Data Normalization

Why Normalization?

- Goal: Define tables carefully
  - Save space
  - Minimize redundancy
  - Protect data
  - Define data correctly and the rest is much easier
  - It especially makes it easier to expand database later

Definitions

- Relational database: A collection of tables.
- **Table:** A collection of columns (attributes) describing an entity. Individual objects are stored as rows of data in the table.
- **Property (attribute):** a characteristic or descriptor of a class or entity.
- Every table has a **primary key.**
  - **The smallest set** of columns that uniquely identifies any row
  - Primary keys can span more than one column (**concatenated keys**)

Primary key

Properties

Class: Employee

Rows/Objects

Employee

EmployeeID	TaxpayerID	LastName	FirstName	HomePhone	Address
12512	888-22-5552	Cartom	Abdul	(603) 323-9893	252 South Street
15293	222-55-3737	Venetiaan	Roland	(804) 888-6667	937 Paramaribo Lane
22343	293-87-4343	Johnson	John	(703) 222-9384	234 Main Street
29387	837-36-2933	Stenheim	Susan	(410) 330-9837	8934 W. Maple

Keys

- Primary key
  - Every table (object) must have a primary key
  - **Uniquely identifies** a row
- Concatenated (or **composite**) key
  - **Multiple columns** needed for primary key
  - Identify repeating relationships (**1 : M or M : N**)
- Key columns are <u>underlined</u>
- **First step**
  - Collect user documents
  - **Identify possible keys: unique or repeating relationships**

Notation

Table name

Table columns

Customer(<u>CustomerID</u>, Phone, Name, Address, City, State, ZipCode)

Primary key is underlined

<u>CustomerID</u>	Phone	LastName	FirstName	Address	City	State	Zipcode
1	502-666-7777	Johnson	Martha	125 Main Street	Alvaton	KY	42122
2	502-888-6464	Smith	Jack	873 Elm Street	Bowling Green	KY	42101
3	502-777-7575	Washington	Elroy	95 Easy Street	Smith's Grove	KY	42171
4	502-333-9494	Adams	Samuel	746 Brown Drive	Alvaton	KY	42122
5	502-474-4746	Rabitz	Victor	645 White Avenue	Bowling Green	KY	42102
6	616-373-4746	Steinmetz	Susan	15 Speedway Drive	Portland	TN	37148
7	615-888-4474	Lasater	Les	67 S. Ray Drive	Portland	TN	37148
8	615-452-1162	Jones	Charlie	867 Lakeside Drive	Castalian Springs	TN	37031
9	502-222-4351	Chavez	Juan	673 Industry Blvd.	Caneyville	KY	42721
10	502-444-2512	Rojo	Maria	88 Main Street	Cave City	KY	42127

Identifying Key Columns

Orders

OrderID	Date	Customer
8367	5-5-04	6794
8368	5-6-04	9263

Each order has only one customer. So Customer is **not** part of the key.

OrderItems

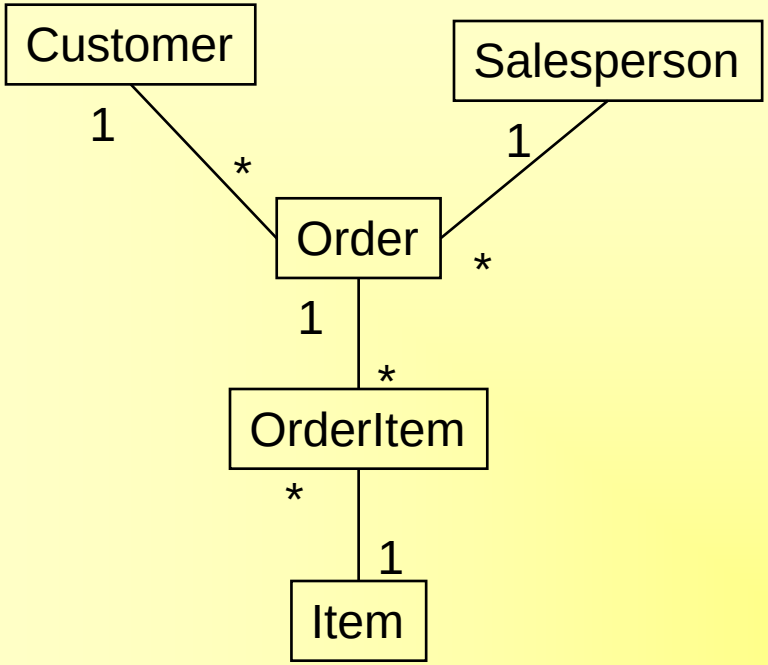
OrderID	Item	Quantity
8367	229	2
8367	253	4
8367	876	1
8368	555	4
8368	229	1

