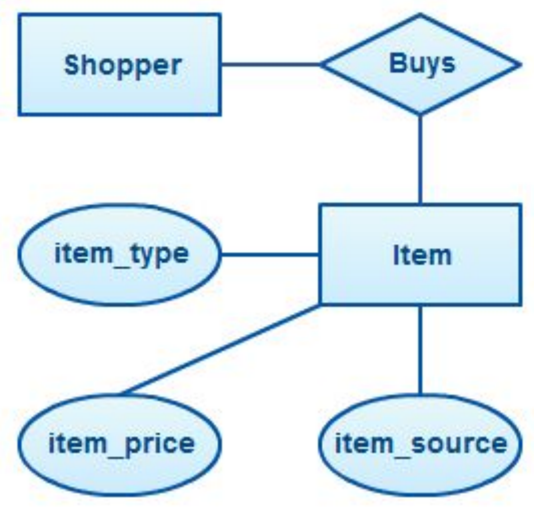
MONTANA
STATE UNIVERSITY

Database Design

Designing a Proper Database

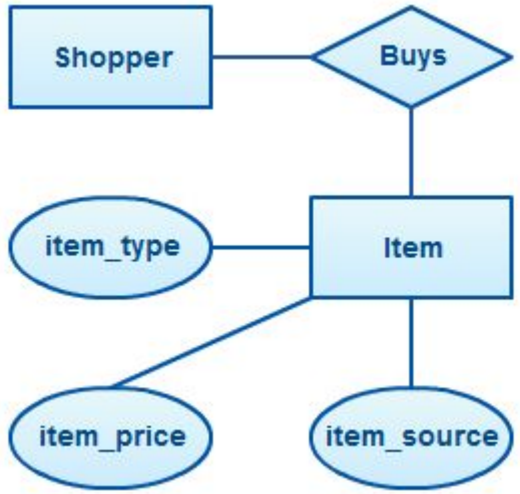
Designing A Database

- We have worked with E/R diagrams
- Often a good *first pass* at a database design
- Makes your needs concrete
  - May expose data problems
  - May give you insights into what data is needed
  - N-to-n → Entity transformation is often valuable



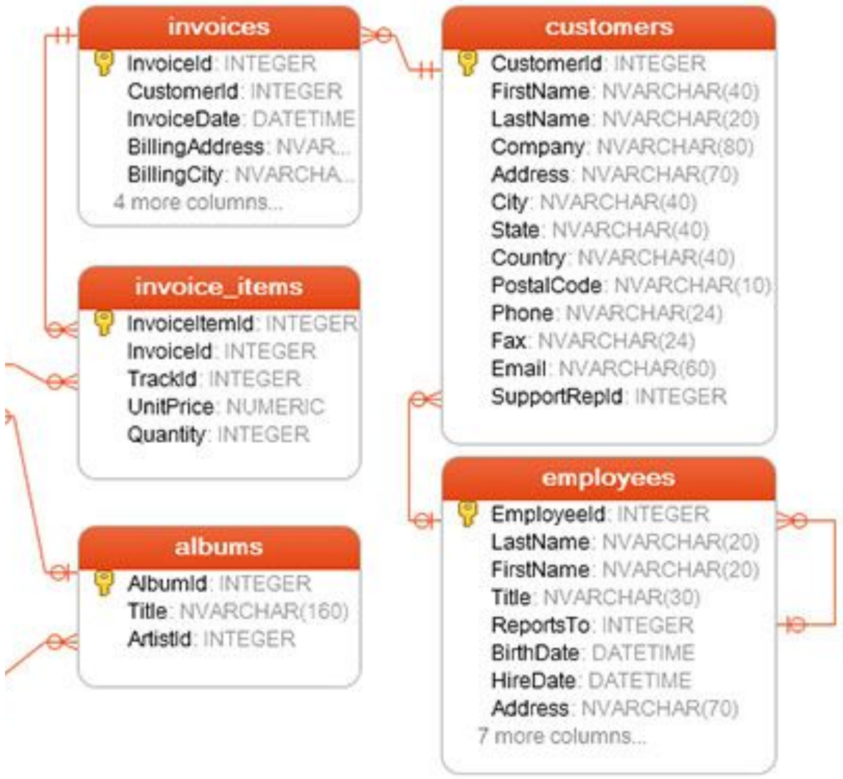
Designing A Database

- General Process
  - What are the core nouns in my system?
  - What are their relationships?
    - 1-1
    - 1-n
    - N-n (hmm)
  - What other nouns are needed to properly model things?
    - Things I haven't thought of
    - Decomposing existing nouns
  - GOTO step 1



