Data Normalization (2)

IS 240 – Database

Lecture #5 – 2010-02-08

M. E. Kabay, PhD, CISSP-ISSMP

Assoc. Prof. Information Assurance
School of Business & Management, Norwich University

mailto:mkabay@norwich.edu V: 802.479.7937

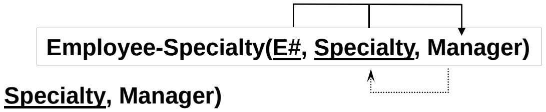
Don't Panic!

- We are not trying to discuss 66 slides in one session
- Much of today's material will serve as reference and review for your studies
- Opportunity to discuss principles and illustrate them with specific cases

Overview of Today's Work

- Continue discussion of normalization
- Work on examples together
- Important to get involved in class discussion
- Can absorb / integrate concepts only by using them
- Do not try to memorize without understanding
  - ❑Waste of your time
- Do the homework exercises seriously
  - ❑They will ensure success
  - ❑Students from previous IS240 classes have reported good experiences in real jobs – praised for understanding principles that help design good databases

Boyce-Codd Normal Form (BCNF)



- Hidden dependency
- Example:
  - ❑ Employee-Specialty(E#, Specialty, Manager)
  - ❑ Is in 3NF now.
- Business rules.
  - ❑ Employee may have many specialties.
  - ❑ Each specialty has many managers.
  - ❑ Each manager has only one specialty.
  - ❑ Employee has only one manager for each specialty.
- Problem is hidden relationship between manager and specialty.
  - ❑ Need separate table for manager.
  - ❑ But then we don't need to repeat specialty.
- In real life, probably accept the duplication (specialty listed in both tables).

Employee(E#, Manager)

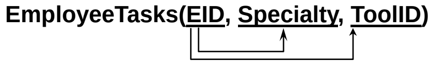
Manager(Manager, Specialty)

Employee(E#, Specialty, Manager)

Manager(Manager, Specialty)

acceptable

Fourth Normal Form (Keys)



- Technically, if you keyed every column, any table would be in 3NF, which does not solve any problems.
- In some cases, there are hidden relationships between key properties.
- Example:
  - ❑ EmployeeTasks(EID, Specialty, ToolID)
  - ❑ In 3NF (BCNF) now.
- Business Rules
  - ❑ Each employee has many specialties.
  - ❑ Each employee has many tools.
  - ❑ Tools and specialties are unrelated

EmployeeSpecialty(EID, Specialty)

EmployeeTools(EID, ToolID)

Domain-Key Normal Form (DKNF)

- DKNF is ultimate goal: table will always be in 4NF, etc.
- Drawbacks
  - ❑ No mechanical method to get to DKNF
  - ❑ No guarantee a table can be converted to DKNF
- Rules
  - ❑ Table => one topic
  - ❑ All business rules explicitly written as domain constraints and key relationships.
  - ❑ No hidden relationships.

Employee(EID, Name, speciality)

Business rule: *An employee can have many specialties.*
So example is not in DKNF, since EID is not unique.

DKNF Examples

Employee(EID, Name, Speciality)

Business rule: *An employee can have many specialties.*
Example is not in DKNF: EID is not unique.

Employee(EID, Name, Speciality)

Business rule: *An employee has one name.*
Example is not DKNF: hidden relationship between EID and name

Employee(EID, Name)

EmployeeSpecialty(EID, Speciality)

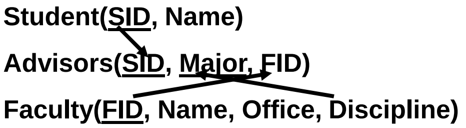
DKNF Examples

Student(SID, Name, Major, Advisor)

Advisor(FID, Name, Office, Discipline)

Business rules: *A student can have many advisors, but only one for each major. Faculty can only be advisors for their discipline.*

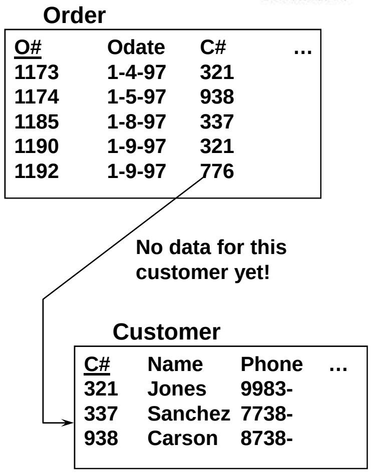
Not in DKNF: Primary key and hidden rule.


Student(SID, Name)
Advisors(SID, Major, FID)
Faculty(FID, Name, Office, Discipline)

DKNF: Foreign key (Major <--> Discipline) makes advisor rule explicit.

Data Rules and Integrity

- Simple business rules
  - ❑ Limits on data ranges
    - ✓ Price > 0
    - ✓ Salary < 100,000
    - ✓ DateHired > 1/12/1995
  - ❑ Choosing from a set
    - ✓ Gender = M, F, Unknown
    - ✓ Jurisdiction=City, County, State, Federal
- Referential Integrity
  - ❑ Foreign key values in one table must exist in the master table.
  - ❑ Order(O#, Odate, C#,…)
  - ❑ C# must exist in the customer table.



Order

O#	Odate	C#	…
1173	1-4-97	321	
1174	1-5-97	938	
1185	1-8-97	337	
1190	1-9-97	321	
1192	1-9-97	776	

No data for this customer yet!

Customer

C#	Name	Phone	…
321	Jones	9983-	
337	Sanchez	7738-	
938	Carson	8738-	

Effect of Business Rules

Location Date Played					Referee Name Phone Number, Address				
Team 1 Name Sponsor		Score			Team 2 Name Sponsor		Score		
Player Name	Phone	Age	Points	Penalties	Player Name	Phone	Age	Points	Penalties

Key business rules:

A player can play on only one team.

There is one referee per match.

Business Rules 1

There is one referee per match.
A player can play on only one team.

Match(MatchID, DatePlayed, Location, RefID)
Score(MatchID, TeamID, Score)
Referee(RefID, Phone, Address)
Team(TeamID, Name, Sponsor)
Player(PlayerID, Name, Phone, DoB, TeamID)
PlayerStats(MatchID, PlayerID, Points, Penalties)

*RefID and TeamID are **not** keys in the Match and Team tables, because of the one-to-one rules.*

Business Rules 2

There can be several referees per match.
A player can play on several teams (substitute), but only on one team per match.

Match(MatchID, DatePlayed, Location, RefID)
Score(MatchID, TeamID, Score)
Referee(RefID, Phone, Address)
Team(TeamID, Name, Sponsor)
Player(PlayerID, Name, Phone, DoB, TeamID)
PlayerStats(MatchID, PlayerID, Points, Penalties)

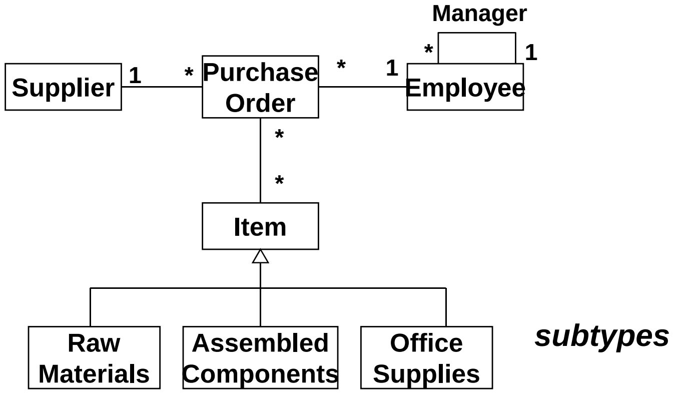
To handle the many-to-many relationship, we need to make RefID and TeamID keys. But if you leave them in the same tables, the tables are not in 3NF. DatePlayed does not depend on RefID. Player Name does not depend on TeamID.

