DATABASE SYSTEMS

DESIGN IMPLEMENTATION AND MANAGEMENT

INTERNATIONAL EDITION

ROB • CORONEL • CROCKETT

Chapter 3
The Relational Database Model

In this chapter, you will learn:

- That the relational database model takes a logical view of data
- The relational model's basic components are relations implemented through tables in a relational DBMS
- How relations are organized in tables composed of rows (tuples) and columns (attributes)

In this chapter, you will learn (continued):

- About relational database operators, the data dictionary, and the system catalog
- How data redundancy is handled in the relational database model
- Why indexing is important

A Logical View of Data

- Relational model
  - Enables programmer to view data logically rather than physically
- Table
  - Has advantages of structural and data independence
  - Resembles a file from conceptual point of view
  - Easier to understand than its hierarchical and network database predecessors

What is a Relation?

- The concept of a relation is modelled on a mathematical construct
- In mathematics,a relation is formally defined as:

Given a number of sets D1, D2, ..., Dn (which are not necessarily distinct), R is a relation on these n sets,it is a set of tuples each of which has its first element from D1, second element form D2, and so on.

Example of a Relation

- Assume we have two sets (*n* =2), one of student's last names (STU_LNAME) and one of the department codes (DEPT_CODE) where they have enrolled.

  STU_LNAME {Bowser, Smithson, Brewer, Robertson}

  DEPT_CODE {BIOL, CIS, EDU}

- Then a relation can be defi ned over the sets STU_LNAME and DEPT_CODE as:

  R = {(Bowser, BIOL), (Smithson, CIS),(Brewer, EDU),(Robertson, EDU)}

Tables and Relations

- Table: two-dimensional structure composed of rows and columns
- Contains group of related entities = an entity set
  - Terms entity set and table are often used interchangeably

Tables and Relations (continued)

- Table also called a relation because the relational model's creator, Codd, used the term relation as a synonym for table
- Think of a table as a persistent relation:
  - A relation whose contents can be permanently saved for future use

Properties of a Relation

TABLE 3.1 Properties of a relation

1	A table is perceived as a two-dimensional structure composed of rows and columns.
2	Each table row (**tuple**) represents a single entity occurrence within the entity set and must be distinct. Duplicate rows are not allowed in a relation.
3	Each table column represents an attribute, and each column has a distinct name.
4	Each cell or column/row intersection in a relation should contain only an atomic value – that is a single data value. Multiple values are not allowed in the cells of a relation.
5	All values in a column must conform to the same data format. For example, if the attribute is assigned an integer data format, all values in the column representing that attribute must be integers.
6	Each column has a specific range of values known as the **attribute domain**.
7	The order of the rows and columns is immaterial to the DBMS.
8	Each table must have an attribute or a combination of attributes that uniquely identifies each row.

Example Relation

FIGURE 3.1(a) The relation LECTURER

Table name: LECTURER

EMP_NUM	LECTURER_OFFICE	LECTURER_EXTENSION	LECTURER_HIGH_DEGREE
103	DRE 156	6783	Ph.D.
104	DRE 102	5561	MA
105	KLR 229D	8665	Ph.D.
106	KLR 126	3899	Ph.D.
110	AAK 160	3412	Ph.D.
114	KLR 211	4436	Ph.D.
155	AAK 201	4440	Ph.D.

FIGURE 3.1(b) The non-relational table COURSE

Table name: COURSE

CRS_CODE	COURSE_NAME
CIS-220	Introduction to Microcomputing Assembly Language Programming
CIS-420	Database Design and Implementation Introduction to Databases Data Modelling: An Introduction
QM-261	Intro. to Statistics Statistical Applications

Attributes and Domains

- Each attribute is a named column within the relational table and draws its values from a **domain**.
- The domain of values for an attribute should contain only atomic values and any one value should not be divisible into components.
- No attributes with more than one value are allowed.

Degree and Cardinality

- **Degree** and **cardinality** are two important properties of the relational model.
- A relation with *N* columns and *N* rows is said to be of degree *N* and cardinality *N*.
- The degree of a relation is the number of its attributes and the cardinality of a relation is the number of its tuples.
- The product of a relation's degree and cardinality is the number of attribute values it contains.