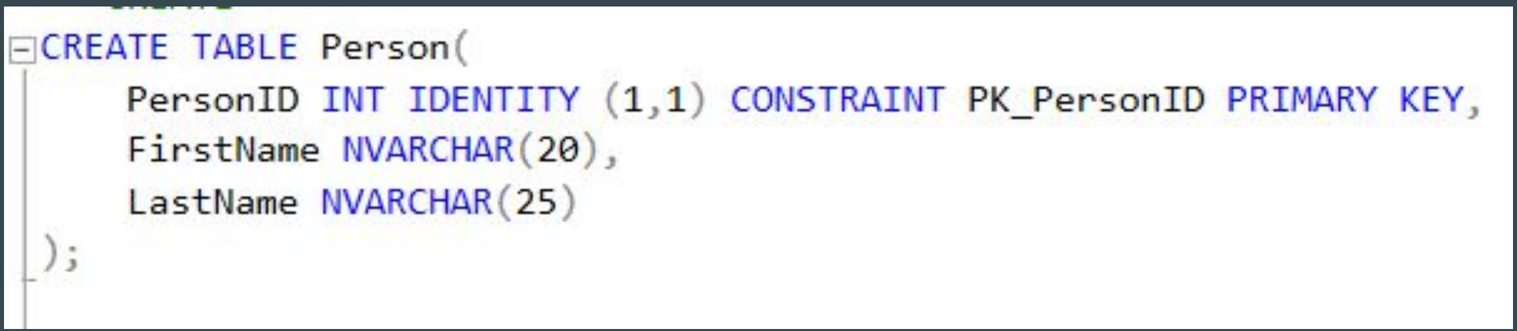
Designing A Database

- Once you have an E/R diagram you can transform it into a relational schema
- A relational schema typically reveals more detailed needs in your data model



Designing A Database

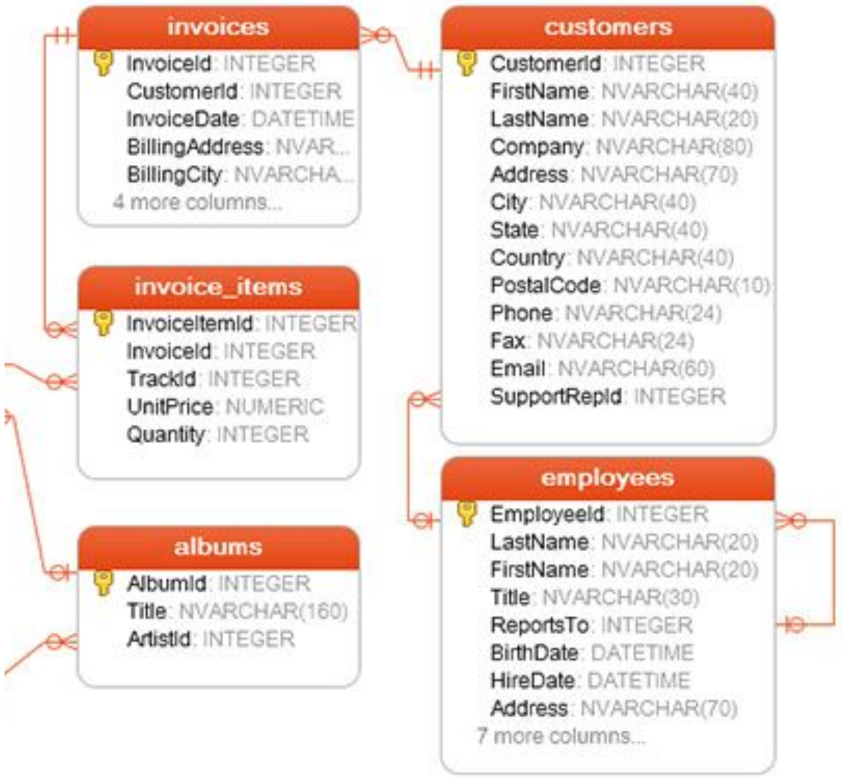
- Finally, this can be transformed into actual database definition statements
- Yet more details and issues will come up
- (we cover DDL later in the course)



```
CREATE TABLE Person(
    PersonID INT IDENTITY (1,1) CONSTRAINT PK_PersonID PRIMARY KEY,
    FirstName NVARCHAR(20),
    LastName NVARCHAR(25)
);
```