Business Rules 2: Normalized

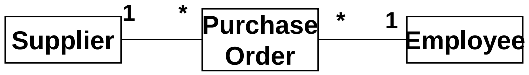
There can be several referees per match.
A player can play on several teams (substitute),
but only on one team per match.

Match(<u>MatchID</u>, DatePlayed, Location)
RefereeMatch(<u>MatchID</u>, <u>RefID</u>)
Score(<u>MatchID</u>, <u>TeamID</u>, Score)
Referee(<u>RefID</u>, Phone, Address)
Team(<u>TeamID</u>, Name, Sponsor)
Player(<u>PlayerID</u>, Name, Phone, DoB)
PlayerStats(<u>MatchID</u>, <u>PlayerID</u>, TeamID, Points, Penalties)

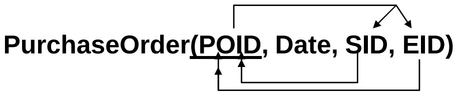
Converting a Class Diagram to Normalized Tables



One-to-Many Relationships



Supplier(<u>SID</u>, Name, Address, City, State, Zip, Phone)
Employee(<u>EID</u>, Name, Salary, Address, …)



PurchaseOrder(<u>POID</u>, Date, SID, EID)

The many side becomes a key (underlined).
Each PO has one supplier and employee.
(Do not key SID or EID)
Each supplier can receive many POs. *(Key PO)*
Each employee can place many POs. *(Key PO)*

One-to-Many Sample Data

Supplier

<u>ID</u>	Name	Address	City	State	Zip	Phone
5676	Jones	123 Elm	Ames	IA	50010	515-777-8988
6731	Markle	938 Oak	Boston	MA	02109	617-222-9999
7831	Paniche	873 Hickory	Jackson	MS	39205	601-333-9932
8872	Swensen	773 Poplar	Wichita	KS	67209	316-999-3312

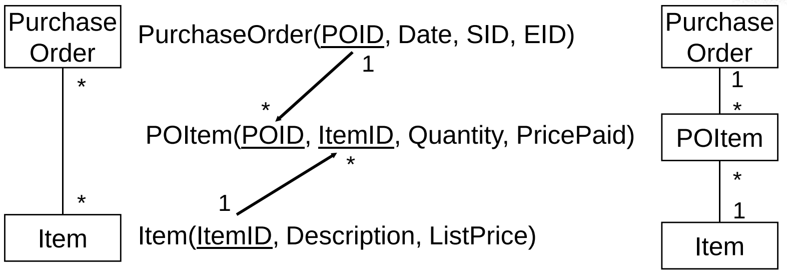
Purchase Order

<u>POID</u>	Date	SID	EID
22234	9-9-2004	5676	221
22235	9-10-2004	5676	554
22236	9-10-2004	7831	221
22237	9-11-2004	8872	335

Employee

<u>EID</u>	Name	Salary	Address
221	Smith	67,000	223 W. 2300
335	Sanchez	82,000	37 W. 7200
554	Johnson	35,000	440 E. 5200

Many-to-Many Relationships



PurchaseOrder(<u>POID</u>, Date, SID, EID)

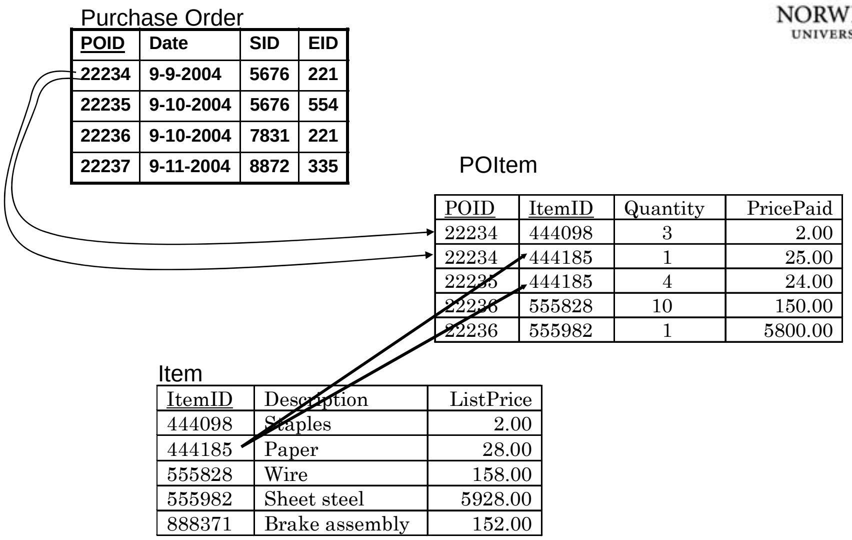
POItem(<u>POID</u>, <u>ItemID</u>, Quantity, PricePaid)

Item(<u>ItemID</u>, Description, ListPrice)

Each POID can have many Items (key/underline ItemID).
Each ItemID can be on many POIDs (key POID).

Need the new intermediate table (POItem) because:
You cannot put <u>ItemID</u> into PurchaseOrder because Date, SID, and EID do not depend on the ItemID.
You cannot put <u>POID</u> into Item because Description and ListPrice do not depend on POID.

Many-to-Many Sample Data



Purchase Order

<u>POID</u>	Date	SID	EID
22234	9-9-2004	5676	221
22235	9-10-2004	5676	554
22236	9-10-2004	7831	221
22237	9-11-2004	8872	335

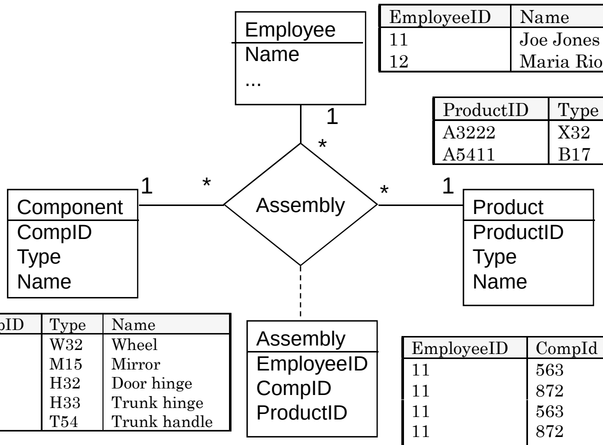
POItem

<u>POID</u>	<u>ItemID</u>	Quantity	PricePaid
22234	444098	3	2.00
22234	444185	1	25.00
22235	444185	4	24.00
22236	555828	10	150.00
22236	555982	1	5800.00

Item

<u>ItemID</u>	Description	ListPrice
444098	Staples	2.00
444185	Paper	28.00
555828	Wire	158.00
555982	Sheet steel	5928.00
888371	Brake assembly	152.00

N-ary Associations



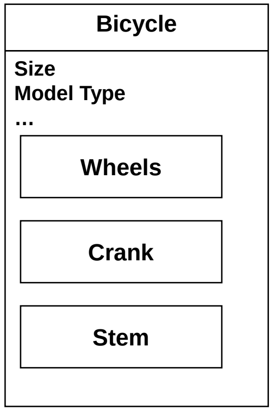
EmployeeID	Name	…
11	Joe Jones	…
12	Maria Rio	…

