DAVID M. KROENKE and DAVID J. AUER

DATABASE CONCEPTS, 4th Edition

The Relational Model

Chapter Objectives

- Learn the conceptual foundation of the relational model
- Understand how relations differ from non-relational tables
- Learn basic relational terminology
- Learn the meaning and importance of keys, foreign keys, and related terminology
- Understand how foreign keys represent relationships
- Learn the purpose and use of surrogate keys
- Learn the meaning of functional dependencies
- Learn to apply a process for normalizing relations

Entity

- An **entity** is something of importance to a user that needs to be represented in a database
- An **entity** represents **one theme** or topic
- In an entity-relationship model (discussed in Chapter 4), **entities are restricted to things that can be represented by a single table**

Relation

- A **relation** is a two-dimensional table that has specific characteristics
- The table dimensions, like a matrix, consist of rows and columns

Characteristics of a Relation

- Rows contain data about an entity
- Columns contain data about attributes of the entity
- Cells of the table hold a single value
- All entries in a column are of the same kind
- Each column has a unique name
- The order of the columns is unimportant
- The order of the rows is unimportant
- No two rows may be identical

A Sample Relation

Attributes

Entity

EmployeeNumber	FirstName	LastName
100	Mary	Abernathy
101	Jerry	Cadley
104	Alex	Copley
107	Megan	Jackson

Example

EmployeeNumber	LastName	Email	Department	DeptPhone
100	Johnson	JJ@somewhere.com	Accounting	834-1100
200	Abernathy	MA@somewhere.com	Finance	834-2100
300	Smathers	LS@somewhere.com	Finance	834-2100
400	Caruthers	TC@somewhere.com	Accounting	834-1100
500	Jackson	TJ@somewhere.com	Production	834-4100
600	Caldera	EC@somewhere.com	Legal	834-3100
700	Bandalone	RB@somewhere.com	Legal	834-3100

Is this a Relation or Not

A Nonrelation Example

EmployeeNumber	Phone	LastName
100	335-6421, 454-9744	Abernathy
101	215-7789	Cadley
104	610-9850	Copley
107	299-9090	Jackson

Why this is non-relation ?

A Nonrelation Example

Why this is non-relation ?

EmployeeNumber	Phone	LastName
100	335-6421	Abernathy
101	215-7789	Cadley
104	610-9850	Copley
100	335-6421	Abernathy
107	299-9090	Jackson

Example

EmployeeNumber	FirstName	LastName	Department	Email	Phone
100	Jerry	Johnson	Accounting	JJ@somewhere.com	236-9987
200	Mary	Abernathy	Finance	MA@somewhere.com	444-8898
300	Liz	Smathers	Finance	LS@somewhere.com	777-0098
400	Tom	Caruthers	Accounting	TC@somewhere.com	236-9987
				Fax:	236-9987
				Home:	555-7171
500	Tom	Jackson	Production	TJ@somewhere.com	444-9980
600	Eleanore	Caldera	Legal	EC@somewhere.com	767-0900
				Fax:	236-9987
				Home:	555-7171
700	Richard	Bandalone	Legal	RB@somewhere.com	767-0900

Is this a Relation or Not

Example

EmployeeNumber	FirstName	LastName	Department	Email	Phone	Comment
100	Jerry	Johnson	Accounting	JJ@somewhere.com	236-9987	Joined the Accounting Department in March after completing his MBA at night. Will sit for CPA exam this fall.
200	Mary	Abernathy	Finance	MA@somewhere.com	444-8898	
300	Liz	Smathers	Finance	LS@somewhere.com	777-0098	
400	Tom	Caruthers	Accounting	TC@somewhere.com	236-9987	
500	Tom	Jackson	Production	TJ@somewhere.com	444-9980	
600	Eleanore	Caldera	Legal	EC@somewhere.com	767-0900	
700	Richard	Bandalone	Legal	RB@somewhere.com	767-0900	Is a full time consultant to legal on a retainer basis.

Is this a Relation or Not

Terminology

Synonyms...

Table	Row	Column
File	Record	Field
Relation	Tuple	Attribute

A Key

- A key is one (or more) columns of a relation that is (are) used to identify a row.