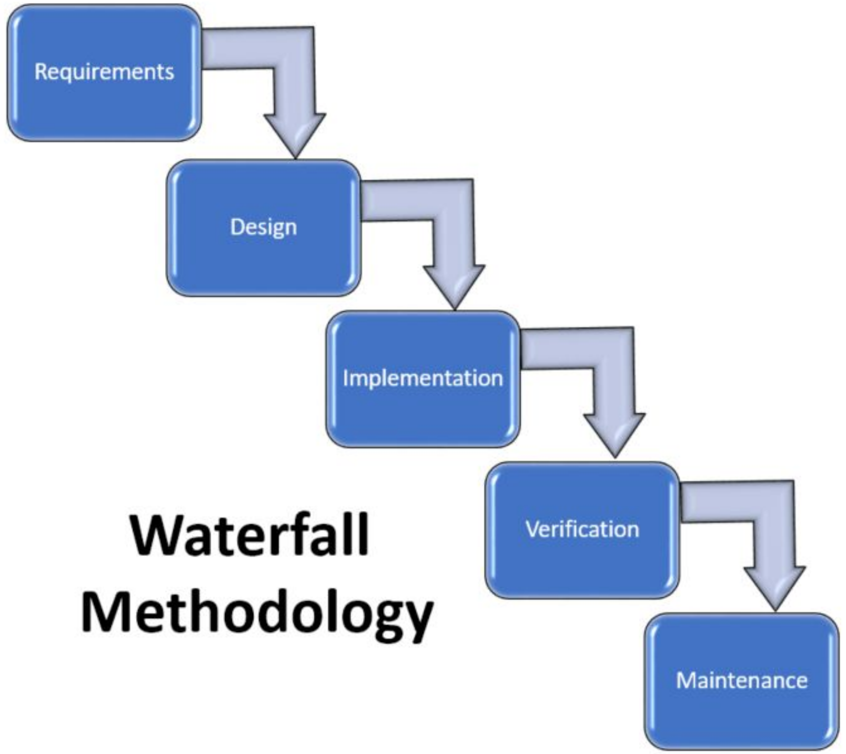
E/R to Database Translation

- Entity → Table
- Relation → FK(s)
  - 1-1 - pick the “lesser” relation to store the fk on
  - 1-n - store the fk on the N side of the relation
  - N-n - Is there an entity here?
    - If yes, add it
    - If no, create a join table (a table with two FKs only)



Designing A Database

- How realistic is this?
  - E/R → Relational Schema → DDL?
- Typically… not very realistic