Each order has many items. Each item can appear on many orders. So OrderID and Item are **both** part of the key.

Primary key (Surrogate) Keys

- Real world keys sometimes cause problems in a database.
- **Example: Customer**
  - **Avoid SSN** (privacy and most businesses are not authorized to ask for verification, so you could end up with duplicate values)
- Often best to let the DBMS generate unique values
  - **Access**: AutoNumber
  - **SQL Server:** Identity
  - **Oracle**: Sequences (but require additional programming)
- Drawback: Numbers are not **related to any business data**, so the application needs to hide them and provide other look up mechanisms.

Common Order System



Customer(<u>CustomerID</u>, Name, Address, City, Phone)

Salesperson(<u>EmployeeID</u>, Name, Commission, DateHired)

Order(<u>OrderID</u>, OrderDate, CustomerID, EmployeeID)

OrderItem(<u>OrderID</u>, <u>ItemID</u>, Quantity)

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

Database Normalization Rules

- 1. Each cell in a table contains **atomic (single-valued) data**.
- 2. Each **non-key column** depends on all of the **primary key columns** (not just some of the columns).
- 3. Each **non-key column** depends on **nothing outside** of the key columns.

Repeating Values for Phone Numbers

CustomerID	LastName	FirstName	Phone
15023	Jones	Mary	222-3034 222-4094 223-0984
63478	Sanchez	Miguel	030-9693 403-4094
94552	O'Reilly	Madeline	849-4948 292-3332 139-3831 339-4040
45791	Stein	Marta	294-4421
49004	Brise	Mer	764-5103

Atomic Values for Phone Numbers

CustomerID	LastName	FirstName	Phone	Fax	CellPhone
15023	Jones	Mary	222-3034	222-4094	223-0984
63478	Sanchez	Miguel	030-9693	403-4094	
94552	O’Reilly	Madelline	849-4948	292-3332	139-3831
45791	Stein	Marta	294-4421		
49004	Brise	Mer	764-5103		

Repeating Values for Phone Numbers

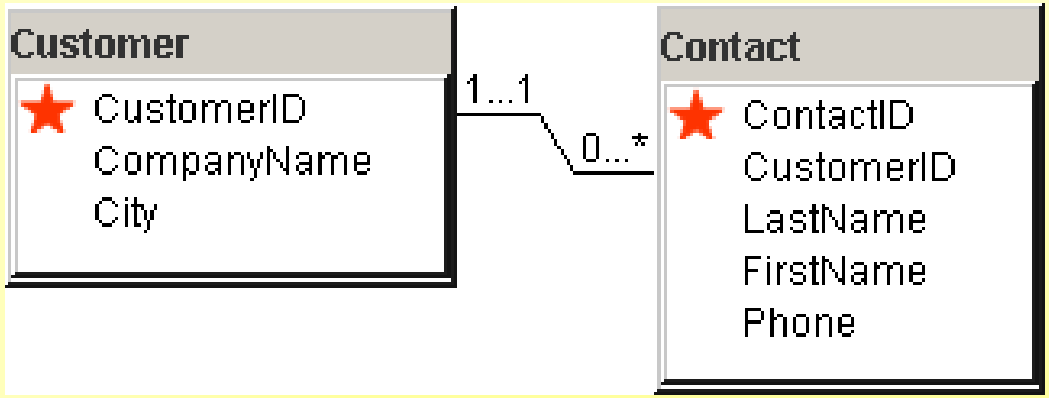
CustomerID	LastName	FirstName
15023	Jones	Mary
63478	Sanchez	Miguel
94552	O'Reilly	Madeline
45791	Stein	Marta
49004	Brise	Mer