Degree and Cardinality

FIGURE 3.2 Degree and cardinality of the DEPARTMENT relation

Table name: DEPARTMENT

	DEPT_NAME	DEPT_ADDRESS	DEPT_EXTENSION
ACCT	Accounting	KLR 211, Box 52	3119
ART	Fine Arts	BBG 185, Box 128	2278
BIOL	Biology	AAK 230, Box 415	4117
CIS	Computer Info. Systems	KLR 333, Box 56	3245

Cardinality = 4

Degree = 4

Relational Schema

- A **relational schema** is a textual representation of the database tables, where each table is described by its name followed by the list of its attributes in parentheses.
- A relational schema *R* can be formally defined as

  R={*a1*, *a2*,...,*an*} where *a1*…*an* are a set of attributes belonging to the relation.

Keys

- Consists of one or more attributes that determine other attributes
- Primary key (PK) is an attribute (or a combination of attributes) that uniquely identifies any given entity (row)
- Key's role is based on determination
  - If you know the value of attribute A, you can look up (determine) the value of attribute B

Keys (continued)

TABLE 3.2 Student classification

Hours completed	Classification
Less than 30	UG1
30–59	UG2
60–89	UG3
90 or more	PG

Relational Database Keys

- Composite key
  - Composed of more than one attribute
- Key attribute
  - Any attribute that is part of a key
- Superkey
  - Any key that uniquely identifies each row
- Candidate key
  - A superkey without redundancies

Keys (continued)

- Nulls:
  - No data entry
  - Not permitted in primary key
  - Should be avoided in other attributes
  - Can represent
    - An unknown attribute value
    - A known, but missing, attribute value
    - A "not applicable" condition
  - Can create problems when functions such as COUNT, AVERAGE, and SUM are used
  - Can create logical problems when relational tables are linked

Keys (continued)

- Controlled redundancy:
  - Makes the relational database work
  - Tables within the database share common attributes that enable the tables to be linked together
  - Multiple occurrences of values in a table are not redundant when they are required to make the relationship work
  - Redundancy exists only when there is unnecessary duplication of attribute values

Keys (continued)

FIGURE 3.4 An example of a simple relational database

Database name: Ch03_SaleCo

Table name: PRODUCT Primary key: PROD_CODE Foreign key: VEND_CODE

PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	€10.23	23	232
123-21UUY	Houselite chain saw, 16cm bar	€150.09	4	235
QER-34256	Sledge hammer, 16kg. head	€14.72	6	231
SRE-657UG	Rat-tail file	€2.36	15	232
ZZX/3245Q	Steel tape, 12m length	€5.36	8	235

link

VEND_CODE	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
230	Shelly K. Smithson	7325	555-1234
231	James Johnson	0181	123-4536
232	Annelise Crystall	7325	224-2134
233	Candice Wallace	0113	342-6567
234	Arthur Jones	0181	123-3324
235	Henry Ortozo	0181	899-3425