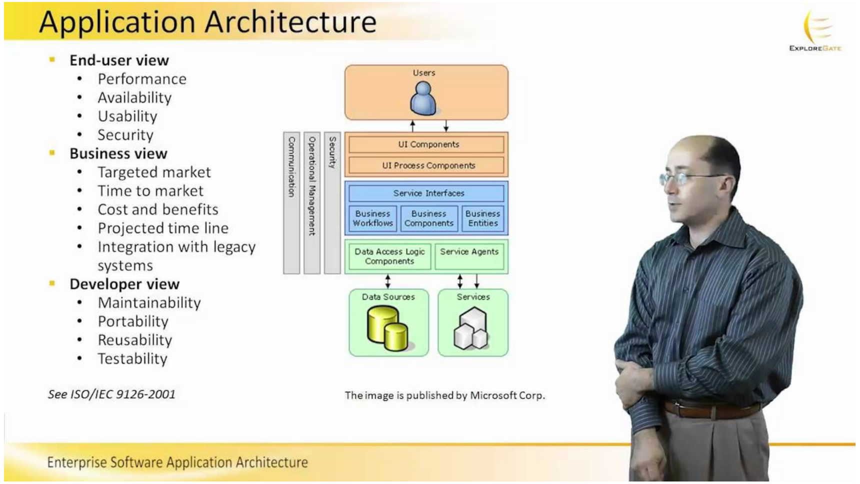
Designing A Database

"No plan survives contact with the enemy."

--Helmuth von Moltke the Elder

Designing A Database

- As you begin working with the database schema, problems will inevitable come up
- You will address them in the database directly
- The E/R diagrams will become increasingly out of date



Application Architecture

- **End-user view**
  - Performance
  - Availability
  - Usability
  - Security
- **Business view**
  - Targeted market
  - Time to market
  - Cost and benefits
  - Projected time line
  - Integration with legacy systems
- **Developer view**
  - Maintainability
  - Portability
  - Reusability
  - Testability

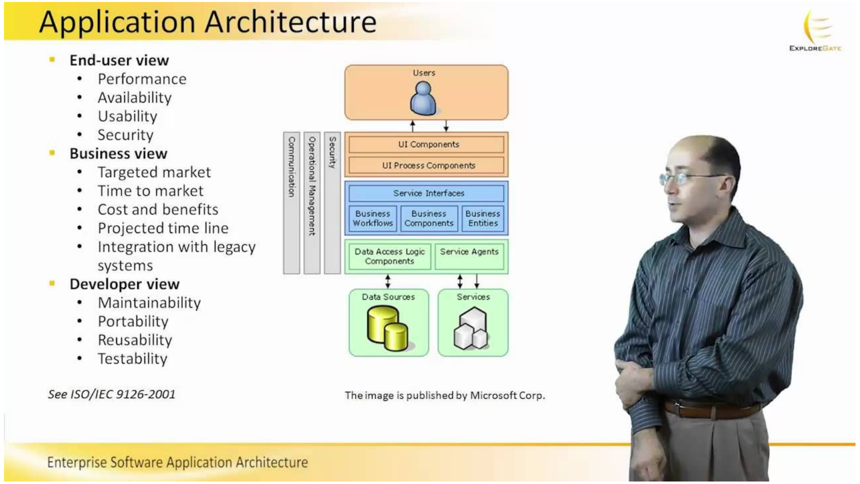
See ISO/IEC 9126-2001

The image is published by Microsoft Corp.

Enterprise Software Application Architecture

Designing A Database

- Application architects can become increasingly irrelevant if they continue to use E/R Diagrams and other artifacts to discuss the system
  - They may also become increasingly annoying
- Maintain a sense of humor during this process



Application Architecture

- End-user view
  - Performance
  - Availability
  - Usability
  - Security
- Business view
  - Targeted market
  - Time to market
  - Cost and benefits
  - Projected time line
  - Integration with legacy systems
- Developer view
  - Maintainability
  - Portability
  - Reusability
  - Testability