Unique

or

Nonunique

Uniqueness of Keys

Unique Key	Non-unique Key
Data value is unique for each row.	Data value may be shared among several rows.
Consequently, the key will uniquely identify a **single row**.	Consequently, the key will identify **set of rows.**

Key Example

EMPLOYEE (EmployeeNumber, LastName, Department, Email, Phone)

EmployeeNumber	LastName	Email	Department	DeptPhone
100	Johnson	JJ@somewhere.com	Accounting	834-1100
200	Abernathy	MA@somewhere.com	Finance	834-2100
300	Smathers	LS@somewhere.com	Finance	834-2100
400	Caruthers	TC@somewhere.com	Accounting	834-1100
500	Jackson	TJ@somewhere.com	Production	834-4100
600	Caldera	EC@somewhere.com	Legal	834-3100
700	Bandalone	RB@somewhere.com	Legal	834-3100

A Composite Key

- A **composite key** is a key that contains two or more attributes
- For a key to be unique, often it must become a composite key

EmployeeNumber	LastName	Email	Department	DeptPhone
100	Johnson	JJ@somewhere.com	Accounting	834-1100
200	Abernathy	MA@somewhere.com	Finance	834-2100
300	Smathers	LS@somewhere.com	Finance	834-2100
400	Caruthers	TC@somewhere.com	Accounting	834-1100
500	Jackson	TJ@somewhere.com	Production	834-4100
600	Caldera	EC@somewhere.com	Legal	834-3100
700	Bandalone	RB@somewhere.com	Legal	834-3100

Composite Key Example

To identify a family member, you need to know FamilyID, FirstName, and a Suffix (e.g., Jr.)

The composite key is:
(FamilyID, FirstName, Suffix)

One needs to know the value of all three columns to uniquely identify an individual.

A Candidate Key

- A **candidate key** is called "candidate" because it is a candidate to be selected as key (primary).
- A candidate key is a **unique key.**

EMPLOYEE (EmployeeNumber, FirstName, LastName, Department, Email, Phone)

EmployeeNumber, Email, Phone

(FirstName, LastName, Department)

A Primary Key

- A **primary key** is a candidate key chosen to be the main key for the relation.
- If you know the value of the **primary key**, you will be able to uniquely identify a single row.

Task performed by a Primary Key

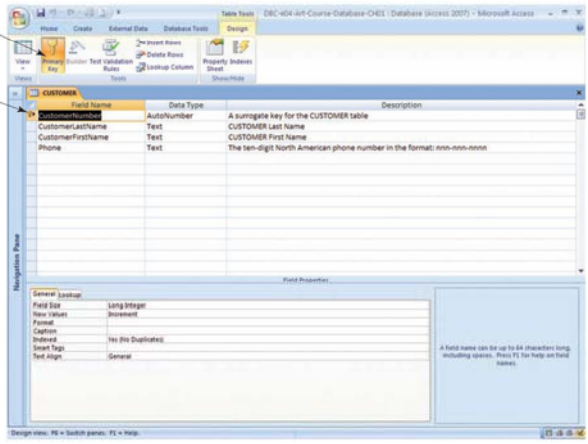
The primary key is used for the following 4 tasks;

1. First, it is used to uniquely identify the rows in a table.
2. Second, it is used to represent rows in relationships.
3. Third, most DBMS products use the values of the primary key to organize the storage of the relation.
4. Finally, primary keys are used in indexes and other structures to improve performance for search operations.

Defining the Primary Key in Microsoft Access

Primary Key button

The **key symbol** indicates which column or columns are being used as the primary key



Defining the Primary Key in Microsoft SQL Server 2005

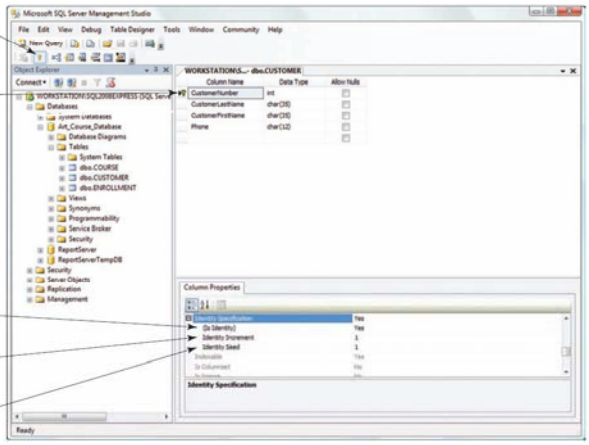
Primary Key button

The **key symbol** indicates which column or columns are being used as the primary key