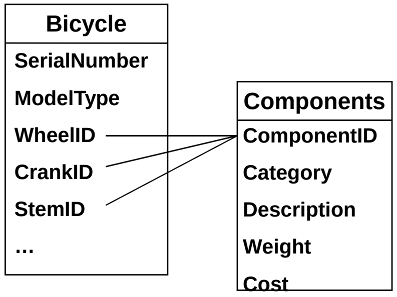
ProductID	Type	Name
A3222	X32	Corvette
A5411	B17	Camaro

CompID	Type	Name
563	W32	Wheel
872	M15	Mirror
882	H32	Door hinge
883	H33	Trunk hinge
888	T54	Trunk handle

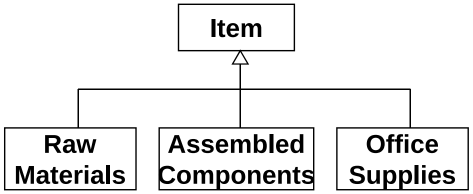
EmployeeID	CompId	ProductID
11	563	A3222
11	872	A3222
11	563	A5411
11	872	A5411
12	563	A3222
12	882	A3222
12	888	A3222
12	883	A5411

Composition





Generalization or Subtypes



Item(ItemID, Description, ListPrice)

RawMaterials(ItemID, Weight, StrengthRating)

AssembledComponents(ItemID, Width, Height, Depth)

OfficeSupplies(ItemID, BulkQuantity, Discount)

Add new tables for each subtype.
Use the same key as the generic type (ItemID)--one-to-one relationship.
Add the attributes specific to each subtype.

Subtypes Sample Data

Item

ItemID	Description	ListPrice
444098	Staples	2.00
444185	Paper	28.00
555828	Wire	158.00
555982	Sheet steel	5928.00
888371	Brake assembly	152.00

RawMaterials

ItemID	Weight	StrengthRating
555828	57	2000
555982	2578	8321

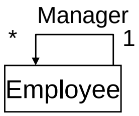
AssembledComponents

ItemID	Width	Height	Depth
888371	1	3	1.5

OfficeSupplies

ItemID	BulkQuantity	Discount
444098	20	10%
444185	10	15%

Recursive Relationships



Employee(EID, Name, Salary, Address, Manager)

Employee

EID	Name	Salary	Address	Manager
221	Smith	67,000	223 W. 2300	335
335	Sanchez	82,000	37 W. 7200	
554	Johnson	35,000	440 E. 5200	335

Add a manager column that contains Employee IDs.
An employee can have only one manager. *(Manager is not a key.)*
A manager can supervise many employees. *(EID is a key.)*

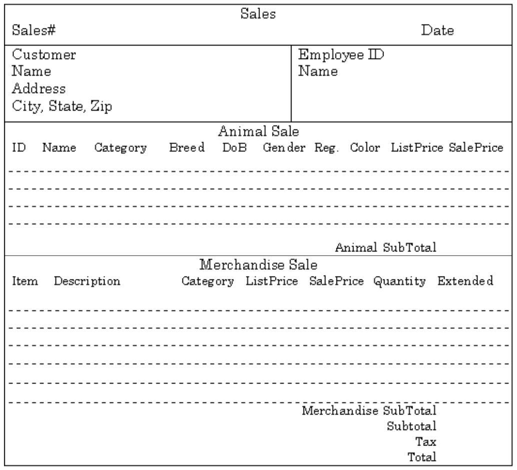
Normalization Examples

- **Possible topics**
  - ❑ **Auto repair**
  - ❑ **Auto sales**
  - ❑ **Department store**
  - ❑ **Hair stylist**
  - ❑ **HRM department**
  - ❑ **Law firm**
  - ❑ **Manufacturing**
  - ❑ **National Park Service**
  - ❑ **Personal stock portfolio**
  - ❑ **Pet shop**
  - ❑ **Restaurant**
  - ❑ **Social club**
  - ❑ **Sports team**

Multiple Views & View Integration

- Collect multiple views
  - ❑ Documents
  - ❑ Reports
  - ❑ Input forms
- Create normalized tables from each view
- Combine the views into one complete model.
- Keep meta-data in a data dictionary
  - ❑ Type of data
  - ❑ Size
  - ❑ Volume
  - ❑ Usage
- Example
  - ❑ Federal Emergency Management Agency (FEMA). Disaster planning and relief.
  - ❑ Make business assumptions as necessary, but try to keep them simple.

The Pet Store: Sales Form



Sales

Sales# Date

Customer
Name
Address
City, State, Zip

Employee ID
Name

Animal Sale

ID	Name	Category	Breed	DoB	Gender	Reg.	Color	ListPrice	SalePrice

Animal SubTotal

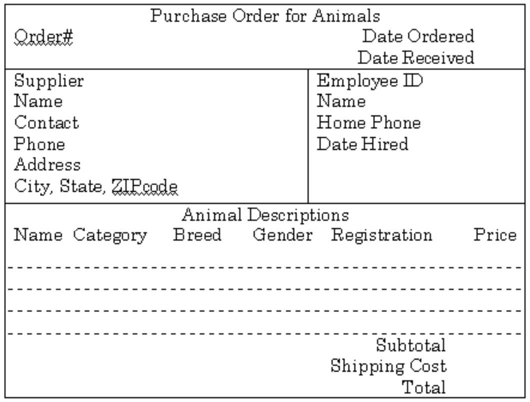
Merchandise Sale

Item	Description	Category	ListPrice	SalePrice	Quantity	Extended

Merchandise SubTotal
Subtotal
Tax
Total

Sales(SaleID, Date, CustomerID, Name, Address, City, State, Zip, EmployeeID, Name, (AnimalID, Name, Category, Breed, DateOfBirth, Gender, Registration, Color, ListPrice, SalePrice), (ItemID, Description, Category, ListPrice, SalePrice, Quantity))

The Pet Store: Purchase Animals



Purchase Order for Animals

Order# Date Ordered
Date Received

Supplier
Name
Contact
Phone
Address
City, State, ZIPcode

Employee ID
Name
Home Phone
Date Hired

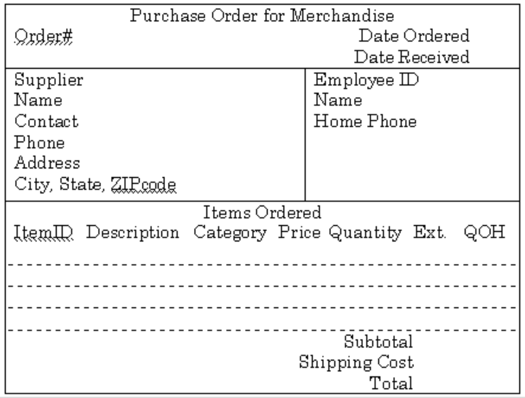
Animal Descriptions

Name	Category	Breed	Gender	Registration	Price

Subtotal
Shipping Cost
Total

AnimalOrder(OrderID, OrderDate, ReceiveDate, SupplierID, Name, Contact, Phone, Address, City, State, Zip, EmployeeID, Name, Phone, DateHired, (AnimalID, Name, Category, Breed, Gender, Registration, Cost), ShippingCost)

The Pet Store: Purchase Merchandise



Purchase Order for Merchandise

Order# Date Ordered
Date Received

Supplier
Name
Contact
Phone
Address
City, State, ZIPcode

Employee ID
Name
Home Phone

Items Ordered

ItemID	Description	Category	Price	Quantity	Ext.	QOH

Subtotal
Shipping Cost
Total

MerchandiseOrder(PONumber, OrderDate, ReceiveDate, SupplierID, Name, Contact, Phone, Address, City, State, Zip, EmployeeID, Name, HomePhone, (ItemID, Description, Category, Price, Quantity, QuantityOnHand), ShippingCost)

