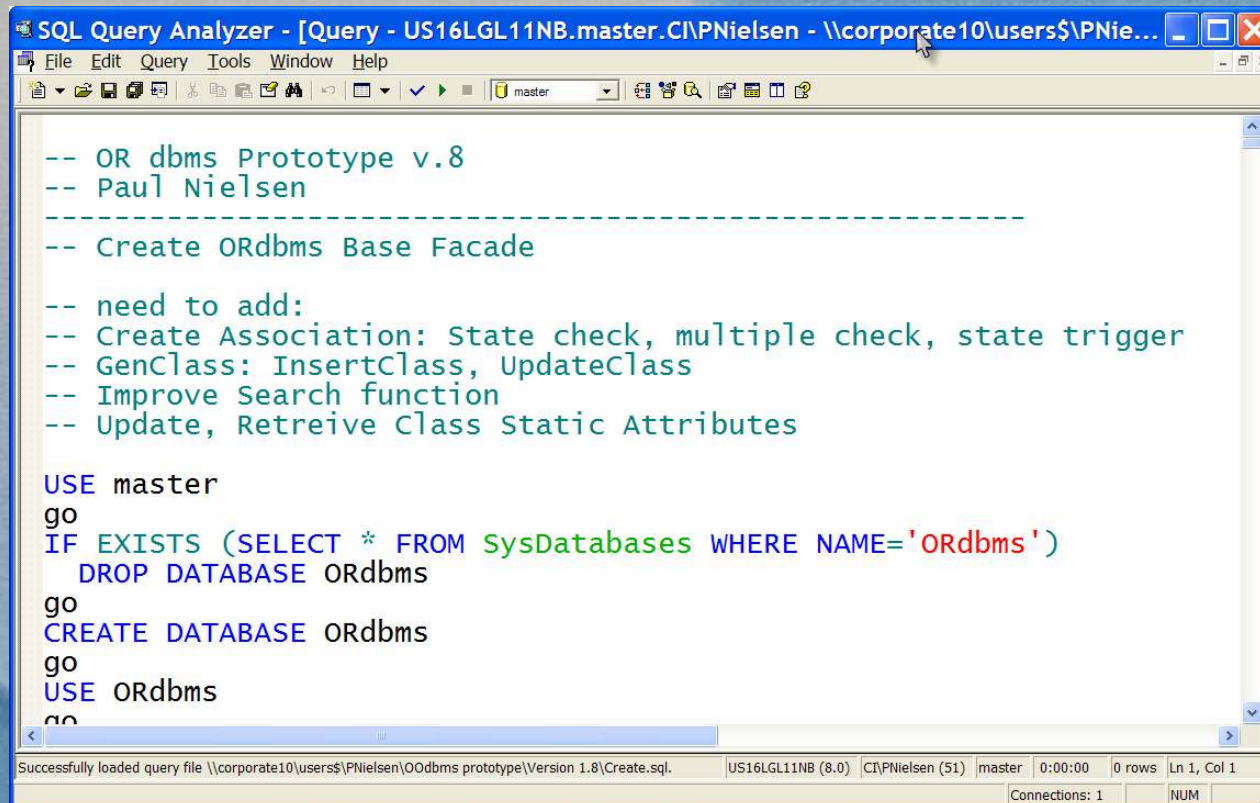
Surrey with four horses

- Collection – Horse Pulls Surrey
 - Surrey
 - Horse
- Add:
 - Horse
 - Horse
 - Horse

Complex Collection

- Collection – School Class
 - 1 Classroom
 - 1 Teacher
 - 1 Teacher Assistant
 - 25 students
 - 28 desks
 - 8 computers

Live Code #6 - Collections



The screenshot shows a window titled "SQL Query Analyzer - [Query - US16LGL11NB.master.CI\PNielsen - \\corporate10\users\$\PNie...". The window contains a SQL script with the following content:

```
-- OR dbms Prototype v.8
-- Paul Nielsen
-----
-- Create ORdbms Base Facade

-- need to add:
-- Create Association: State check, multiple check, state trigger
-- GenClass: InsertClass, UpdateClass
-- Improve Search function
-- Update, Retrieve Class Static Attributes

USE master
go
IF EXISTS (SELECT * FROM SysDatabases WHERE NAME='ORdbms')
    DROP DATABASE ORdbms
go
CREATE DATABASE ORdbms
go
USE ORdbms
go
```

The status bar at the bottom of the window displays: "Successfully loaded query file \\corporate10\users\$\PNielsen\Oodbms prototype\Version 1.8\Create.sql. US16LGL11NB (8.0) CI\PNielsen (51) master | 0:00:00 | 0 rows | Ln 1, Col 1 | Connections: 1 | NUM".

Nordic: Next Steps

- 37,000 Feet
- Class Management
- Object Management
- Association Management
- **Next Steps**

Next Steps

- Performance Tests / Tuning
- Class Designer GUI
- Object Browser GUI
- Web-Services Object Service
- Visio Class Diagram

Thanks,

www.nordicdb.com