See ISO/IEC 9126-2001

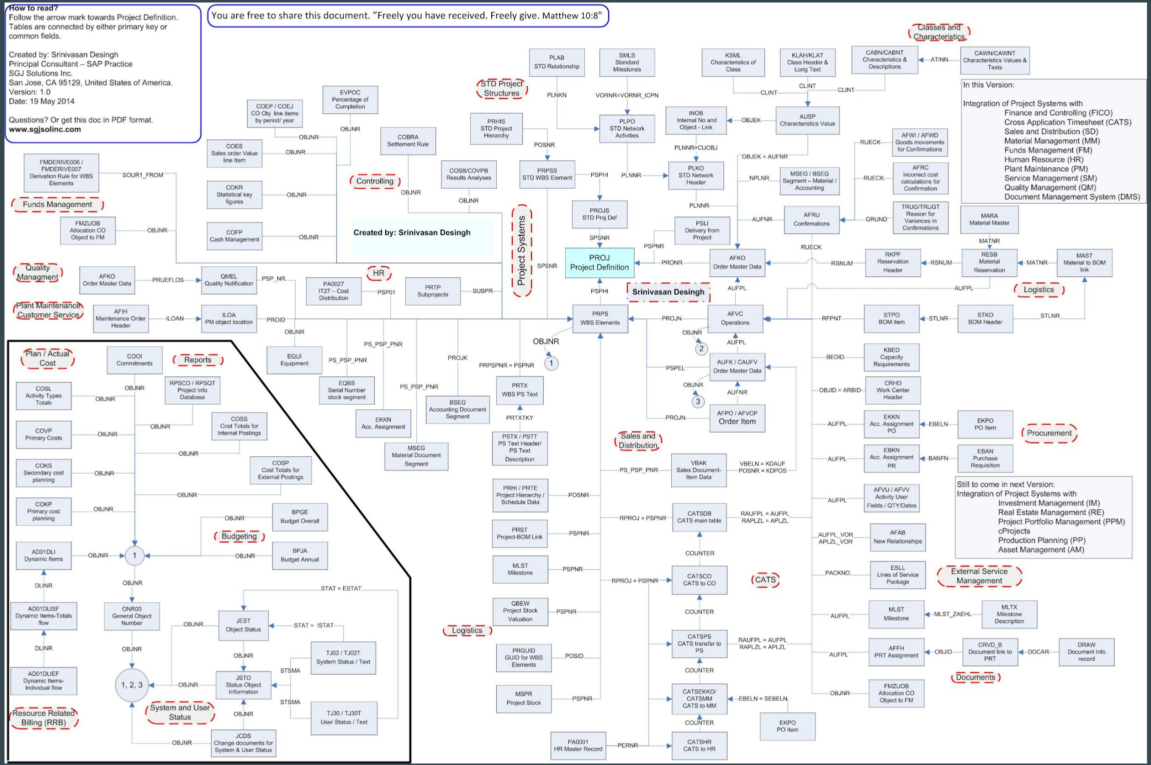
The image is published by Microsoft Corp.

Enterprise Software Application Architecture

Case Study - SAP

- SAP is one of the most valuable software companies in the world
- Worth 2 billion dollars today
- Sells software to help big companies manage things like inventory
- Here is a small part of their schema...

Case Study - SAP



How to read?
Follow the arrow mark towards Project Definition.
Tables are connected by either primary key or common fields.

Created by: Srinivasan Desingh
Principal Consultant – SAP Practice
SGJ Solutions Inc.
San Jose, CA 95129, United States of America.
Version: 1.0
Date: 19 May 2014

Questions? Or get this doc in PDF format.
www.sgjsolinc.com

You are free to share this document. "Freely you have received. Freely give. Matthew 10:8"

In this Version:

Integration of Project Systems with
- Finance and Controlling (FICO)
- Cross Application Timesheet (CATS)
- Sales and Distribution (SD)
- Material Management (MM)
- Funds Management (FM)
- Human Resource (HR)
- Plant Maintenance (PM)
- Service Management (SM)
- Quality Management (QM)
- Document Management System (DMS)

Still to come in next Version:
Integration of Project Systems with
- Investment Management (IM)
- Real Estate Management (RE)
- Project Portfolio Management (PPM)
- cProjects
- Production Planning (PP)
- Asset Management (AM)