CustomerID	PhoneType	Phone
15023	Land	222-3034
15023	Fax	222-4094
15023	Cell	223-0984
63478	Land	030-9693
63478	Fax	403-4094
94552	Land	849-4948
94552	Fax	292-3332
94552	Cell	139-3831
94552	Laptop	339-4040
45791	Land	294-4421
49004	Land	764-5103

Simple Form

Customer ID Company Name
City
Contact LastName, FirstName
Phone

Initial Design



Customer(**<u>CustomerID</u>**, CompanyName, City)

Contact(**<u>ContactID</u>**, CustomerID, LastName, FirstName)

Sample Database for Sales

Sale ID					Date
Customer First Name Last Name Address City, State ZIPCode					
ItemID	**Description**	**List Price**	**Quantity**	**QOH**	**Value**
					Total

Initial Form Evaluation

SaleForm(<u>SaleID</u>, SaleDate, CustomerID, FirstName, LastName,
Address, City, State, ZIPCode,
(<u>ItemID</u>, Description, ListPrice, Quantity, QuantityOnHand))

Identify potential keys
Identify repeating groups

Sale ID					Date
Customer First Name Last Name Address City, State ZIPCode					
ItemID	**Description**	**List Price**	**Quantity**	**QOH**	**Value**
					Total

Initial Objects

Initial Object	Key	Sample Properties
Customer	Assign CustomerID	Name Address Phone
Item	Assign ItemID	Description List Price Quantity On Hand
Sale	Assign SaleID	Sale Date
SaleItems	SaleID + ItemID	Quantity

Initial Form Evaluation

SaleForm(<u>SaleID</u>, SaleDate, CustomerID, FirstName, LastName,
Address, City, State, ZIPCode,
(<u>ItemID</u>, Description, ListPrice, Quantity, QuantityOnHand))

Identify potential keys
Identify repeating groups

Sale ID					Date
Customer First Name Last Name Address City, State ZIPCode					
ItemID	**Description**	**List Price**	**Quantity**	**QOH**	**Value**
					Total

Problems with Repeating Sections

SaleForm(SaleID, SaleDate, CustomerID, FirstName, LastName, Address, City, State, ZIPCode, (ItemID, Description, ListPrice, Quantity, QuantityOnHand))

SaleID	Date	CID	FirstName	LastName	Address	City	State	ZIP	ItemID	Description	ListPrice	Quantity	QOH

Problems with Repeating Sections

SaleForm(SaleID, SaleDate, CustomerID, FirstName, LastName, Address, City, State, ZIPCode,
(ItemID, Description, ListPrice, Quantity, QuantityOnHand))

SaleID	Date	CID	FirstName	LastName	Address	City	State	ZIP	ItemID	Description	ListPrice	Quantity	QOH
11851	7/15	15023	Mary	Jones	111 Elm	Chicago	IL	60601	15	Air Tank	192.00	2	15
									27	Regulator	251.00	1	5
									32	Mask 1557	65.00	1	6
11852	7/15	63478	Miguel	Sanchez	222 Oro	Madrid			15	Air Tank	192.00	4	15
									33	Mask 2020	91.00	1	3
11853	7/16	15023	Mary	Jones	111 Elm	Chicago	IL	60601	41	Snorkel 71	44.00	2	15
									75	Wet suit-S	215.00	1	3
11854	7/17	94552	Madeline	O'Reilly	333 Tam	Dublin			75	Wet suit-S	215.00	2	3
									32	Mask 1557	65.00	1	6
									57	Snorkel 95	83.00	1	17

Database Normalization Rules

- 1. Each cell in a table contains **atomic (single-valued) data**.
- 2. Each **non-key column** depends on all of the **primary key columns** (not just some of the columns).
- 3. Each **non-key column** depends on **nothing outside** of the key columns.