Pet Store Normalization

Sale(SaleID, Date, CustomerID, EmployeeID)
SaleAnimal(SaleID, AnimalID, SalePrice)
SaleItem(SaleID, ItemID, SalePrice, Quantity)
Customer(CustomerID, Name, Address, City, State, Zip)
Employee(EmployeeID, Name)
Animal(AnimalID, Name, Category, Breed, DateOfBirth,
Gender, Registration, Color, ListPrice)
Merchandise(ItemID, Description, Category, ListPrice)

AnimalOrder(OrderID, OrderDate, ReceiveDate, SupplierID, EmpID, ShipCost)
AnimalOrderItem(OrderID, AnimalID, Cost)
Supplier(SupplierID, Name, Contact, Phone, Address, City, State, Zip)
Employee(EmployeeID, Name, Phone, DateHired)
Animal(AnimalID, Name, Category, Breed, Gender, Registration, Cost)

MerchandiseOrder(PONumber, OrderDate, ReceiveDate, SID, EmpID, ShipCost)
MerchandiseOrderItem(PONumber, ItemID, Quantity, Cost)
Supplier(SupplierID, Name, Contact, Phone, Address, City, State, Zip)
Employee(EmployeeID, Name, Phone)
Merchandise(ItemID, Description, Category, QuantityOnHand)

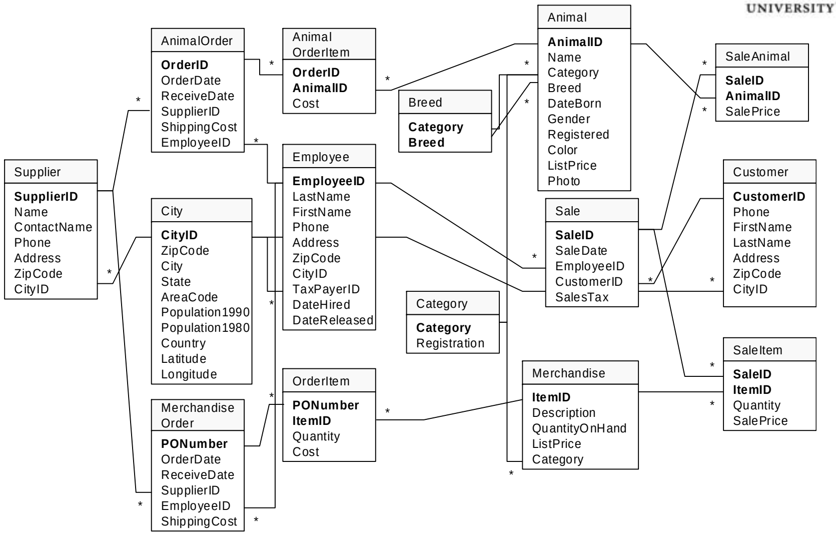
Pet Store View Integration

Sale(SaleID, Date, CustomerID, EmployeeID)
SaleAnimal(SaleID, AnimalID, SalePrice)
SaleItem(SaleID, ItemID, SalePrice, Quantity)
Customer(CustomerID, Name, Address, City, State, Zip)
Employee(EmployeeID, Name, Phone, DateHired)
Animal(AnimalID, Name, Category, Breed, DateOfBirth,
Gender, Registration, Color, ListPrice, Cost)
Merchandise(ItemID, Description, Category, ListPrice, QuantityOnHand)

AnimalOrder(OrderID, OrderDate, ReceiveDate, SupplierID, EmpID, ShipCost)
AnimalOrderItem(OrderID, AnimalID, Cost)
Supplier(SupplierID, Name, Contact, Phone, Address, City, State, Zip)
~~Employee(EmployeeID, Name, Phone, DateHired)~~
~~Animal(AnimalID, Name, Category, Breed, Gender, Registration, Cost)~~

MerchandiseOrder(PONumber, OrderDate, ReceiveDate, SID, EmpID, ShipCost)
MerchandiseOrderItem(PONumber, ItemID, Quantity, Cost)
~~Supplier(SupplierID, Name, Contact, Phone, Address, City, State, Zip)~~
~~Employee(EmployeeID, Name, Phone)~~
~~Merchandise(ItemID, Description, Category, QuantityOnHand)~~

Pet Store Class Diagram



Supplier: **SupplierID**, Name, ContactName, Phone, Address, ZipCode, CityID

AnimalOrder: **OrderID**, OrderDate, ReceiveDate, SupplierID, ShippingCost, EmployeeID

City: **CityID**, ZipCode, City, State, AreaCode, Population1990, Population1980, Country, Latitude, Longitude

Merchandise Order: **PONumber**, OrderDate, ReceiveDate, SupplierID, EmployeeID, ShippingCost

Animal OrderItem: **OrderID**, **AnimalID**, Cost

Employee: **EmployeeID**, LastName, FirstName, Phone, Address, ZipCode, CityID, TaxPayerID, DateHired, DateReleased

OrderItem: **PONumber**, **ItemID**, Quantity, Cost

Breed: **Category**, **Breed**

Category: **Category**, Registration

Animal: **AnimalID**, Name, Category, Breed, DateBorn, Gender, Registered, Color, ListPrice, Photo

Sale: **SaleID**, SaleDate, EmployeeID, CustomerID, SalesTax

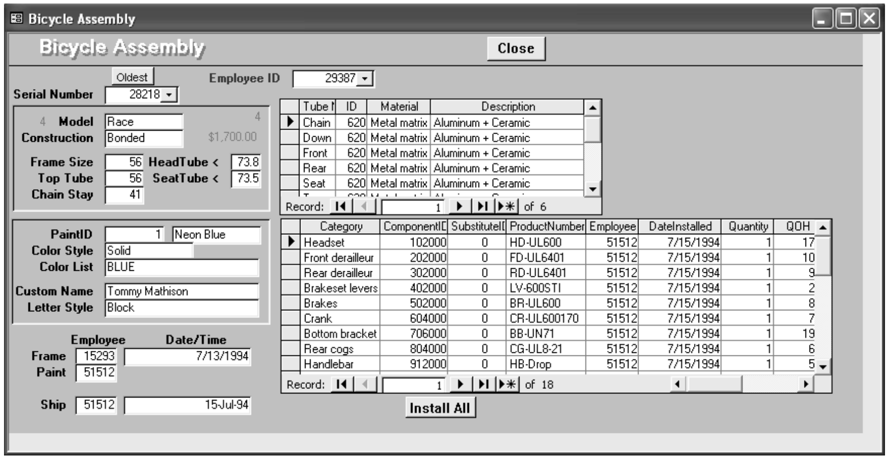
Merchandise: **ItemID**, Description, QuantityOnHand, ListPrice, Category

SaleAnimal: **SaleID**, **AnimalID**, SalePrice

Customer: **CustomerID**, Phone, FirstName, LastName, Address, ZipCode, CityID

SaleItem: **SaleID**, **ItemID**, Quantity, SalePrice

Rolling Thunder Integration Example



Bicycle Assembly

Employee ID 29387

Serial Number 28218

4 Model Race 4
Construction Bonded $1,700.00

Frame Size 56 HeadTube < 73.8
Top Tube 56 SeatTube < 73.5
Chain Stay 41

	Tube N	ID	Material	Description
▶	Chain	620	Metal matrix	Aluminum + Ceramic
	Down	620	Metal matrix	Aluminum + Ceramic
	Front	620	Metal matrix	Aluminum + Ceramic
	Rear	620	Metal matrix	Aluminum + Ceramic
	Seat	620	Metal matrix	Aluminum + Ceramic