Created by: Srinivasan Desingh

PROJ Project Definition
Project Systems
Srinivasan Desingh
STD Project Structures
Classes and Characteristics
Controlling
Funds Management
Quality Managment
Plant Maintenance Customer Service
HR
Logistics
Procurement
Sales and Distribution
CATS
External Service Management
Documents
Plan / Actual Cost
Reports
Budgeting
System and User Status
Resource Related Billing (RRB)
PLAB STD Relationship
SMLS Standard Milestones
KSML Characteristics of Class
KLAH/KLAT Class Header & Long Text
CABN/CABNT Characteristics & Descriptions
CAWN/CAWNT Characteristics Values & Texts
INOB Internal No and Object - Link
AUSP Characteristics Value
PRHIS STD Project Hierarchy
PLPO STD Network Activities
PRPSS STD WBS Element
PLKO STD Network Header
PROJS STD Proj Def
PSLI Delivery from Project
MSEG / BSEG Segment – Material / Accounting
AFRU Confirmations
AFWI / AFWD Goods movements for Confirmations
AFRC Incorrect cost calculations for Confirmation
TRUG/TRUGT Reason for Variances in Confirmations
MARA Material Master
AFKO Order Master Data
RKPF Reservation Header
RESB Material Reservation
MAST Material to BOM link
PRPS WBS Elements
AFVC Operations
STPO BOM Item
STKO BOM Header
AUFK / CAUFV Order Master Data
AFPO / AFVCP Order Item
KBED Capacity Requirements
CRHD Work Center Header
EKKN Acc. Assignment PO
EKPO PO Item
EBKN Acc. Assignment PR
EBAN Purchase Requisition
AFVU / AFVV Activity User Fields / QTY/Dates
AFAB New Relationships
ESLL Lines of Service Package
MLST Milestone
MLTX Milestone Description
AFFH PRT Assignment
CRVD_B Document link to PRT
DRAW Document Info record
FMZUOB Allocation CO Object to FM
EVPOC Percentage of Completion
COEP / COEJ CO Obj line items by period/ year
COES Sales order Value line Item
COKR Statistical key figures
COFP Cash Management
COBRA Settlement Rule
COSB/COVPB Results Analyses
FMDERIVE006 / FMDERIVE007 Derivation Rule for WBS Elements
FMZUOB Allocation CO Object to FM
AFKO Order Master Data
QMEL Quality Notification
AFIH Maintenance Order Header
ILOA PM object location
PA0027 IT27 – Cost Distribution
PRTP Subprojects
EQUI Equipment
EQBS Serial Number stock segment
EKKN Acc. Assignment
MSEG Material Document Segment
BSEG Accounting Document Segment
PRTX WBS PS Text
PSTX / PSTT PS Text Header/ PS Text Description
PRHI / PRTE Project Hierarchy / Schedule Data
PRST Project-BOM Link
MLST Milestone
QBEW Project Stock Valuation
PRGUID GUID for WBS Elements
MSPR Project Stock
PA0001 HR Master Record
VBAK Sales Document-Item Data
CATSDB CATS main table
CATSCO CATS to CO
CATSPS CATS transfer to PS
CATSEKKO/ CATSMM CATS to MM
CATSHR CATS to HR
EKPO PO Item
COOI Commitments
RPSCO / RPSQT Project Info Database
COSL Activity Types Totals
COSS Cost Totals for Internal Postings
COVP Primary Costs
COSP Cost Totals for External Postings
COKS Secondary cost planning
COKP Primary cost planning
BPGE Budget Overall
BPJA Budget Annual
AD01DLI Dynamic Items
AD01DLISF Dynamic Items-Totals flow
AD01DLIEF Dynamic Items-Individual flow
ONR00 General Object Number
JEST Object Status
JSTO Status Object Information
TJ02 / TJ02T System Status / Text
TJ30 / TJ30T User Status / Text
JCDS Change documents for System & User Status
1, 2, 3
OBJNR
PSPHI
PSPNR
POSNR
PLNKN
VORNR=VORNR_ICPN
CLINT
ATINN
OBJEK
OBJEK = AUFNR
PLNNR=CUOBJ
PLNNR
NPLNR
AUFNR
RUECK
GRUND
MATNR
RSNUM
AUFPL
PRONR
SPSNR
PSPEL
PROJN
RFPNT
STLNR
BEDID
OBJID = ARBID
EBELN
BANFN
AUFPL_VOR APLZL_VOR
PACKNO
MLST_ZAEHL
OBJID
DOCAR
SOUR1_FROM
PRUEFLOS
PSP_NR
ILOAN
PROID
PSP01
SUBPR
PS_PSP_PNR
PROJK
PRPSPNR = PSPNR
PRTXTKY
VBELN = KDAUF POSNR = KDPOS
RAUFPL = AUFPL RAPLZL = APLZL
RPROJ = PSPNR
COUNTER
EBELN = SEBELN
PERNR
POSID
DLINR
STAT = ESTAT
STAT = ISTAT
STSMA

