

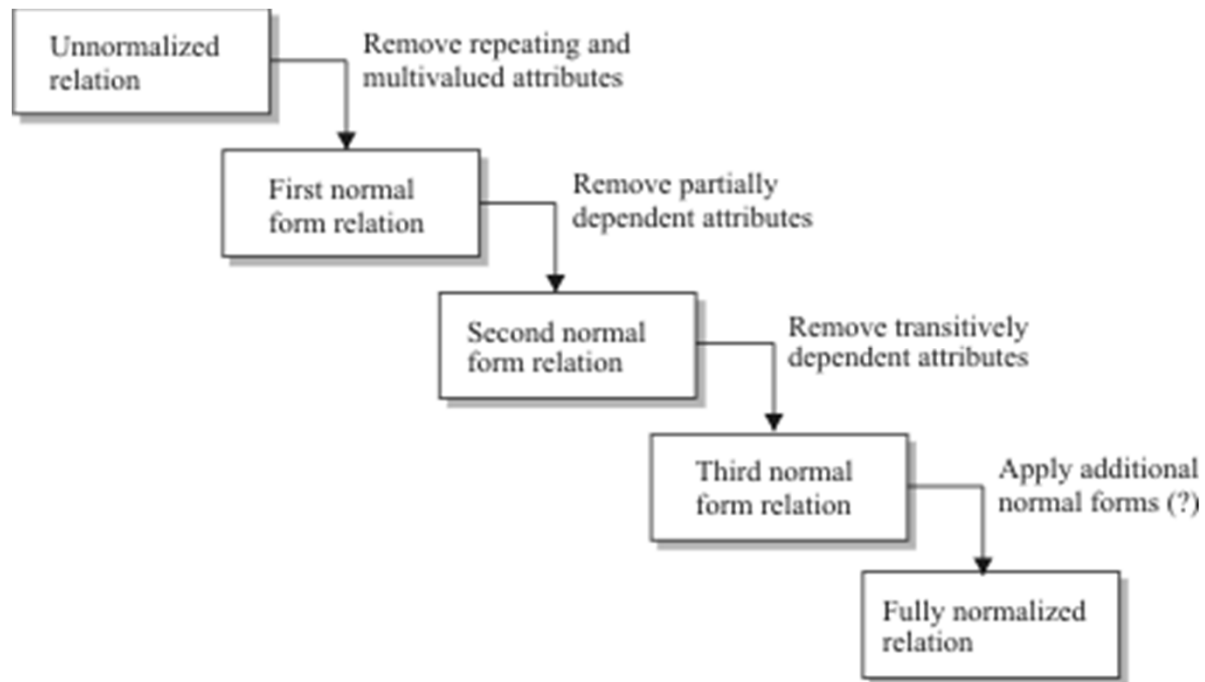
Logical Database Design Using Normalization

Chapter 6

What Is Normalization?

- A technique for producing a set of *relations* that possess a certain set of properties
- Removes issues associated with inserts, updates and deletes
- Process seems simple on the surface, but requires considerable practice to master it

The Normalization Process



The Need for Normalization

- **Insert Anomaly:** a situation in which you cannot insert a new tuple (row) because of an artificial dependency on another relation (entity).
- **Delete Anomaly:** a situation wherein a deletion of data about one particular entity causes unintended loss of data that characterizes another entity.
- **Update Anomaly:** A situation in which an update of a single data value requires multiple tuples (rows) of data to be updated.

Sample User View

Acme Industries INVOICE

Customer Number: 1454837
Customer: W. Coyote
General Delivery
Falling Rocks, AZ 84211
(599) 555-9345

Terms: Net 30
Ship Via: USPS

Order Date: 07/31/2009

<u>Product No.</u>	<u>Description</u>	<u>Quantity</u>	<u>Unit Price</u>	<u>Extended Amount</u>
SPR-2290	Super Strength Springs	2	24.00	\$ 48.00
STR-67	Foot Straps, Leather	2	2.50	\$ 5.00
HLM-45	Deluxe Crash Helmet	1	67.88	\$ 67.88
SFR-1	Rocket, Solid Fuel	1	128,200.40	\$ 128,200.40
ELT-1	Emergency Location Transmitter	1	79.88	** FREE GIFT **
ORDER TOTAL:		7		\$ 128,321.28

User View Represented in Tabular Forms

Represented with multi-valued columns

Customer Number	Customer Name	Customer Address	Cust. City	Cust. State	Cust. ZIP	Customer Phone	Terms	Ship Via	Order Date	Product Number	Description	Quantity	Unit Price	Extended Amount
145837	W. Coyote	General Delivery	Falling Rocks	AZ	84211	599-555-9345	Net 30	USPS	07/31/2009	SPR-2290	Super Strength Springs	2	24.00	\$48.00
										STR-67	Foot Straps, Leather	2	2.50	\$5.00
										HLM-45	Deluxe Crash Helmet	1	67.88	\$67.88
										SFR-1	Rocket, Solid Fuel	1	128,200.40	\$128,200.40
										ELT-1	Emergency Location Transmitter	1	79.88	\$0.00
ORDER TOTAL											7		\$128,321.28	