Record: 1 of 6

PaintID 1 Neon Blue
Color Style Solid
Color List BLUE

Custom Name Tommy Mathison
Letter Style Block

	Employee	Date/Time
Frame	15293	7/13/1994
Paint	51512	
Ship	51512	15-Jul-94

	Category	ComponentID	SubstituteID	ProductNumber	Employee	DateInstalled	Quantity	QOH
▶	Headset	102000	0	HD-UL600	51512	7/15/1994	1	17
	Front derailleur	202000	0	FD-UL6401	51512	7/15/1994	1	10
	Rear derailleur	302000	0	RD-UL6401	51512	7/15/1994	1	9
	Brakeset levers	402000	0	LV-600STI	51512	7/15/1994	1	2
	Brakes	502000	0	BR-UL600	51512	7/15/1994	1	8
	Crank	604000	0	CR-UL600170	51512	7/15/1994	1	7
	Bottom bracket	706000	0	BB-UN71	51512	7/15/1994	1	19
	Rear cogs	804000	0	CG-UL8-21	51512	7/15/1994	1	6
	Handlebar	912000	0	HB-Drop	51512	7/15/1994	1	5

Record: 1 of 18

Bicycle Assembly form. The main EmployeeID control is not stored directly, but the value is entered in the assembly column when the employee clicks the column.

Initial Tables for Bicycle Assembly

BicycleAssembly(
SerialNumber, Model, Construction, FrameSize, TopTube, ChainStay, HeadTube, SeatTube,
PaintID, PaintColor, ColorStyle, ColorList, CustomName, LetterStyle, EmpFrame, EmpPaint,
BuildDate, ShipDate,
(Tube, TubeType, TubeMaterial, TubeDescription),
(CompCategory, ComponentID, SubstID, ProdNumber, EmpInstall, DateInstall, Quantity, QOH))

Bicycle(SerialNumber, Model, Construction, FrameSize, TopTube, ChainStay, HeadTube, SeatTube, PaintID, ColorStyle, CustomName, LetterStyle, EmpFrame, EmpPaint, BuildDate, ShipDate)

Paint(PaintID, ColorList)

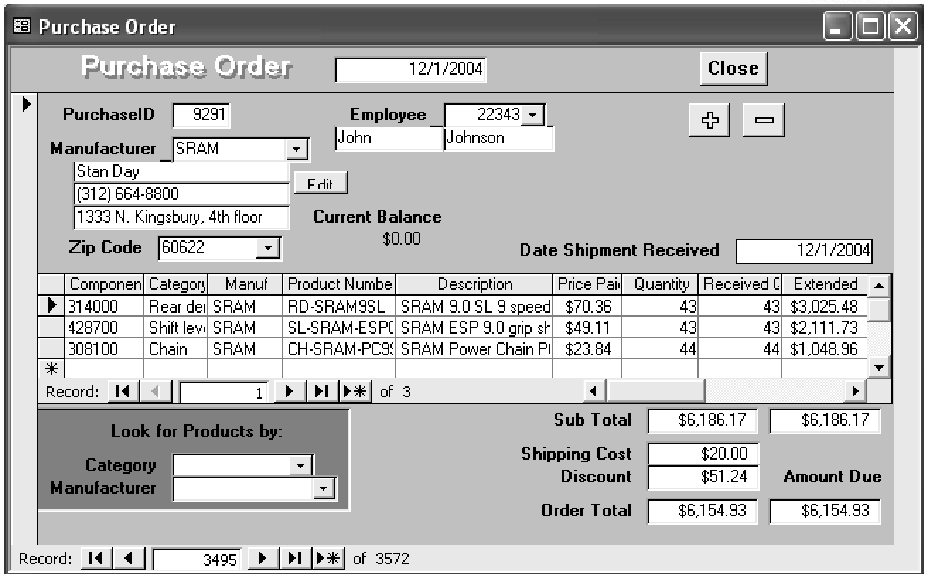
BikeTubes(SerialNumber, TubeID, Quantity)

TubeMaterial(TubeID, Type, Material, Description)

BikeParts(SerialNumber, ComponentID, SubstID, Quantity, DateInstalled, EmpInstalled)

Component(ComponentID, ProdNumber, Category, QOH)

Rolling Thunder: Purchase Order



Purchase Order

Purchase Order 12/1/2004 Close

PurchaseID 9291
Employee 22343
John Johnson
Manufacturer SRAM
Stan Day
(312) 664-8800
1333 N. Kingsbury, 4th floor
Edit
Current Balance $0.00
Zip Code 60622
Date Shipment Received 12/1/2004

Componen	Category	Manuf	Product Numbe	Description	Price Pai	Quantity	Received Q	Extended
314000	Rear der	SRAM	RD-SRAM9SL	SRAM 9.0 SL 9 speed	$70.36	43	43	$3,025.48
428700	Shift leve	SRAM	SL-SRAM-ESP(	SRAM ESP 9.0 grip sh	$49.11	43	43	$2,111.73
308100	Chain	SRAM	CH-SRAM-PC9	SRAM Power Chain Pl	$23.84	44	44	$1,048.96

Record: 1 of 3

Look for Products by:
Category
Manufacturer

Sub Total $6,186.17 $6,186.17
Shipping Cost $20.00
Discount $51.24
Amount Due
Order Total $6,154.93 $6,154.93

Record: 3495 of 3572

RT Purchase Order: Initial Tables

PurchaseOrder(PurchaseID, PODate, EmployeeID, FirstName, LastName, ManufacturerID, MfgName, Address, Phone, CityID, CurrentBalance, ShipReceiveDate, (ComponentID, Category, ManufacturerID, ProductNumber, Description, PricePaid, Quantity, ReceiveQuantity, ExtendedValue, QOH, ExtendedReceived), ShippingCost, Discount)

PurchaseOrder(PurchaseID, PODate, EmployeeID, ManufacturerID, ShipReceiveDate, ShippingCost, Discount)

Employee(EmployeeID, FirstName, LastName)

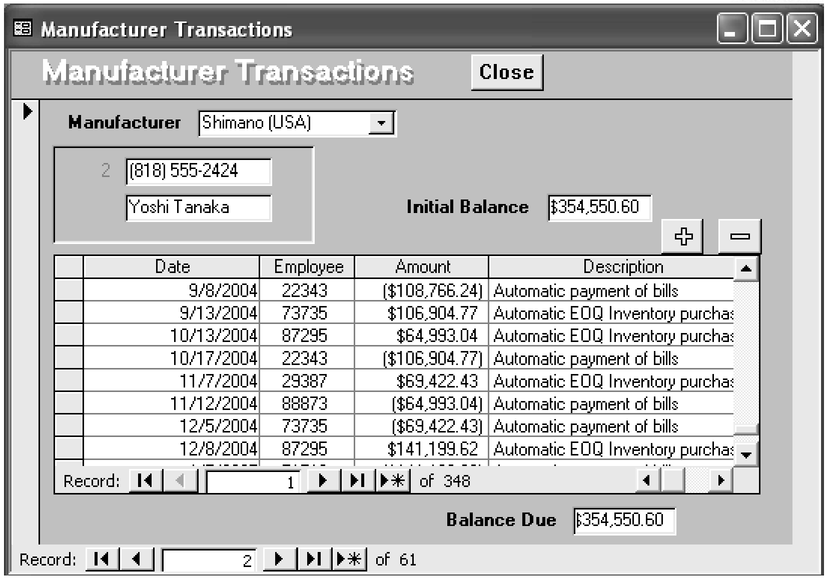
Manufacturer(ManufacturerID, Name, Address, Phone, Address, CityID, CurrentBalance)