Problems with Repeating Sections

SaleForm(SaleID, SaleDate, CustomerID, FirstName, LastName, Address, City, State, ZIPCode,
(ItemID, Description, ListPrice, Quantity, QuantityOnHand))

Repeating section

Duplication

Not atomic

SaleID	Date	CID	FirstName	LastName	Address	City	State	ZIP	ItemID	Description	ListPrice	Quantity	QOH
11851	7/15	15023	Mary	Jones	111 Elm	Chicago	IL	60601	15	Air Tank	192.00	2	15
									27	Regulator	251.00	1	5
									32	Mask 1557	65.00	1	6
11852	7/15	63478	Miguel	Sanchez	222 Oro	Madrid			15	Air Tank	192.00	4	15
									33	Mask 2020	91.00	1	3
11853	7/16	15023	Mary	Jones	111 Elm	Chicago	IL	60601	41	Snorkel 71	44.00	2	15
									75	Wet suit-S	215.00	1	3
11854	7/17	94552	Madeline	O'Reilly	333 Tam	Dublin			75	Wet suit-S	215.00	2	3
									32	Mask 1557	65.00	1	6
									57	Snorkel 95	83.00	1	17

First Normal Form Definition

Eliminating Repeated Groups

SaleForm(SaleID, SaleDate, CustomerID, FirstName, LastName, Address, City, State, ZIPCode,
(ItemID, Description, ListPrice, Quantity, QuantityOnHand))

First Normal Form Definition

Eliminating Repeated Groups

SaleForm(SaleID, SaleDate, CustomerID, FirstName, LastName, Address, City, State, ZIPCode,
(ItemID, Description, ListPrice, Quantity, QuantityOnHand))

First Normal Form Definition

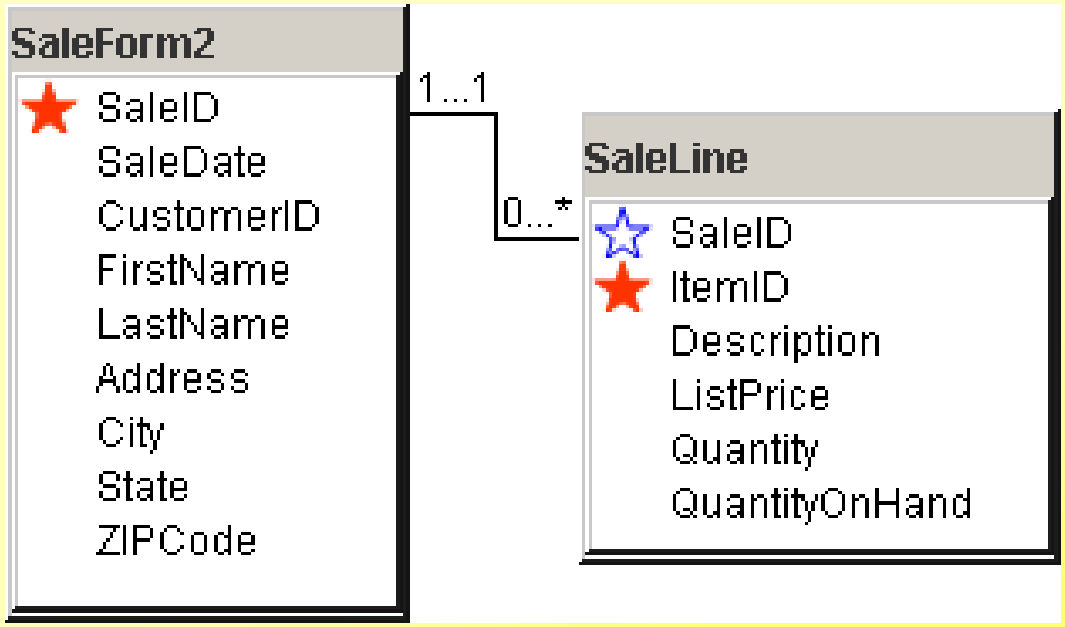
Eliminating Repeated Groups

SaleForm(SaleID, SaleDate, CustomerID, FirstName, LastName, Address, City, State, ZIPCode,
(ItemID, Description, ListPrice, Quantity, QuantityOnHand))

SaleForm2(SaleID, SaleDate, CustomerID, FirstName, LastName, Address, City, State, ZIPCode)

SaleLine(SaleID, ItemID, Description, ListPrice, Quantity, QuantityOnHand)