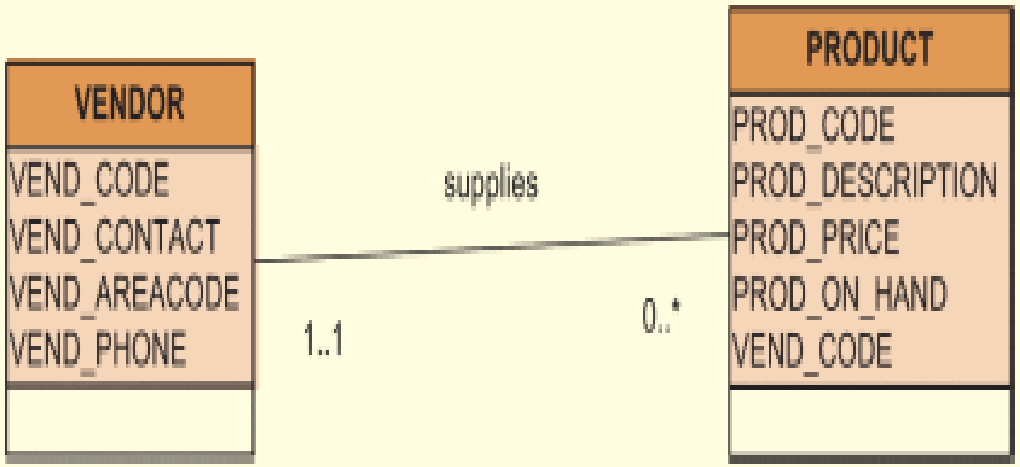
Table name: VENDOR

Primary key: VEND_CODE

Foreign key: none

Keys (continued)

FIGURE 3.5 The UML entity relationship diagram for the CH03_SaleCo database



Keys (continued)

- Foreign key (FK)
  - An attribute whose values match primary key values in the related table
- Referential integrity
  - FK contains a value that refers to an existing valid tuple (row) in another relation
- Secondary key
  - Key used strictly for data retrieval purposes

Keys (continued)

TABLE 3.3 Relational database keys

Key type	Definition
Superkey	An attribute (or combination of attributes) that uniquely identifies each row in a table.
Candidate key	A minimal superkey. A superkey that does not contain a subset of attributes that is itself a superkey.
Primary key	A candidate key selected to uniquely identify all other attribute values in any given row. Cannot contain null entries.
Secondary key	An attribute (or combination of attributes) used strictly for data retrieval purposes.
Foreign key	An attribute (or combination of attributes) in one table whose values must either match the primary key in another table or be null.

Integrity Rules

TABLE 3.4 Integrity rules

Entity Integrity	Description
Requirement	All primary key entries are unique, and no part of a primary key may be null.
Purpose	Each row will have a unique identity, and foreign key values can properly reference primary key values.
Example	No invoice can have a duplicate number, nor can it be null. In short, all invoices are uniquely identified by their invoice number.
Referential Integrity	**Description**
Requirement	A foreign key may have either a null entry (as long as it is not a part of its table's primary key) or an entry that matches the primary key value in a table to which it is related. (Every non-null foreign key value *must* reference an *existing* primary key value.)
Purpose	It is possible for an attribute NOT to have a corresponding value, but it will be impossible to have an invalid entry. The enforcement of the referential integrity rule makes it impossible to delete a row in one table whose primary key has mandatory, matching, foreign key values in another table.
Example	A customer might not yet have an assigned sales representative (number), but it will be impossible to have an invalid sales representative (number).

Integrity Rules (continued)

FIGURE 3.6 An illustration of integrity rules

Database name: Ch03_InsureCo
Table name: CUSTOMER
Primary key: CUS_CODE
Foreign key: AGENT_CODE

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	0181	844-2573	12-Mar-06	502
10011	Dunne	Leona	K	0161	894-1238	23-May-06	501
10012	Smith	Kathy	W	0181	894-2285	05-Jan-06	502
10013	Olowski	Paul	F	0181	894-2180	20-Sep-06	
10014	Orlando	Myron		0181	222-1672	04-Dec-06	501
10015	O'Brian	Amy	B	0161	442-3381	29-Aug-06	503
10016	Brown	James	G	0181	297-1228	01-Mar-06	502
10017	Williams	George		0181	290-2556	23-Jun-06	503
10018	Farriss	Anne	G	1061	382-7185	09-Nov-06	501
10019	Smith	Olette	K	0181	297-3809	18-Feb-06	503

Table name: AGENT
Primary key: AGENT_CODE
Foreign key: none

AGENT_CODE	AGENT_LNAME	AGENT_AREACODE	AGENT_PHONE	AGENT_YTD_SLS
501	Alby	0161	228-1249	€1,371,008.46
502	Hahn	0181	882-1244	€3,923,932.59
503	Okon	0181	123-5589	€2,444,244.52

Integrity Rules (continued)

TABLE 3.5 A dummy variable value used as a flag

AGENT_CODE	AGENT_AREACODE	AGENT_PHONE	AGENT_LNAME	AGENT_YTD_SALES
-99	0000	000-0000	None	€0.00

Chapter 5, Entity Relationship (ER) Modelling, discusses several ways in which nulls may be handled.