City(CityID, Name, ZipCode)

PurchaseItem(PurchaseID, ComponentID, Quantity, PricePaid, ReceivedQuantity)

Component(ComponentID, Category, ManufacturerID, ProductNumber, Description, QOH)

Rolling Thunder: Transactions



Manufacturer Transactions

Manufacturer Transactions Close

Manufacturer Shimano (USA)
2 (818) 555-2424
Yoshi Tanaka
Initial Balance $354,550.60

Date	Employee	Amount	Description
9/8/2004	22343	($108,766.24)	Automatic payment of bills
9/13/2004	73735	$106,904.77	Automatic EOQ Inventory purchas
10/13/2004	87295	$64,993.04	Automatic EOQ Inventory purchas
10/17/2004	22343	($106,904.77)	Automatic payment of bills
11/7/2004	29387	$69,422.43	Automatic EOQ Inventory purchas
11/12/2004	88873	($64,993.04)	Automatic payment of bills
12/5/2004	73735	($69,422.43)	Automatic payment of bills
12/8/2004	87295	$141,199.62	Automatic EOQ Inventory purchas

Record: 1 of 348

Balance Due $354,550.60

Record: 2 of 61

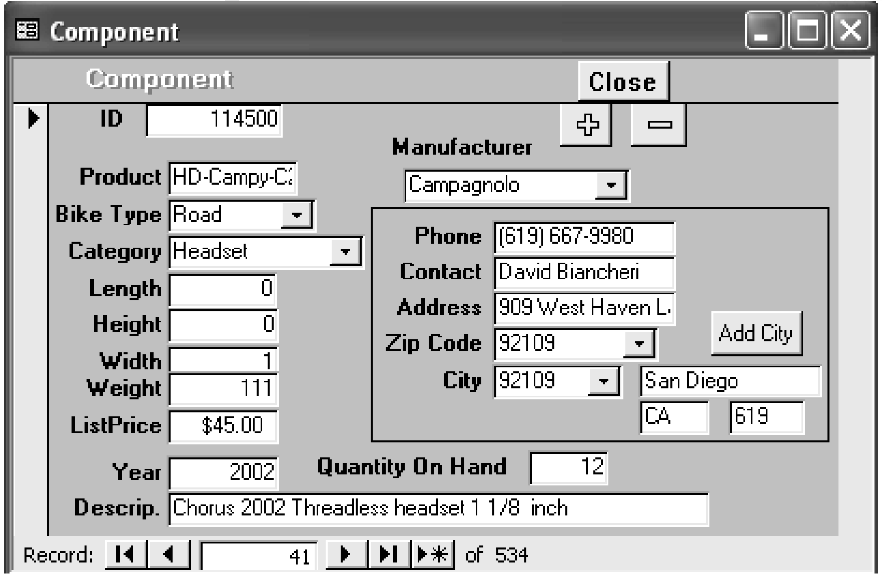
RT Transactions: Initial Tables

ManufacturerTransactions(ManufacturerID, Name, Phone, Contact, BalanceDue, (TransDate, Employee, Amount, Description))

Manufacturer(<u>ManufacturerID</u>, Name, Phone, Contact, BalanceDue)

ManufacturerTransaction(<u>ManufacturerID</u>, <u>TransactionDate</u>, EmployeeID, Amount, Description)

Rolling Thunder: Components



ID	114500
Product	HD-Campy-C
Bike Type	Road
Category	Headset
Length	0
Height	0
Width	1
Weight	111
ListPrice	$45.00
Year	2002
Descrip.	Chorus 2002 Threadless headset 1 1/8 inch
Manufacturer	Campagnolo
Phone	(619) 667-9980
Contact	David Biancheri
Address	909 West Haven L
Zip Code	92109
City	92109 San Diego CA 619
Quantity On Hand	12

Record: 41 of 534

RT Components: Initial Tables

ComponentForm(ComponentID, Product, BikeType, Category, Length, Height, Width, Weight, ListPrice,Description, QOH, ManufacturerID, Name, Phone, Contact, Address, ZipCode, CityID, City, State, AreaCode)

Component(<u>ComponentID</u>, ProductNumber, BikeType, Category, Length, Height, Width,Weight, ListPrice, Description, QOH, ManufacturerID)

Manufacturer(<u>ManufacturerID</u>, Name, Phone, Contact, Address, ZipCode, CityID)

City(<u>CityID</u>, City, State, ZipCode, AreaCode)

RT: Integrating Tables

Duplicate Manufacturer tables:

PO Mfr(<u>ManufacturerID</u>, Name, Address, Phone, CityID, CurrentBalance)
Mfg Mfr(<u>ManufacturerID</u>, Name, Phone, Contact, BalanceDue)
Comp Mfr(<u>ManufacturerID</u>, Name, Phone, Contact, Address, ZipCode, CityID)

Note that each form can lead to duplicate tables. Look for tables with the same keys, but do not expect them to be named exactly alike.
Find all of the data and combine it into one table.

Manufacturer(<u>ManufacturerID</u>, Name, Contact, Address, Phone, Address, CityID, |ZipCode, CurrentBalance)

RT Example: Integrated Tables

Bicycle(SerialNumber, Model, Construction, FrameSize, TopTube, ChainStay, HeadTube, SeatTube, PaintID, ColorStyle, CustomName, LetterStyle, EmpFrame, EmpPaint, BuildDate, ShipDate)

Paint(PaintID, ColorList)

BikeTubes(SerialNumber, TubeID, Quantity)

TubeMaterial(TubeID, Type, Material, Description)

BikeParts(SerialNumber, ComponentID, SubstID, Quantity, DateInstalled, EmpInstalled)

Component(ComponentID, ProductNumber, BikeType, Category, Length, Height, Width, Weight, ListPrice, Description, QOH, ManufacturerID)

PurchaseOrder(PurchaseID, PODate, EmployeeID, ManufacturerID, ShipReceiveDate, ShippingCost, Discount)

PurchaseItem(PurchaseID, ComponentID, Quantity, PricePaid, ReceivedQuantity)

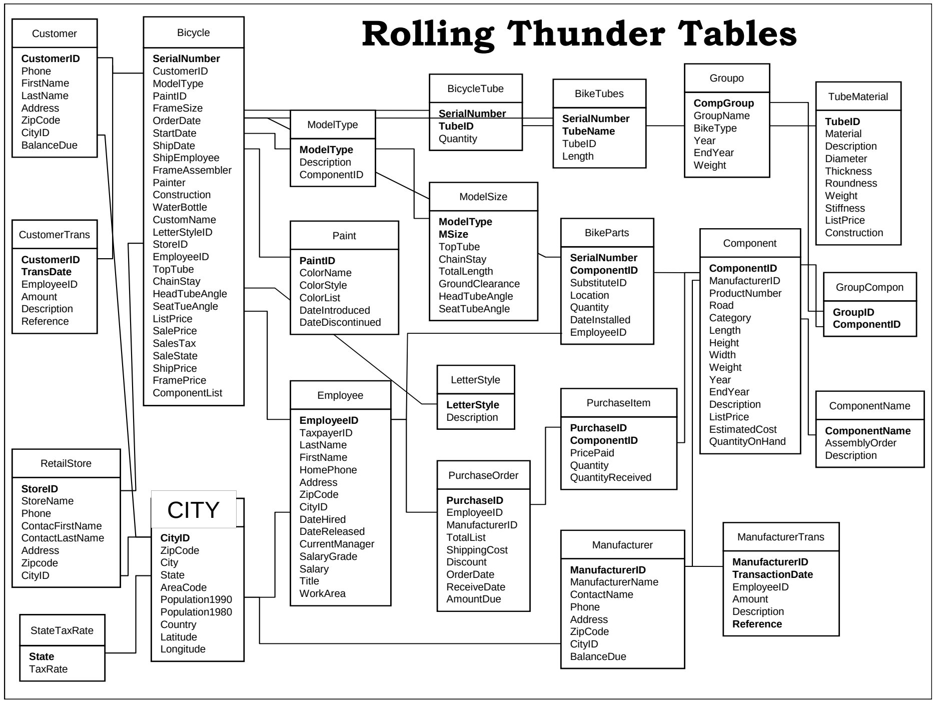
Employee(EmployeeID, FirstName, LastName)