The **Is Identity** setting

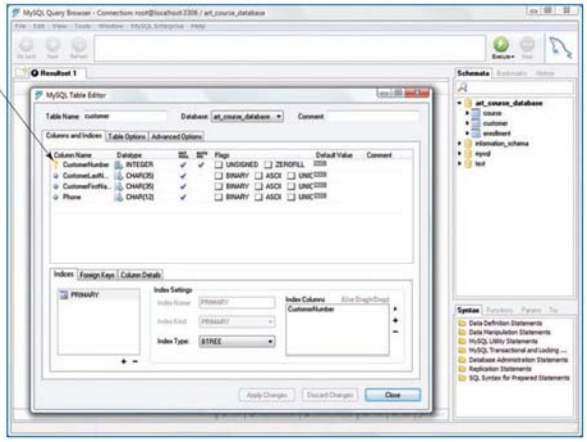
The **Identity Increment** setting

The **Identity Seed** setting



Defining the Primary Key in MySQL

The **key symbol** indicates which column or columns are being used as the primary key



A Surrogate Key

- A **surrogate key** is a unique, numeric value that is added to a relation by the DBMS to serve as the primary key.
- Surrogate key values have **no meaning to users** and are usually hidden on forms, queries and reports.
- A surrogate key is often used in place of a composite primary key.

More on Surrogate Key

A surrogate key is an artificial key that is created to act as the primary key for a relation.

The surrogate key is a unique, numeric value that is appended to the relation (by DBMS).

Surrogate keys are used in situations when no suitable primary key exists within the user data, or when all available primary keys within the data are too cumbersome for an efficient design.

Surrogate key values have no meaning to the users and are normally hidden on all forms, reports, and displays. Most DBMS products have the ability to automatically generate values for surrogate keys as needed.

Surrogate Key Example

If the Family Member primary key is FamilyID, FirstName, Suffix, it would be easier to append and use a surrogate key of FamMemberID.

FamilyID, FirstName and Suffix remain in the relation.

PROPERTY (Street, City, State, Zip, OwnerID)

Primary key is long and non-numeric.

PROPERTY (PropertyID, Street, City, State, Zip, OwnerID)

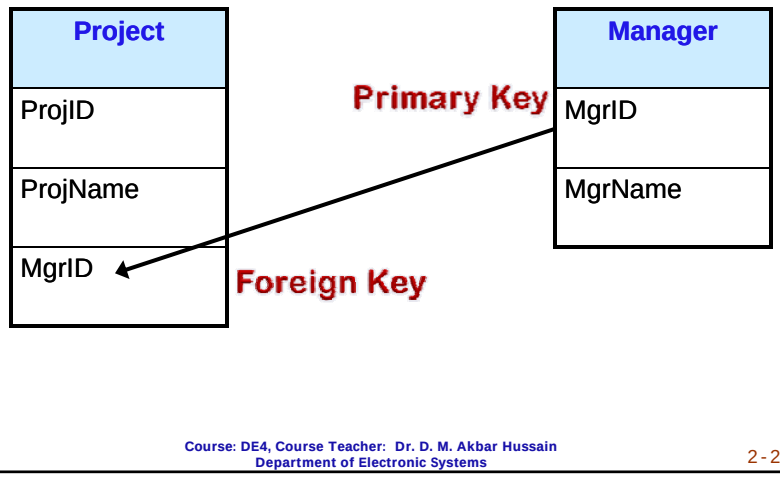
Relationships Between Tables

- A table may be related to other tables
- For example:
  - An Employee works in a Department
  - A Manager controls a Project