The Data Dictionary and System Catalog

- Data dictionary
  - Provides detailed accounting of all tables found within the user/designer-created database
  - Contains (at least) all the attribute names and characteristics for each table in the system
  - Contains metadata—data about data
  - Sometimes described as "the database designer's database" because it records the design decisions about tables and their structures

A Sample Data Dictionary

TABLE 3.6 A sample data dictionary

Table Name	Attribute Name	Contents	Type	Format	Domain	Required	PK or FK	FK Referenced table
CUSTOMER	CUS_CODE	Customer account code	CHAR(5)	99999	10000–99999	Y	PK	
	CUS_LNAME	Customer last name	VARCHAR2(20)	Xxxxxxx		Y		
	CUS_FNAME	Customer first name	VARCHAR2(20)	Xxxxxxx		Y		
	CUS_INITIAL	Customer initial	CHAR(1)	X				
	CUS_RENEW_DATE	Customer insurance renewal date	DATE	dd-mmm-yyyy				
	AGENT_CODE	Agent code	CHAR(3)	999	100–999		FK	AGENT
AGENT	AGENT_CODE	Agent code	CHAR(3)	999		Y	PK	
	AGENT_AREACODE	Agent area code	CHAR(4)	999		Y		
	AGENT_PHONE	Agent telephone number	CHAR(14)	999-9999		Y		
	AGENT_LNAME	Agent last name	VARCHAR2(20)	Xxxxxxx		Y		
	AGENT_YTD_SLS	Agent year-to-date sales	NUMBER(9,2)	9,999,999.99	0.00–9,999,999.99	Y		

FK = Foreign key
PK = Primary key
CHAR = Fixed character length data (1–255 characters)
VARCHAR2 = Variable character length data (1–4,000 characters)
NUMBER = Numeric data (NUMBER(9,2) is used to specify numbers with two decimal places and up to nine digits, including the decimal places. Some RDBMSs permit the use of a MONEY or CURRENCY data type.)

The Data Dictionary and System Catalog (continued)

- System catalog
  - Contains metadata
  - Detailed system data dictionary that describes all objects within the database
  - Terms "system catalog" and "data dictionary" are often used interchangeably
  - Can be queried just like any user/designer-created table

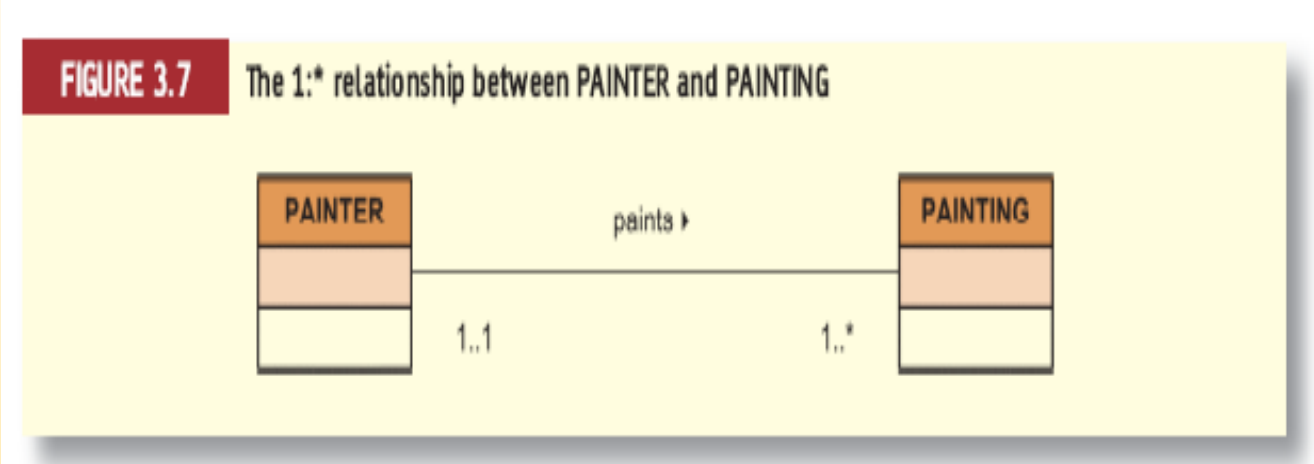
Relationships within the Relational Database