Current Design



Multiple Repeating: Independent Groups

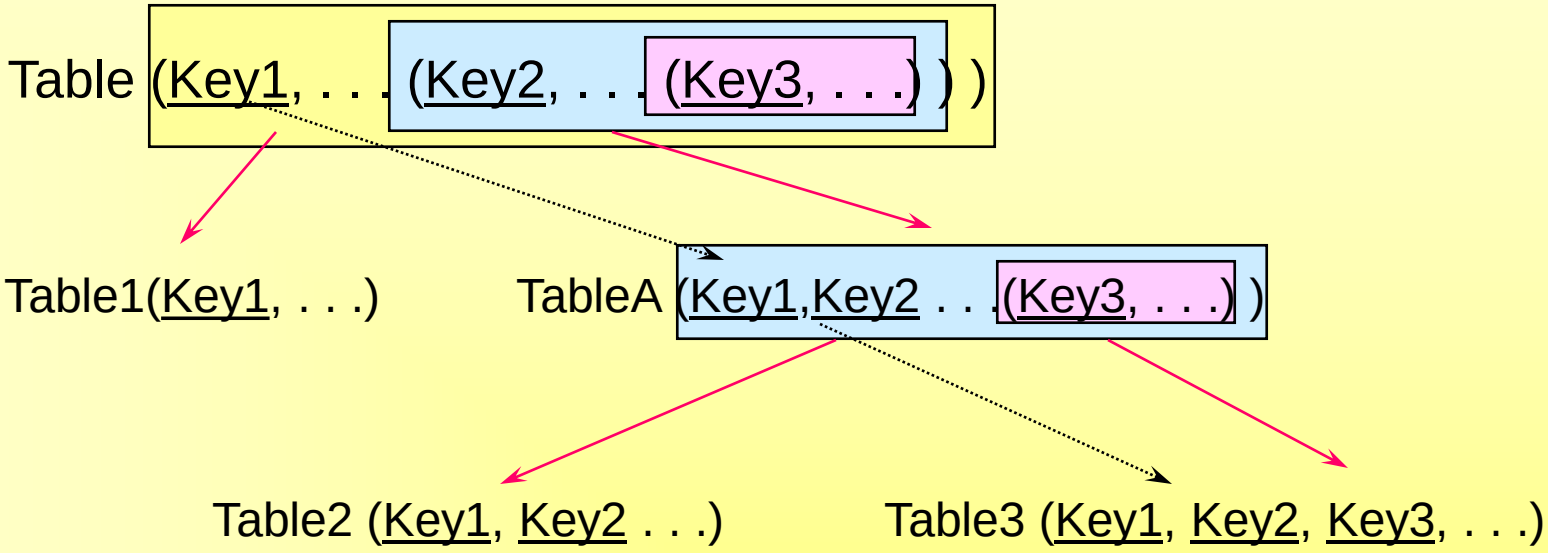
FormA(Key1, Simple Columns, (Group1, A, B, C), (Group2, X, Y))

MainTable(Key1, Simple Columns)

Group1(Key1, Group1, A, B, C)

Group2(Key1, Group2, X, Y)

Nested Repeating Sections



- Nested: Table (Key1, aaa. . . (Key2, bbb. . . (Key3, ccc. . .)))
- First Normal Form (1NF)
  - Table1(Key1, aaa . . .)
  - Table2(Key1, Key2, bbb . .)
  - Table3(Key1, Key2, Key3, ccc. . .)

First Normal Form Definition

Eliminating Repeated Groups

SaleForm(SaleID, SaleDate, CustomerID, FirstName, LastName, Address, City, State, ZIPCode,
(ItemID, Description, ListPrice, Quantity, QuantityOnHand))

SaleForm2(SaleID, SaleDate, CustomerID, FirstName, LastName, Address, City, State, ZIPCode)

SaleLine(SaleID, ItemID, Description, ListPrice, Quantity, QuantityOnHand)

First Normal Form Problems (Data)

SaleLine(SaleID, ItemID, Description, ListPrice, Quantity, QuantityOnHand)