Manufacturer(ManufacturerID, Name, Contact, Address, Phone, CityID, ZipCode, CurrentBalance)

ManufacturerTransaction(ManufacturerID, TransactionDate, EmployeeID, Amount, Description, Reference)

City(CityID, City, State, ZipCode, AreaCode)

Rolling Thunder Tables



View Integration (FEMA Example 1)

Team Roster

Team# Date Formed
Home Base
Response time (days)

Leader
Name Fax Phone
Address, C,S,Z Home phone

Team Members/Crew

ID	**Name**	**Home phone**	**Specialty**	**DoB**	**SSN**	**Salary**

Total Salary

- This first form is kept for each team that can be called on to help in emergencies.

View Integration (FEMA Example 2)

Disaster Name
Local Agency
Political Contact

HQ Location
Commander

On-Site Problem Report

Date Reported Assigned Problem# Severity
Problem Description

Reported By: Specialty Specialty Rating
Verified By: Specialty Specialty Rating

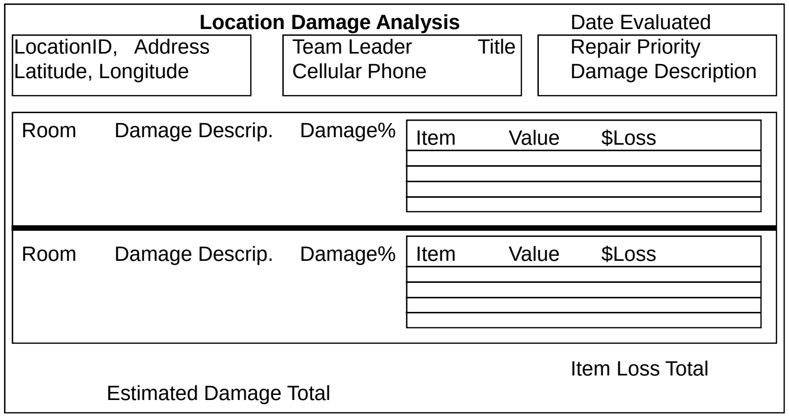
SubProblem Details

Sub Prob#	Category	Description	Action	Est. Cost

Total Est. Cost

- Major problems are reported to HQ to be prioritized and scheduled for correction.

View Integration (FEMA Example 3)



Location Damage Analysis Date Evaluated

LocationID, Address
Latitude, Longitude

Team Leader Title
Cellular Phone

Repair Priority
Damage Description

Room Damage Descrip. Damage%

Item	Value	$Loss

Room Damage Descrip. Damage%

Item	Value	$Loss

Item Loss Total

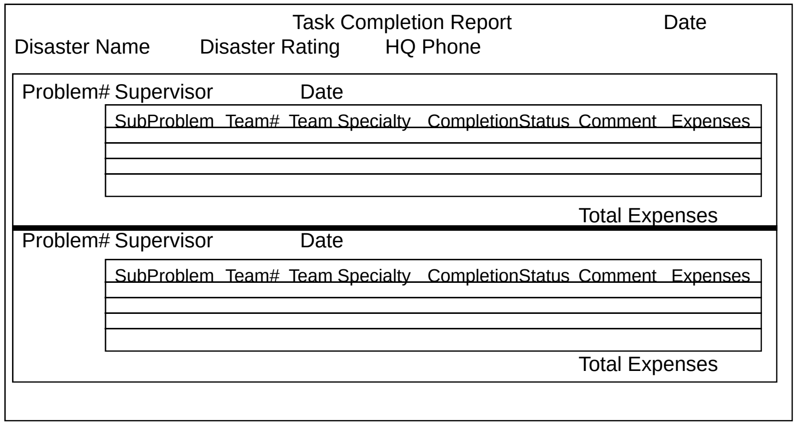
Estimated Damage Total

- **On-site teams examine buildings and file a report on damage at that location.**

View Integration (FEMA Example 3a)

- **Location Analysis(LocationID, MapLatitude, MapLongitude, Date, Address, Damage, PriorityRepair, Leader, LeaderPhone, LeaderTitle, (Room, Description, PercentDamage, (Item, Value, Loss)))**

View Integration (FEMA Example 4)



Task Completion Report Date

Disaster Name Disaster Rating HQ Phone

Problem# Supervisor Date

SubProblem	Team#	Team Specialty	CompletionStatus	Comment	Expenses

Total Expenses

Problem# Supervisor Date

SubProblem	Team#	Team Specialty	CompletionStatus	Comment	Expenses

Total Expenses

- **Teams file task completion reports. If a task is not completed, the percentage accomplished is reported as the completion status.**

View Integration (FEMA Example 4a)

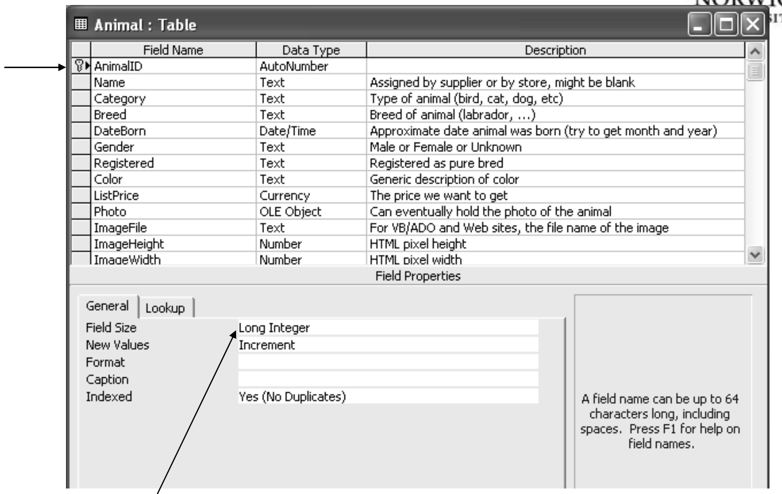
- **TasksCompleted(Date, DisasterName, DisasterRating, HQPhone, (Problem#, Supervisor, (SubProblem, Team#, CompletionStatus, Comments, Expenses))**

DBMS Table Definition

- Enter Tables
  - Columns
  - Keys
  - Data Types
    - ✓ Text
    - ✓ Memo
    - ✓ Number
      - Byte
      - Integer, Long
      - Single, Double
    - ✓ Date/Time
    - ✓ Currency
    - ✓ AutoNumber (Long)
    - ✓ Yes/No
    - ✓ OLE Object
  - Descriptions
- Column Properties
  - Format
  - Input Mask
  - Caption
  - Default
  - Validation Rule
  - Validation Text
  - Required & Zero Length
  - Indexed
- Relationships
  - One-to-One
  - One-to-Many
  - Referential Integrity
  - Cascade Update/Delete
  - Define *before* entering data