- 1:* relationship
  - Relational modeling ideal
  - Should be the norm in any relational database design
- 1:1 relationship
  - Should be rare in any relational database design
- *:* relationships
  - Cannot be implemented as such in the relational model
  - *:* relationships can be changed into two 1:* relationships

The 1:* Relationship

- Relational database norm
- Found in any database environment

The 1:* Relationship (continued)



The 1:* Relationship (continued)

FIGURE 3.8 The implemented 1:* relationship between PAINTER and PAINTING

Database name: Ch03_Museum
Primary key: PAINTER_NUM

Table name: PAINTER
Foreign key: none

PAINTER_NUM	PAINTER_LNAME	PAINTER_FNAME	PAINTER_INITIAL
123	Ross	Georgette	P
126	Itero	Julio	G

PAINTING_NUM	PAINTING_TITLE	PAINTER_NUM
1338	Dawn Thunder	123
1339	Vanilla Roses To Nowhere	123
1340	Tired Flounders	126
1341	Hasty Exit	123
1342	Plastic Paradise	126

Table name: PAINTING
Primary Key: PAINTING_NUM

Foreign Key: PAINTER_NUM

The 1:* Relationship (continued)

FIGURE 3.9 The 1:* relationship between COURSE and CLASS

The 1:* Relationship (continued)

FIGURE 3.10 The implemented 1:* relationship between COURSE and CLASS

Database name: Ch03_TinyUniversity Table name: COURSE

Primary key: CRS_CODE Foreign key: none

CRS_CODE	DEPT_CODE	CRS_DESCRIPTION	CRS_CREDIT
ACCT-211	ACCT	Accounting I	3
ACCT-212	ACCT	Accounting II	3
CIS-220	CIS	Introduction to Microcomputing	3
CIS-420	CIS	Database Design and Implementation	4
QM-261	CIS	Introduction to Statistics	3
QM-362	CIS	Statistical Applications	4

Table name: CLASS

Primary key: CLASS_CODE Foreign key: CRS_CODE

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	LECT_NUM
10012	ACCT-211	1	MWF 8:00-8:50 a.m.	BUS311	105
10013	ACCT-211	2	MWF 9:00-9:50 a.m.	BUS200	105
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10015	ACCT-212	1	MWF 10:00-10:50 a.m.	BUS311	301
10016	ACCT-212	2	Th 6:00-8:40 p.m.	BUS252	301
10017	CIS-220	1	MWF 9:00-9:50 a.m.	KLR209	228
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10019	CIS-220	3	MWF 10:00-10:50 a.m.	KLR209	228
10020	CIS-420	1	W 6:00-8:40 p.m.	KLR209	162
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114
10022	QM-261	2	TTh 1:00-2:15 p.m.	KLR200	114
10023	QM-362	1	MWF 11:00-11:50 a.m.	KLR200	162
10024	QM-362	2	TTh 2:30-3:45 p.m.	KLR200	162

The 1:1 Relationship

- One entity can be related to only one other entity, and vice versa
- Sometimes means that entity components were not defined properly
- Could indicate that two entities actually belong in the same table
- As rare as 1:1 relationships should be, certain conditions absolutely require their use

The 1:1 Relationship (continued)

FIGURE 3.11 The 1:1 relationship between LECTURER and DEPARTMENT

The 1:1 Relationship (continued)