SaleID	ItemID	Description	ListPrice	Quantity	QOH
11851	15	Air Tank	192.00	2	15
11851	27	Regulator	251.00	1	5
11851	32	Mask 1557	65.00	1	6
11852	15	Air Tank	192.00	4	15
11852	33	Mask 2020	91.00	1	3
11853	41	Snorkel 71	44.00	2	15
11853	75	West suit-S	215.00	1	3
11854	75	Wet suit-S	215.00	2	3
11854	32	Mask 1557	65.00	1	6
11854	57	Snorkel 95	83.00	1	17

Database Normalization Rules

- 1. Each cell in a table contains **atomic (single-valued) data**.
- 2. Each **non-key column** depends on all of the **primary key columns** (not just some of the columns).
- 3. Each **non-key column** depends on **nothing outside** of the key columns.

First Normal Form Problems (Data)

SaleLine(**SaleID, ItemID**, Description, ListPrice, Quantity, QuantityOnHand)

Duplication for columns that depend only on **ItemID**

SaleID	ItemID	Description	ListPrice	Quantity	QOH
11851	15	Air Tank	192.00	2	15
11851	27	Regulator	251.00	1	5
11851	32	Mask 1557	65.00	1	6
11852	15	Air Tank	192.00	4	15
11852	33	Mask 2020	91.00	1	3
11853	41	Snorkel 71	44.00	2	15
11853	75	West suit-S	215.00	1	3
11854	75	Wet suit-S	215.00	2	3
11854	32	Mask 1557	65.00	1	6
11854	57	Snorkel 95	83.00	1	17

Second Normal Form Definition

Eliminating Redundant Data

Depends on both SaleID and ItemID

SaleLine(SaleID, ItemID, Description, ListPrice, Quantity, QuantityOnHand)

Depend only on ItemID

- Each non-key column must depend on the **entire** key.
  - Only applies to **concatenated keys**
  - Some columns only depend on **part of the key**
  - Split those into a new table.
- Dependence (definition)
  - If given a value for the key you always know the value of the property in question, then that property is said to depend on the key.
  - If you change part of a key and the questionable property does not change, then the table is **not** in 2NF.

Second Normal Form Example

SaleLine(<u>SaleID</u>, <u>ItemID</u>, Description, ListPrice, Quantity, QuantityOnHand)

SaleItems(<u>SaleID</u>, <u>ItemID</u>, Quantity)

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

Second Normal Form Example (Data)

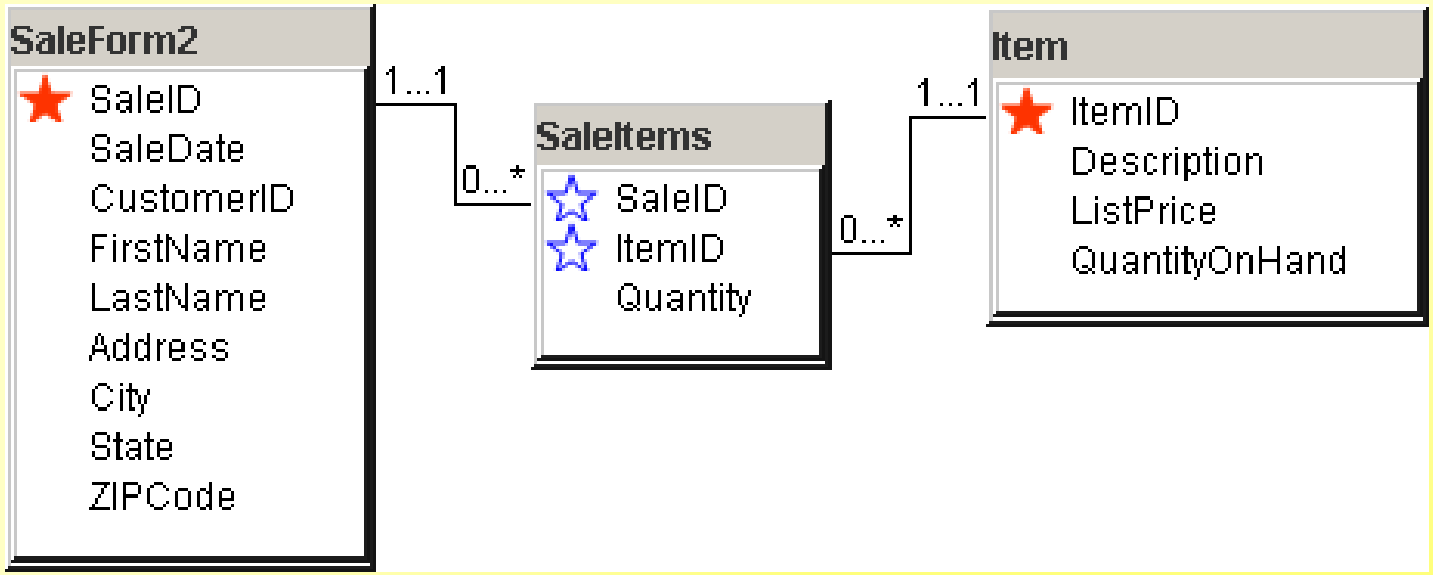
SaleItems(SaleID, ItemID, Quantity)