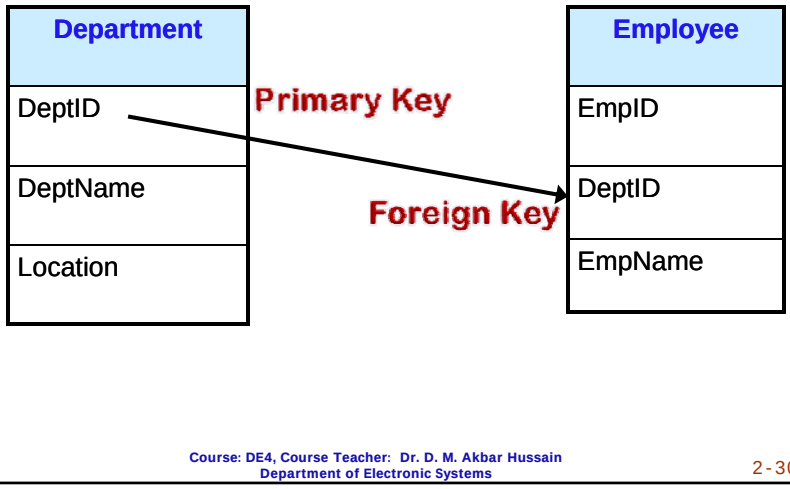
A Foreign Key

- To preserve relationships, you may need to create a **foreign key.**
- A foreign key is a primary key from one table placed into another table.
- The key is called a foreign key in the table that received the key.

Foreign Key Example I



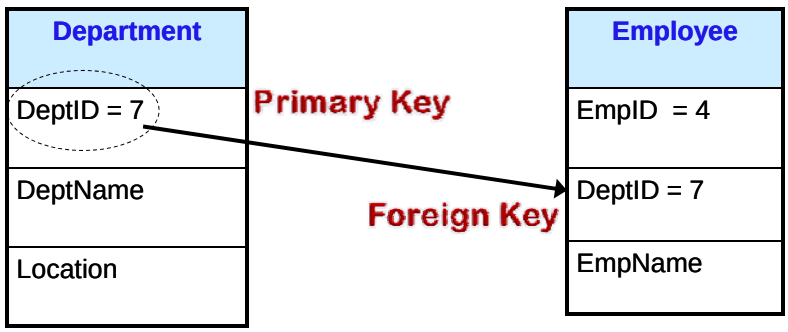
Foreign Key Example II



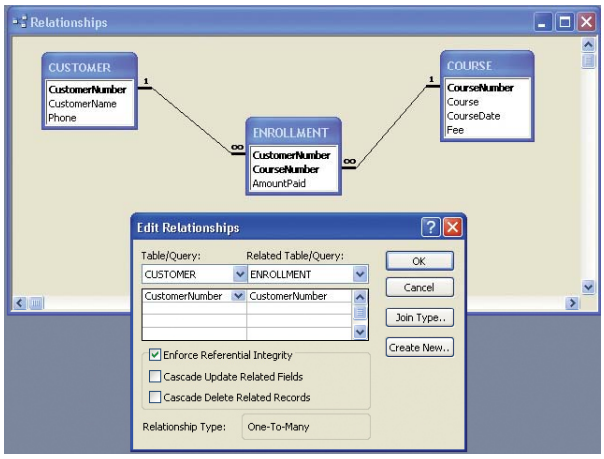
Referential Integrity

- **Referential integrity** states that every value of a foreign key must match a value of an existing primary key
- For example (see previous slide)
  - If EmpID = 4 in EMPLOYEE has a DeptID = 7 (a foreign key), a Department with DeptID = 7 must exist in DEPARTMENT
  - The primary key value must exist before the foreign key value is entered.