Database name: Ch03_TinyUniversity Table name: LECTURER

Primary key: EMP_NUM Foreign key: DEPT_CODE

EMP_NUM	DEPT_CODE	LECT_OFFICE	LECT_EXTENSION	LECT_HIGH_DEGREE
103	HIST	DRE 156	6783	Ph.D.
104	ENG	DRE 102	5561	MA
105	ACCT	KLR 229D	8665	Ph.D.
106	MKT/MGT	KLR 126	3899	Ph.D.
110	BIOL	AAK 160	3412	Ph.D.
114	ACCT	KLR 211	4436	Ph.D.
155	MATH	AAK 201	4440	Ph.D.
160	ENG	DRE 102	2248	Ph.D.
162	CIS	KLR 203E	2359	Ph.D.
191	MKT/MGT	KLR 409B	4016	DBA
195	PSYCH	AAK 297	3550	Ph.D.
209	CIS	KLR 333	3421	Ph.D.
228	CIS	KLR 300	3000	Ph.D.
297	MATH	AAK 194	1145	Ph.D.
299	ECON/FIN	KLR 284	2851	Ph.D.
301	ACCT	KLR 244	4683	Ph.D.
335	ENG	DRE 208	2000	Ph.D.
342	SOC	BBG 208	5514	Ph.D.
387	BIOL	AAK 230	8665	Ph.D.
401	HIST	DRE 156	6783	MA
425	ECON/FIN	KLR 284	2851	MBA
435	ART	BBG 185	2278	Ph.D.

The 1:* DEPARTMENT employs LECTURER relationship is implemented through the placement of the DEPT_CODE foreign key in the LECTURER table.

The 1:1 LECTURER chairs DEPARTMENT relationship is implemented through the placement of the EMP_NUM foreign key in the DEPARTMENT table.

Table name: DEPARTMENT

Primary key: DEPT_CODE

Foreign key: EMP_NUM

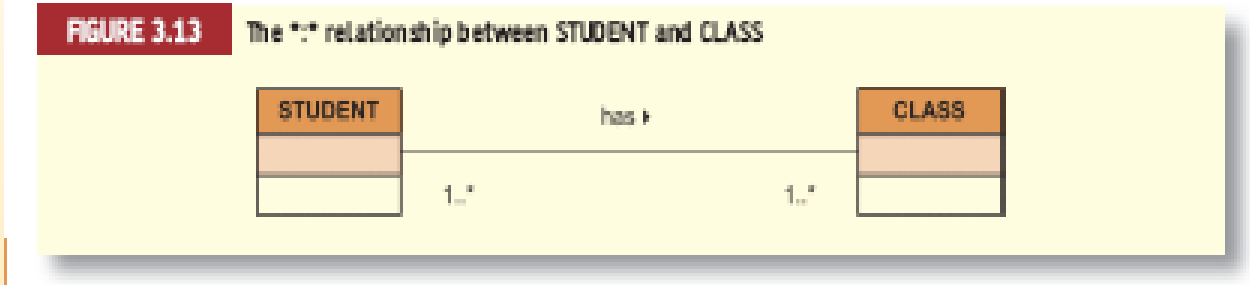
The 1:1 Relationship (continued)

DEPT_CODE	DEPT_NAME	SCHOOL_CODE	EMP_NUM	DEPT_ADDRESS	DEPT_EXTENSION
ACCT	Accounting	BUS	114	KLR 211, Box 52	3119
ART	Fine Arts	A&SCI	435	BBG 185, Box 128	2278
BIOL	Biology	A&SCI	397	AAK 230, Box 415	4117
CIS	Computer Info. Systems	BUS	209	KLR 333, Box 56	3245
ECON/FIN	Economics/Finance	BUS	299	KLR 284, Box 63	3126
ENG	English	A&SCI	160	DRE 102, Box 223	1004
HIST	History	A&SCI	103	DRE 156, Box 284	1867
MATH	Mathematics	A&SCI	297	AAK 194, Box 422	4234
MKT/MGT	Marketing/Management	BUS	108	KLR 126, Box 55	3342
PSYCH	Psychology	A&SCI	195	AAK 297, Box 438	4110
SOC	Sociology	A&SCI	342	BBG 208, Box 132	2008

The *:* Relationship

- Can be implemented by breaking it up to produce a set of 1:* relationships
- Can avoid problems inherent to *:* relationship by creating a composite entity or bridge entity

The *:* Relationship (continued)



The *:* Relationship (continued)