SaleID	ItemID	Quantity
11851	15	2
11851	27	1
11851	32	1
11852	15	4
11852	33	1
11853	41	2
11853	75	1
11854	75	2
11854	32	1
11854	57	1

ItemID	Description	ListPrice	QOH
15	Air Tank	192.00	15
27	Regulator	251.00	5
32	Mask 1557	65.00	6
33	Mask 2020	91.00	3
41	Snorkel 71	44.00	15
57	Snorkel 95	83.00	17
75	Wet suit-S	215.00	3
77	Wet suit-M	215.00	7

Item(ItemID, Description, ListPrice, QuantityOnHand)

Second Normal Form in DB Design



Second Normal Form Problems (Data)

SaleForm2(<u>SaleID</u>, SaleDate, CustomerID, FirstName, LastName, Address, City, State, ZIPCode)

<u>SaleID</u>	Date	CustomerID	FirstName	LastName	Address	City	State	ZIP
11851	7/15	15023	Mary	Jones	111 Elm	Chicago	IL	60601
11852	7/15	63478	Miguel	Sanchez	222 Oro	Madrid		
11853	7/16	15023	Mary	Jones	111 Elm	Chicago	IL	60601
11854	7/17	94552	Madeline	O'Reilly	333 Tam	Dublin		

Database Normalization Rules

- 1. Each cell in a table contains **atomic (single-valued) data**.
- 2. Each **non-key column** depends on all of the **primary key columns** (not just some of the columns).
- 3. Each **non-key column** depends on **nothing outside** of the key columns.

Second Normal Form Problems (Data)

SaleForm2(<u>SaleID</u>, SaleDate, CustomerID, FirstName, LastName, Address, City, State, ZIPCode)

<u>SaleID</u>	Date	CustomerID	FirstName	LastName	Address	City	State	ZIP
11851	7/15	15023	Mary	Jones	111 Elm	Chicago	IL	60601
11852	7/15	63478	Miguel	Sanchez	222 Oro	Madrid		
11853	7/16	15023	Mary	Jones	111 Elm	Chicago	IL	60601
11854	7/17	94552	Madeline	O'Reilly	333 Tam	Dublin		

Duplication

Third Normal Form Definition

Eliminating Columns not Dependant on Keys

Depend on SaleID

SaleForm2(SaleID, SaleDate, CustomerID, FirstName, LastName, Address, City, State, ZIPCode)

Depend on CustomerID

Third Normal Form Example

SaleForm2(SaleID, SaleDate, CustomerID, FirstName, LastName, Address, City, State, ZIPCode)