Foreign Key



Referential Integrity in Microsoft Access

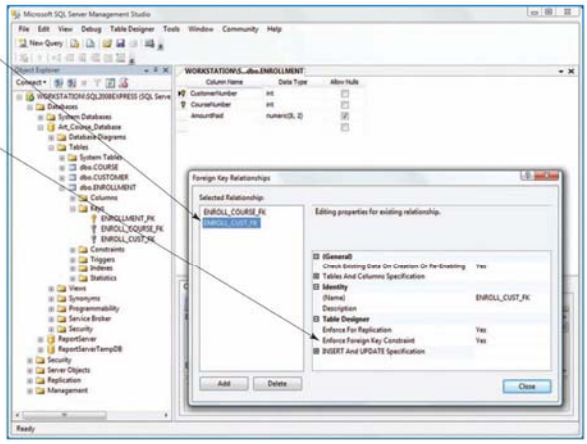


Identify Primary & Foreign Key

Foreign Keys in Microsoft SQL Server 2005

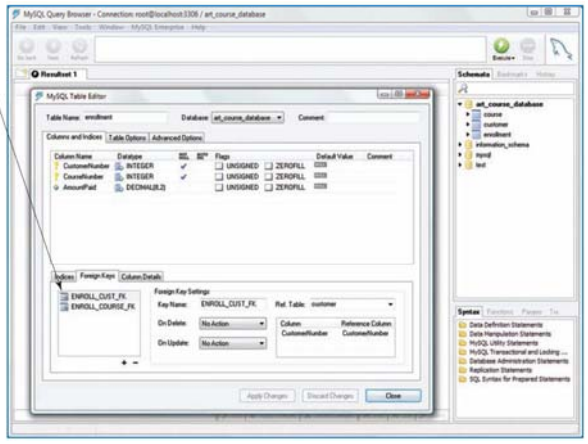
The relationship is between CUSTOMER and ENROLLMENT

We are enforcing the foreign key constraint— which *is* the referential integrity constraint



Foreign Keys in MySQL

The relationship is between CUSTOMER and ENROLLMENT



The Null Value

- A **Null value** means that no data was entered
- This is different from a zero, space character or tab character

ItemNum	ItemName	Color	Quantity
100	Small T-Shirt	Red	15
150	Small T-Shirt	Blue	5
200	Small T-Shirt	Green	7
300	Med T-Shirt	Red	8
400	Spring Hat		5

The Problem of Null Values

- A Null is often ambiguous. It could mean…
  - The column value is not appropriate for the specific row
  - The column value is not decided
  - The column value is unknown
- Each may have entirely different implications

Functional Dependency

- A relationship between attributes in which one attribute (or group of attributes) determines the value of another attribute in the same table
- Illustration…
  - The price of one cookie can determine the price of a box of 12 cookies, so the BoxPrice is dependent on quantity & Unit price of Cookie.

(CookiePrice, Qty) ⟶ BoxPrice

Determinants

- The attribute (or attributes) that we use as the starting point (the variable on the left side of the equation) is called a *determinant*

Candidate/Primary Keys and Functional Dependency

- By definition…

  A candidate key of a relation will functionally determine all other attributes in the row

- Likewise, by definition…

  A primary key of a relation will functionally determine all other attributes in the row

EmployeeNumber	FirstName	LastName	Department	Email	Phone
100	Jerry	Johnson	Accounting	JJ@somewhere.com	236-9987
200	Mary	Abernathy	Finance	MA@somewhere.com	444-8898
300	Liz	Smathers	Finance	LS@somewhere.com	777-0098
400	Tom	Caruthers	Accounting	TC@somewhere.com	236-9987
500	Tom	Jackson	Production	TJ@somewhere.com	444-9980
600	Eleanore	Caldera	Legal	EC@somewhere.com	767-0900
700	Richard	Bandalone	Legal	RB@somewhere.com	767-0900

Primary Key and Functional Dependency Example

(EmployeeID) → (EmpLastName, EmpPhone)

(ProjectID) → (ProjectName, StartDate)

Relation & Data

OBJECT (ObjectColor, Weight, Shape)

Object Color	Weight	Shape
Red	5	Ball
Blue	3	Cube
Yellow	7	Cube

Primary & Candidate Key (Formal Definition)

This actually means that primary key is *"one or more attributes which can functionally determine all other attributes of the relation".*

Normalization

- Functional Dependencies and determinants can be used to form relations.
- A process of analyzing a relation to ensure that it is *well formed.*
- More specifically, if a relation is **normalized** (well formed), rows can be inserted, deleted or modified without creating update anomalies.
- Normalization is a complex topic.
- Requirement is that: A Table or Relation should have one theme.

Normalization

ADVISER_LIST(AdviserID, AdviserName, Department, Phone, Office, StudentID, StudentName)

StudentID → (AdviserID, AdviserName, Department, Phone, Office, StudentName)

AdviserID → (AdviserName, Department, Phone, Office)

Normalization Principles

- Relational design principles for normalized relations:
  - To be a well-formed relation, every ***determinant*** must be a ***candidate key***
  - Any relation that is not well formed should be broken into two or more well-formed relations.

