Introduction to Relational Databases

TRAINSIGNAL

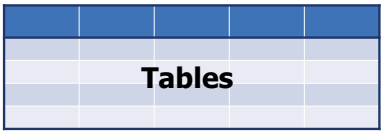
THE GLOBAL LEADER IN PROFESSIONAL COMPUTER TRAINING

Database Concepts

Databases–

A collection of objects

Database Objects



- Store data
- Core component to all databases

Database Objects

Other objects work with table data

- Views are virtual tables
  - Simplify structure
- Stored procedures are commands that can be executed
  - Make changes
  - Return data
- Functions are commands that can be executed
  - Return data

Table Concepts

Tables

Columns

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

Tables

Columns

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

TRAINSIGNAL

Tables

Columns

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

TRAINSIGNAL

Tables

Columns

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

Tables

Columns

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

Tables

Rows

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

Tables

Rows

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

Tables

Rows

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

Tables

Rows

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

Tables

Rows

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

Tables

Rows

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

Tables

Fields

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

Tables

Fields

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

Tables

Fields

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

Tables

Fields

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

Tables

Fields

CustomerID	Last Name	First Name	JoinDate	Region
42	Harrison	Christopher	1/4/2012	Northeast
43	Thomas	Dave	1/6/2012	Northeast
44	Smith	Karin	1/9/2012	Southwest
45	Jacobs	Susan	1/12/2012	West
46	Monroe	William	1/13/2012	Southwest
47	Williams	Timothy	1/20/2012	Midwest

Constraints–

Ensure valid values

Triggers–

Execute code in response to modifications

Primary Key–

Unique values;

Required for each row;

Designed to uniquely identify each row

Foreign Key–

Used to create relationships between tables;

Refers to the column pointing to the primary key column in the other table

Table Relationships

Relationship Concepts

- Divide data into multiple tables
- Faster updates
- More flexible reporting and querying
- Aids in data integrity

Most queries require data from multiple tables

- Accomplished through joins

Table Relationship Types

One-to-One

- One row refers to one row
- Used to divide data when details aren't frequently needed
- Not commonly implemented

Table Relationship Types

One-to-Many

- One row refers to many rows
- Parent child relationships
  - One customer can have many orders

Table Relationship Types

Many-to-Many

- Many rows refer to many rows
  - One product can be ordered many times
  - One order can have many products
- Implemented through two one-to-many relationships
  - Middle table sometimes referred to as a "join" table

Table Design

Design Concepts

A database is only as useful as its design
There are frequently no true right answers
- Some implementations are more useful than others

Requires forethought and planning
Requires experience
Typically done by database administrator or developer
Normalization
- Process of designing tables
- Levels or forms
  - 5 forms

Design Scenario

Need to store customer orders
A customer can place multiple orders
An order can contain multiple products
Products can be ordered multiple times

First Pass

Orders					
OrderDate	Customer	Product1	Quantity1	Product2	Quantity2
4/6/2012	Christopher	Widget	5	Mouse	3
4/8/2012	Christopher	Mouse	6		
4/10/2012	Dave	Widget	2	Mug	7
4/12/2012	Karin	Mug	3	Shirt	9

Problem:
How many columns do we add?

TRAINSIGNAL

First Pass Problems

How many columns do we create?

Normalization Rules

- First normal form
  - Table has no repeating groups
  - Create a new table to store products

Resolution

- Create a new table for products
- Point products at orders for main order information

TRAINSIGNAL

Second Pass

Orders		
OrderID	OrderDate	Customer
1	4/6/2012	Christopher
2	4/8/2012	Christopher
3	4/10/2012	Dave
4	4/12/2012	Karin

***Orders** table changes*
Added OrderID as primary key
Migrated product data to new table

Products		
OrderID	Name	Quantity
1	Widget	5
1	Mouse	3
2	Mouse	6
3	Widget	2
3	Mug	7
4	Mug	3
4	Shirt	9

***Products** table notes*
OrderID is foreign key
Points to OrderID in Orders

Problem:
Duplicated row data

TRAINSIGNAL

Second Pass Problems

Duplicated row data

Normalization Rules

- Second normal form
  - No non-prime attribute is dependent on a subset of any key
  - Or, put another way...
    - If a column could uniquely identify an item (Product Name, for instance), don't duplicate it

Second Pass Resolution

Create new customer table

- One-to-many relationship
  - One customer to many orders

Create new order details table

- Many-to-many relationship
  - Product can be ordered many times
  - Order can have many products

Third Pass

Orders

OrderID	CustomerID	OrderDate
1	1	4/6/2012
2	1	4/8/2012
3	2	4/10/2012
4	3	4/12/2012

***Orders** table changes*
Migrated customer data to new table
Added CustomerID colulmn as foreign key\
CustomerID points to Customers table

Customers

CustomerID	Name	JoinDate
1	Christopher	3/5/2012
2	Dave	3/31/2012
3	Karin	4/6/2012

***Customers** table notes*
CustomerID is primary key

Third Pass

Orders

OrderID	CustomerID	OrderDate
1	1	4/6/2012
2	1	4/8/2012
3	2	4/10/2012
4	3	4/12/2012

Products

ProductID	Name	ListPrice
1	Widget	$42.00
2	Mouse	$16.00
3	Mug	$8.00
4	Shirt	$24.00

OrderDetails

OrderID	ProductID	Quantity	SalePrice	LineTotal
1	1	5	$42.00	$210.00
1	2	3	$16.00	$48.00
2	2	6	$16.00	$96.00
3	1	2	$42.00	$48.00
3	3	7	$8.00	$56.00
4	3	3	$8.00	$24.00
4	4	9	$24.00	$216.00

Orders and Products Notes

Created OrderDetails table for many-to-many relationship
- Stores ProductID, OrderID, and Quantity
- Added columns for SalePrice and LineTotal

Updated Products table
- Removed Order information and added listprice

OrdersDetails Notes

Storing SalePrice allows product value to change
- Always have the actual price the customer paid

Storing LineTotal makes for easier queries
- Violates Third Normal Form
  - Every value must only depend on the key
    - LineTotal depends on SalePrice and Quantity
    - No calculated values

Using Normalization

Use normalization as a guide
- Not a series of absolute rules
- Some denormalization is standard

Most people don't think in terms of normalization
- After a while, basic database design starts to become innate

In a Nutshell…

Don't duplicate column data
- How many products can we order?

Don't duplicate row data
- How many orders will a customer have?

Don't store calculated fields*
- You can always figure out the answer
- Commonly violated for read performance and ease of use

Ensure data will remain consistent if changes occur
- What happens if a product price changes

Working with Data

Manipulating Data

Retrieve data
- `SELECT` statements

Create data
- `INSERT` statements

Modify data
- `UPDATE` statements

Delete data
- `DELETE` statements

Result Sets vs. Row Based Logic

SQL statements can update sets of rows
- Referred to as a result set
- Bounded by a predicate or filter

SQL can update individual rows
- Done through cursors
- Problems
  - Complex logic
  - Slower performance

Variables

Variables store temporary data

Components
- Name
  - All variables start with an @ sign
  - System variables start with @@
- Data type
  - What type of information will be stored
    - Integer
    - DateTime
- Value
  - The information being stored
  - Can be NULL, meaning no value

```
DECLARE @<Name> <DataType>;
  SET @<Name> = <Value>;
```