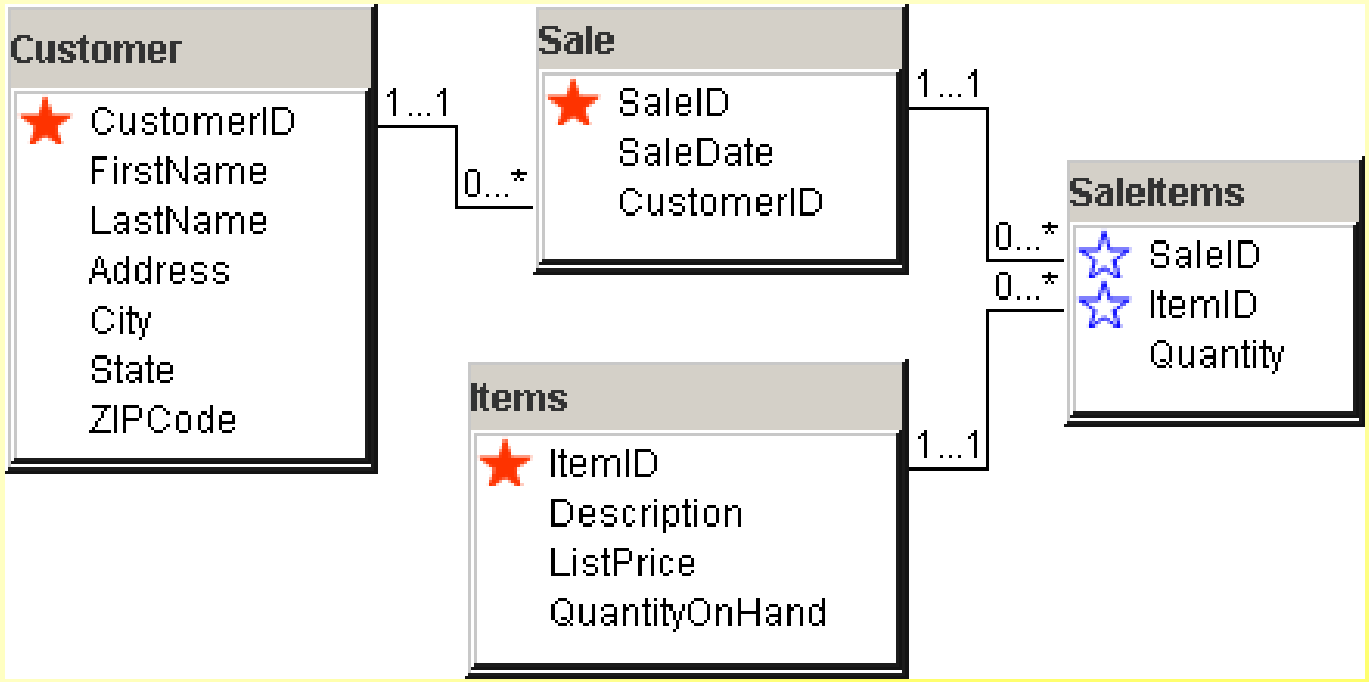
Sale(SaleID, SaleDate, CustomerID)

SaleID	Date	CustomerID
11851	7/15	15023
11852	7/15	63478
11853	7/16	15023
11854	7/17	94552

Customer(CustomerID, FirstName, LastName, Address, City, State, ZIPCode)

CustomerID	FirstName	LastName	Address	City	State	ZIP
15023	Mary	Jones	111 Elm	Chicago	IL	60601
63478	Miguel	Sanchez	222 Oro	Madrid		
94552	Madeline	O'Reilly	333 Tam	Dublin		

Third Normal Form Tables



Third Normal Form Tables

Customer(CustomerID, FirstName, LastName, Address, City, State, ZIPCode)

Sale(SaleID, SaleDate, CustomerID)

SaleItems(SaleID, ItemID, Quantity)

Item(ItemID, Description, ListPrice, QuantityOnHand)

3NF Rules/Procedure

- Split out repeating sections
  - Be sure to include a key from the parent section in the new piece so the two parts can be recombined.
- Verify that the keys are correct
  - Is each row uniquely identified by the primary key?
  - Are one-to-many and many-to-many relationships correct?
  - Check “many” for keyed columns and “one” for non-key columns.
- **Make sure that each non-key column depends on the whole key and nothing but the key.**
  - No hidden dependencies.

Fourth Normal Form (Keys)
Isolates Independent-Multiple-Relationships

EmployeeTasks(EID, Specialty, ToolID)

- Business rules.
- Each employee has many specialties.
- Each employee has many tools.
- Tools and specialties are unrelated.

EmployeeSpecialty(EID, Specialty)
EmployeeTools(EID, ToolID)