Case Study - SAP

- The SAP schema is famously *a nightmare*
  - It is **huge**
    - **More than 8000 tables**
  - Large parts of it make no sense
  - The table names are often in german
  - There are normalization issues all over it
    - We will talk about normalization in the next lecture
  - **SAP is still worth 2 billion dollars**

Advice On Database Modeling

- As the system matures, accept that data modeling mistakes made early on are going to stick with you
- The database schema get harder to fix over time
  - That’s OK! Learn to live with it.
- QUESTION: Why is a database so much harder to fix than code?
- LeadDyno DB Story Time!

Advice On Database Modeling

- Start with E/R diagramming
  - But don’t take it too seriously
- Get to a real world database **ASAP**
- Iterate your app and database as quickly as possible early on when database size is relatively small
- Try to figure out the crucial entities in your system and get those as right as possible
- Read up on stoicism

Polymorphism & Databases

•••

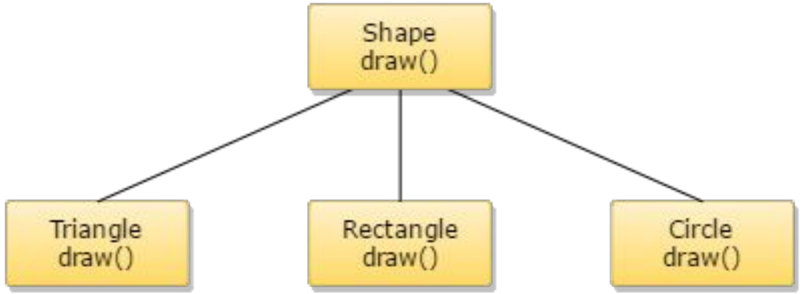
A Mismatch?

Coding With Databases

- Almost all databases are fronted by some software
- That software is written in a particular language
- That language is *probably* object oriented
    - Java (this class)
    - Javascript :(
    - PHP XD

Polymorphism

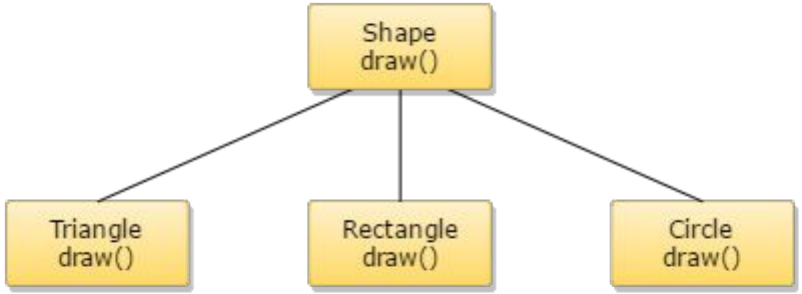
- You are probably familiar with this idea from your Object Oriented classes
  - Super-classes
  - Sub-classes
  - Sub-classes extend the super-class
    - Add methods and ***attributes***



Polymorphism

The Relational Model

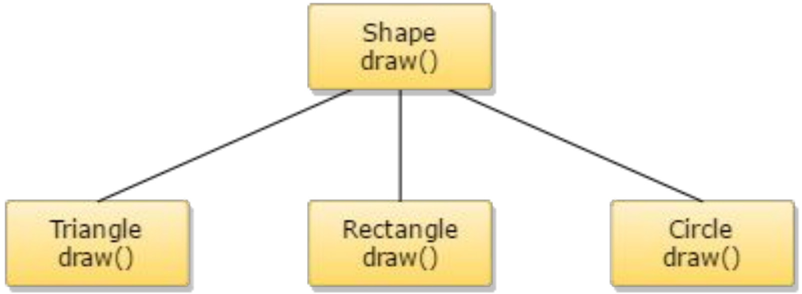
- Relations are just tables and foriegn keys
- How should we model this object hierarchy if we want to store it in a table?