Table Definition in Access

Key



Animal : Table

	Field Name	Data Type	Description
🔑	AnimalID	AutoNumber	
	Name	Text	Assigned by supplier or by store, might be blank
	Category	Text	Type of animal (bird, cat, dog, etc)
	Breed	Text	Breed of animal (labrador, ...)
	DateBorn	Date/Time	Approximate date animal was born (try to get month and year)
	Gender	Text	Male or Female or Unknown
	Registered	Text	Registered as pure bred
	Color	Text	Generic description of color
	ListPrice	Currency	The price we want to get
	Photo	OLE Object	Can eventually hold the photo of the animal
	ImageFile	Text	For VB/ADO and Web sites, the file name of the image
	ImageHeight	Number	HTML pixel height
	ImageWidth	Number	HTML pixel width

Field Properties

Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Numeric Subtypes or text length

Data Volume

- Estimate the total size of the database.
  - Current.
  - Future growth.
  - Guide for hardware and software purchases.
- For each table.
  - Use data types to estimate the number of bytes used for each row.
  - Multiply by the estimated number of rows.
- Add the value for each table to get the total size.
- For concatenated keys (and similar tables).
  - OrderItems(<u>O#</u>, <u>Item#</u>, Qty)
  - Hard to "know" the total number of items ordered.
    - ✓ Start with the total number of orders.
    - ✓ Multiply by the average number of items on a typical order.
- Need to know time frame or how long to keep data.
  - Do we store all customer data forever?
  - Do we keep all orders in the active database, or do we migrate older ones?

Data Volume Example

Customer(<u>C#</u>, Name, Address, City, State, Zip)
Row: 4 + 15 + 25 + 20 + 2 + 10 = 76

Order(<u>O#</u>, C#, Odate)
Row: 4 + 4 + 8 = 16

OrderItem(<u>O#</u>, <u>P#</u>, Quantity, SalePrice)
Row: 4 + 4 + 4 + 8 = 20

$$Orders\,in\,3\,yrs = 1000\,Customers * \frac{10\,Orders}{Customer} * 3\,yrs = 30{,}000$$

$$OrderLines = 30{,}000\,Orders * \frac{5\,Lines}{Order} = 150{,}000$$

- Business rules
  - Three year retention.
  - 1000 customers.
  - Average 10 orders per customer per year.
  - Average 5 items per order.

Customer	76 * 1000	76,000
Order	16 * 30,000	480,000
OrderItem	20 * 150,000	3,000,000
Total		3,556,000

Appendices

- ➢ Review same material as textbook
- ➢ Excellent review for you
  - ❑ Explain the slides to each other and to yourself
  - ❑ Show how examples cause problems
  - ❑ Explain how you would solve those problems

Appendix: Formal Definitions: Terms

Formal	Definition	Informal
Relation	**A set of attributes with data that changes over time. Often denoted R.**	**Table**
Attribute	**Characteristic with a real-world domain. Subsets of attributes are multiple columns, often denoted X or Y.**	**Column**
Tuple	**The data values returned for specific attribute sets are often denoted as t[X]**	**Row of data**
Schema	**Collection of tables and constraints/relationships**	
Functional dependency	**X → Y**	**Business rule dependency**

Appendix: Functional Dependency

Derives from a real-world relationship/constraint.

Denoted X → Y for sets of attributes X and Y

Holds when any rows of data that have identical values for X attributes also have identical values for their Y attributes:

If t1[X] = t2[X], then t1[Y] = t2[Y]

X is also known as a determinant if X is non-trivial (not a subset of Y).

Appendix: Keys

Keys are attributes that are ultimately used to identify rows of data.

A key K (sometimes called candidate key) is a set of attributes

(1) With FD K → U where U is all other attributes in the relation

(2) If K’ is a subset of K, then there is no FD K’ → U

A set of key attributes functionally determines all other attributes in the relation, and it is the smallest set of attributes that will do so (there is no smaller subset of K that determines the columns.)

Appendix: First Normal Form

A relation is in first normal form (1NF) if and only if all attributes are atomic.

Atomic attributes are single valued, and cannot be composite, multi-valued or nested relations.

Example:

Customer(CID, Name: First + Last, Phones, Address)

NOT 1NF

CID	Name: First + Last	Phones	Address
111	Joe Jones	111-2223 111-3393 112-4582	123 Main

Appendix: Second Normal Form

A relation is in **second normal form (2NF)** if it is in 1NF and each non-key attribute is fully functionally dependent on the primary key.

K → Ai for each non-key attribute Ai

That is, there is no subset K' such that K' → Ai

Example:

OrderProduct(OrderID, ProductID, Quantity, Description)

NOT 2NF

OrderID	ProductID	Quantity	Description
32	15	1	Blue Hose
32	16	2	Pliers
33	15	1	Blue Hose

Appendix: Transitive Dependency

Given functional dependencies: X → Y and Y → Z, the transitive dependency X → Z must also hold.

Example:

There is an FD between OrderID and CustomerID. Given the OrderID key attribute, you always know the CustomerID.

There is an FD between CustomerID and the other customer data, because CustomerID is the primary key. Given the CustomerID, you always know the corresponding attributes for Name, Phone, and so on.

Consequently, given the OrderID (X), you always know the corresponding customer data by transitivity.

Appendix: Third Normal Form

A relation is in third normal form if and only if it is in 2NF and no non-key attributes are transitively dependent on the primary key.

That is, K → Ai for each attribute, (2NF) and

There is no subset of attributes X such that K → X → Ai

NOT 3NF

Example:

Order(OrderID, OrderDate, CustomerID, Name, Phone)

OrderID	OrderDate	CustomerID	Name	Phone
32	5/5/2004	1	Jones	222-3333
33	5/5/2004	2	Hong	444-8888
34	5/6/2004	1	Jones	222-3333

Appendix: Boyce-Codd Normal Form

A relation is in Boyce-Codd Normal Form (BCNF) if and only if it is in 3NF and every determinant is a candidate key (or K is a superkey).

That is, K → Ai for every attribute, and there is no subset X (key or nonkey) such that X → Ai where X is different from K.

Example: Employees can have many specialties, and many employees can be within a specialty. Employees can have many managers, but a manager can have only one specialty: Mgr → Specialty

EmpSpecMgr(EID, Specialty, ManagerID)

EID	Speciality	ManagerID
32	Drill	1
33	Weld	2
34	Drill	1

FD ManagerID → Specialty is not currently a key.

NOT BCNF

Appendix: Multi-Valued Dependency

A multi-valued dependency (MVD) exists when there are at least three attributes in a relation (A, B, and C; and they could be sets), and one attribute (A) determines the other two (B and C) but the other two are independent of each other.

That is, A →B and A → C but B and C have no FDs

Example:

Employees have many specialties and many tools, but tools and specialties are not directly related.

Appendix: Fourth Normal Form

A relation is in fourth normal form 4NF if and only if it is in BCNF and there are no multi-valued dependencies.

That is, all attributes of R are also functionally dependent on A.

If A → → B, then all attributes of R are also functionally dependent on A: A → Ai for each attribute.

NOT 4NF

Example:

EmpSpecTools(EID, Specialty, ToolID)

EmpSpec(EID, Specialty)

EmpTools(EID, ToolID)

OK: 4NF

HOMEWORK

- ➢ REQUIRED by Sun 20 Feb 2010 at 23:59
  - ❑Study Chapter 3 AGAIN using SQ3R
  - ❑Check the review questions again to be sure they all make sense to you and you can easily answer them
  - ❑For 20 points, submit written answers to Ch 3 problem 7
- ➢ OPTIONAL by Sun 20 Feb 2010 at 23:59
  - ❑For 10 extra points
  - ❑Solve and submit solution to Corner Med problem 14 if you haven't already submitted it for credit.