Represented with multi-valued data moved to different rows

Customer Number	Customer Name	Customer Address	Cust. City	Cust. State	Cust. ZIP	Customer Phone	Terms	Ship Via	Order Date	Product Number	Description	Quantity	Unit Price	Extended Amount
145837	W. Coyote	General Delivery	Falling Rocks	AZ	84211	599-555-9345	Net 30	USPS	07/31/2009	SPR-2290	Super Strength Springs	2	24.00	\$48.00
145837	W. Coyote	General Delivery	Falling Rocks	AZ	84211	599-555-9345	Net 30	USPS	07/31/2009	STR-67	Foot Straps, Leather	2	2.50	\$5.00
145837	W. Coyote	General Delivery	Falling Rocks	AZ	84211	599-555-9345	Net 30	USPS	07/31/2009	HLM-45	Deluxe Crash Helmet	1	67.88	\$67.88
145837	W. Coyote	General Delivery	Falling Rocks	AZ	84211	599-555-9345	Net 30	USPS	07/31/2009	SFR-1	Rocket, Solid Fuel	1	128,200.40	\$128,200.40
145837	W. Coyote	General Delivery	Falling Rocks	AZ	84211	599-555-9345	Net 30	USPS	07/31/2009	ELT-1	Emergency Location Transmitter	1	79.88	\$0.00
ORDER TOTAL											7		\$128,321.28	

Choosing a Primary Key

- A unique identifier (primary key) is essential to the normalization process
- Composed of one or more attributes
- Choices (in order of preference):
 - Obvious (there is only one candidate key)
 - The candidate key least likely to have its value changed
 - The simplest candidate key
 - The shortest candidate key
- An artificial (surrogate) key is always an alternative

First Normal Form

- A relation is in first normal form (1NF) when it contains no multi-valued attributes and no repeating attributes or attribute groups
- Move noncompliant attributes to new relations
- DO NOT combine unrelated repeating attributes into common relations (this causes other normalization problems)

Unnormalized Relation

INVOICE:

INVOICE NUMBER, customer number, customer name, street address, city, state, zip, telephone, terms, ship via, order date, (product number, description, quantity, unit price, extended amount), total order amount

Transformed to 1NF

INVOICE:

INVOICE NUMBER, customer number, customer name, street address, city, state, zip, telephone, terms, ship via, order date, total order amount

INVOICE LINE ITEM:

INVOICE NUMBER, PRODUCT NUMBER, description, quantity, unit price, extended amount

Functional Dependency

- **Functional Dependency:** “B” is functionally dependent on “A” if for every value of “A”, there is exactly one value of “B”
 - Mathematically, we say that “A” **determines** “B”
 - Physically, we might say that “A” is a unique identifier for “B”
- **Full Functional Dependency:** “B” is functionally dependent on “A” but not on any subset of “A”
- **Transitive Dependency:** If “B” and “C” are both dependent on “A”, but “C” is also dependent on “B”, we say that “C” is **transitively dependent** on “B”

Second Normal Form: Eliminating Functional Dependencies

- To be in second normal form (2NF), a relation must:
 - Be in first normal form
 - Have all attributes functionally dependent on the *entire* primary key
- Relations with keys composed of a single attribute cannot possibly have 2NF compliance issues
- To achieve 2NF, move any partially dependent attributes to relations where they depend on the entire key

Relation in 1NF

INVOICE LINE ITEM:

INVOICE NUMBER, PRODUCT NUMBER, description, quantity, unit price, extended amount

(INVOICE is already in 2NF because it has a single attribute for a primary key)

Transformed to 2NF

INVOICE:

INVOICE NUMBER, PRODUCT NUMBER, quantity, sale price, extended amount

PRODUCT:

PRODUCT NUMBER, description, unit price

Third Normal Form: Eliminating Transitive Dependencies

- Transitive dependency occurs when a non-key attribute depends on another non-key attributes
- A relation is in third normal form (3NF) if:
 - It is in 2NF
 - It contains no transitive dependencies
- NOTE: Candidate keys are exempt from this rule (primary keys and alternate keys naturally depend on each other)
- Easily calculated attributes are considered to be transitively dependent

Relations in 2NF

INVOICE: INVOICE NUMBER, customer number, customer name, street address, city, state, zip, telephone, terms, ship via, order date, total order amount

INVOICE LINE ITEM: INVOICE NUMBER, PRODUCT NUMBER, quantity, sale price, extended amount

PRODUCT: PRODUCT NUMBER, description, unit price

Transformed to 3NF

INVOICE: INVOICE NUMBER, customer number, terms, ship via, order date

INVOICE LINE ITEM: INVOICE NUMBER, PRODUCT NUMBER, quantity, sale price

PRODUCT: PRODUCT NUMBER, description, unit price

CUSTOMER: CUSTOMER NUMBER, name, address, city, state, zip, telephone

Denormalization

- Considered a last resort for data structures which support transaction processing applications
- Don't give into pressure to denormalize to solve *perceived* performance problems
- Options include:
 - Recombining relations that were split to satisfy normalization rules
 - Storing redundant data in physical tables
 - Storing summarized data in physical tables