Normalization Process

- Identify all Candidate Keys of the relation.
- Identify all the functional dependencies in the relation.
- Examine the determinants, if any determinant is not a candidate key then obviously it is not a welformed relation, so:
  - Place the columns of the functional dependencies in a new relation of their own.
  - Make the determinant of the functional dependency the primary key of the new relation.
  - Leave a copy of the determinant as a foreign key in the original relation.
  - Create a referential integrity constriants between original relation new relation.
- Repeat previous step until every determinant of every relation is a candidate key.

Normalization Example

StudentNum	StudentName	DormName	DormCost
100	Smith	Stephens	$3,500.00
200	Johnson	Alexander	$3,800.00
300	Abernathy	Horan	$4,000.00
400	Smith	Alexander	$3,800.00
500	Wilcox	Stephens	$3,500.00
600	Webber	Horan	$4,000.00
700	Simon	Stephens	$3,500.00

(StudentNum) → (StudentName, DormName, DormCost)

However, if...

(DormName) → (DormCost)

Then DormCost should be placed into its own relation, resulting in the relations:

(StudentID) → (StudentName, DormName)

(DormName) → (DormCost)

Normalization Example

StudentNum	StudentName	DormName
100	Smith	Stephens
200	Johnson	Alexander
300	Abernathy	Horan
400	Smith	Alexander
500	Wilcox	Stephens
600	Webber	Horan
700	Simon	Stephens

DormName	DormCost
Stephens	$3,500.00
Alexander	$3,800.00
Horan	$4,000.00

Normalization Example

Attorney	ClientNumber	ClientName	MeetingDate	Duration
Boxer	1000	ABC, Inc	5/5/2006	2.00
Boxer	2000	ZZZ Partners	5/5/2006	5.50
James	1000	ABC, Inc	5/7/2006	3.00
Boxer	1000	ABC, Inc	5/9/2006	4.00
Wu	3000	Malcomb Zoe	5/11/2006	7.00

(Attorney,ClientNumber) → (ClientName, MeetingDate, Duration)

However, if...

(ClientNumber) → (ClientName)

Then ClientName should be placed into its own relation, resulting in the relations:

(AttorneyID,ClientID) → (ClientName, MeetingDate, Duration)

(ClientID) → (ClientName)

Normalization Example

Attorney	ClientNumber	MeetingDate	Duration
Boxer	1000	5/5/2006	2.00
Boxer	2000	5/5/2006	5.50
James	1000	5/7/2006	3.00
Boxer	1000	5/9/2006	4.00
Wu	3000	5/11/2006	7.00

ClientNumber	ClientName
1000	ABC, Inc
2000	ZZZ Partners
3000	Malcomb Zoe

Modification Problem

PrescriptionNumber	Date	Drug	Dosage	CustomerName	CustomerPhone	CustomerEmail
P10001	10/17/2006	DrugA	10mg	Alvin Smith	575.323.2233	ASmith@somewhere.com
P10003	10/17/2006	DrugB	35mg	Jeff Rhodes	575.334.4455	JRhodes@somewhere.com
P10004	10/17/2006	DrugA	20mg	Sarah Smith	575.323.2233	SSmith@somewhere.com
P10007	10/18/2006	DrugC	20mg	Michael Frye	575.345.6677	MFrye@elsewhere.com
P10010	10/18/2006	DrugC	25mg	Jeff Rhodes	575.334.4455	JRhodes@somewhere.com

CustomerName	CustomerPhone	CustomerEmail
Alvin Smith	575.323.2233	ASmith@somewhere.com
Jeff Rhodes	575.334.4455	JRhodes@somewhere.com
Sarah Smith	575.323.2233	SSmith@somewhere.com
Michael Frye	575.345.6677	MFrye@elsewhere.com
Jeff Rhodes	575.334.4455	JRhodes@somewhere.com

PrescriptionNumber	Date	Drug	Dosage	CustomerEmail
P10001	10/17/2006	DrugA	10mg	ASmith@somewhere.com
P10003	10/17/2006	DrugB	35mg	JRhodes@somewhere.com
P10004	10/17/2006	DrugA	20mg	SSmith@somewhere.com
P10007	10/18/2006	DrugC	20mg	MFrye@elsewhere.com
P10010	10/18/2006	DrugC	25mg	JRhodes@somewhere.com