Polymorphism

Three Approaches

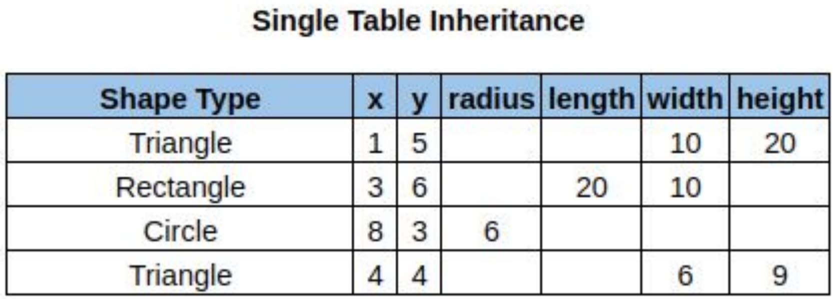
- Single Table Inheritance
- Class Table Inheritance
- Concrete Table Inheritance



Polymorphism

Single Table Inheritance

- A single table is used for all sub-class instances
- The columns are the union of all columns of sub-classes
- Advantages?
- Disadvantages?



Single Table Inheritance

Shape Type	x	y	radius	length	width	height
Triangle	1	5			10	20
Rectangle	3	6		20	10	
Circle	8	3	6			
Triangle	4	4			6	9

Class Table Inheritance

- There is one table per class in the object hierarchy
- Sub-classes include a foreign key reference to their parent classes
- Advantages?
- Disadvantages?

Shapes

id	x	y
1	1	5
2	3	6
3	8	3
4	4	4

Circles

shape_id	radius
3	6

Rectangles

shape_id	length	width
2	20	10

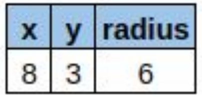
Triangles

shape_id	length	width
1	10	20
4	6	9

Concrete Table Inheritance

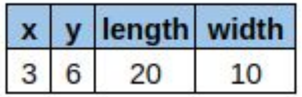
- There is one table per **concrete** class in your object hierarchy
- Advantages?
- Disadvantages?

Circles



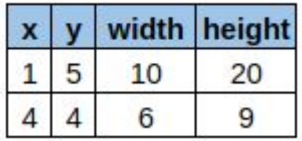
x	y	radius
8	3	6

Rectangles



x	y	length	width
3	6	20	10

Triangles



x	y	width	height
1	5	10	20
4	4	6	9

Polymorphism In Practice

- Most systems that I have experience with use a mix:
  - Single Table Inheritance for closely related things with many foreign keys in common
  - Concrete Table Inheritance for more distant relations
  - I have never seen class table inheritance work out well
- My advice is to focus on foreign keys: *the more keys* two objects have in common, the more likely you are to prefer *single table inheritance*

What Are We Dealing With Here?

- This is a single instance of a more general problem...

The Object-Relational Impedance Mismatch

“The object-relational impedance mismatch is a set of conceptual and technical difficulties that are often encountered when a relational database management system (RDBMS) is being served by an application program (or multiple application programs) written in an object-oriented programming language or style, particularly because objects or class definitions must be mapped to database tables defined by a relational schema.”

The Object-Relational Impedance Mismatch

- While inheritance is one issue, there are many others:
  - Encapsulation isn’t part of the relational model
  - Interfaces don’t exist at the relational level
  - Field accessibility isn’t specified at the relational level
  - Database transactions do not map well to objects
  - And so on
- We will discuss this more thoroughly when we talk about Object/Relational mapping tools

The Object-Relational Impedance Mismatch

- There is, however, a fundamental conceptual and cultural difference between Object Oriented and Relational thinking
- Object Oriented
  - Imperative
  - Nouns + Verbs
  - Graphs
- Relational Thinking
  - Declarative
  - Normalized
  - Set relationships

Object Databases

- Some people reject the relational model entirely
- OODBMS (Object-oriented database management systems) explicitly encode OO semantics
- Smalltalk/GemTalk
- Java/db4o
- A small but *passionate* community

GEMTALK™ SYSTEMS

db4objects BY VERSANT

MONTANA
STATE UNIVERSITY