TABLE 3.7 Sample student enrolment data

Student's Last Name	Selected Classes
Bowser	Accounting 1, ACCT-211, code 10014 Intro to Microcomputing, CIS-220, code 10018 Intro to Statistics, QM-261, code 10021
Smithson	Accounting 1, ACCT-211, code 10014 Intro to Microcomputing, CIS-220, code 10018 Intro to Statistics, QM-261, code 10021

The *:* Relationship (continued)

FIGURE 3.14 The *:* relationship between STUDENT and CLASS]

Database name: Ch03_CollegeTry Table name: STUDENT

Primary key: STU_NUM Foreign key: none

STU_NUM	STU_LNAME	CLASS_CODE
321452	Bowser	10014
321452	Bowser	10018
321452	Bowser	10021
324257	Smithson	10014
324257	Smithson	10018
324257	Smithson	10021

Table name: CLASS

Primary Key: CLASS_CODE Foreign Key: STU_NUM

CLASS_ CODE	STU_ NUM	CRS_ CODE	CLASS_ SECTION	CLASS_TIME	CLASS_ ROOM	PROF_ NUM
10014	321452	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10014	324257	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	321452	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10018	324257	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10021	321452	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114
10021	324257	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114

The *:* Relationship (continued)

- Implementation of a composite entity
- Yields required *:* to 1:* conversion
- Composite entity table must contain at least the primary keys of original tables
- Linking table contains multiple occurrences of the foreign key values
- Additional attributes may be assigned as needed

The *:* Relationship (continued)

FIGURE 3.15 Converting the *:* relationship into two 1:* relationships

Database name: Ch03_CollegeTry2 Table name: STUDENT

Primary key: STU_NUM Foreign key: none

STU_NUM	STU_LNAME
321452	Bowser
324257	Smithson

Table name: ENROL

Primary key: CLASS_CODE+STU_NUM

Foreign keys: CLASS_CODE, STU_NUM

CLASS_CODE	STU_NUM	ENROLL_GRADE
10014	321452	C
10014	324257	B
10018	321452	A
10018	324257	B
10021	321452	C
10021	324257	C

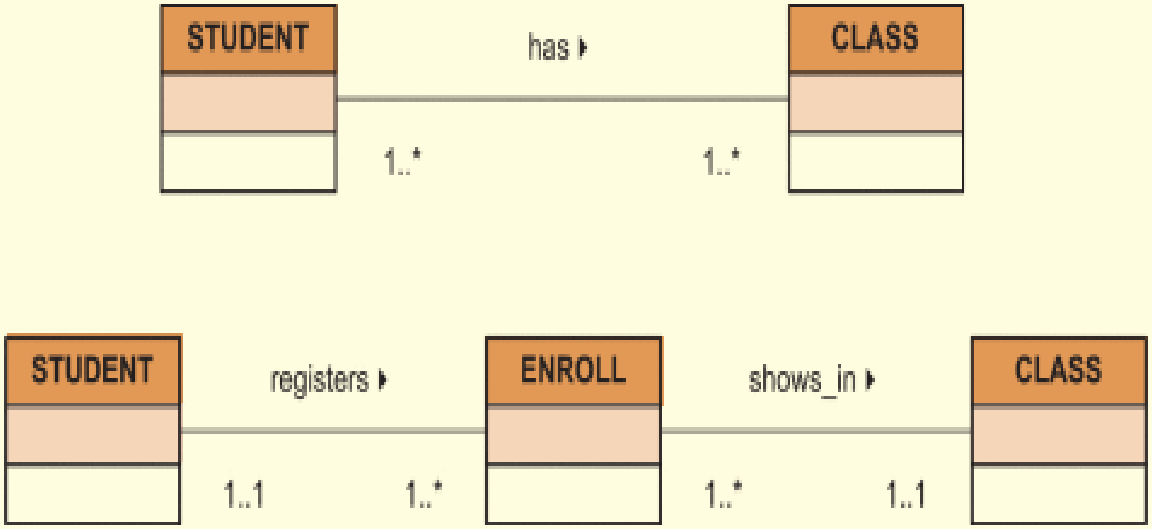
Table name: CLASS

Primary key: CLASS_CODE Foreign key: CRS_CODE

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114

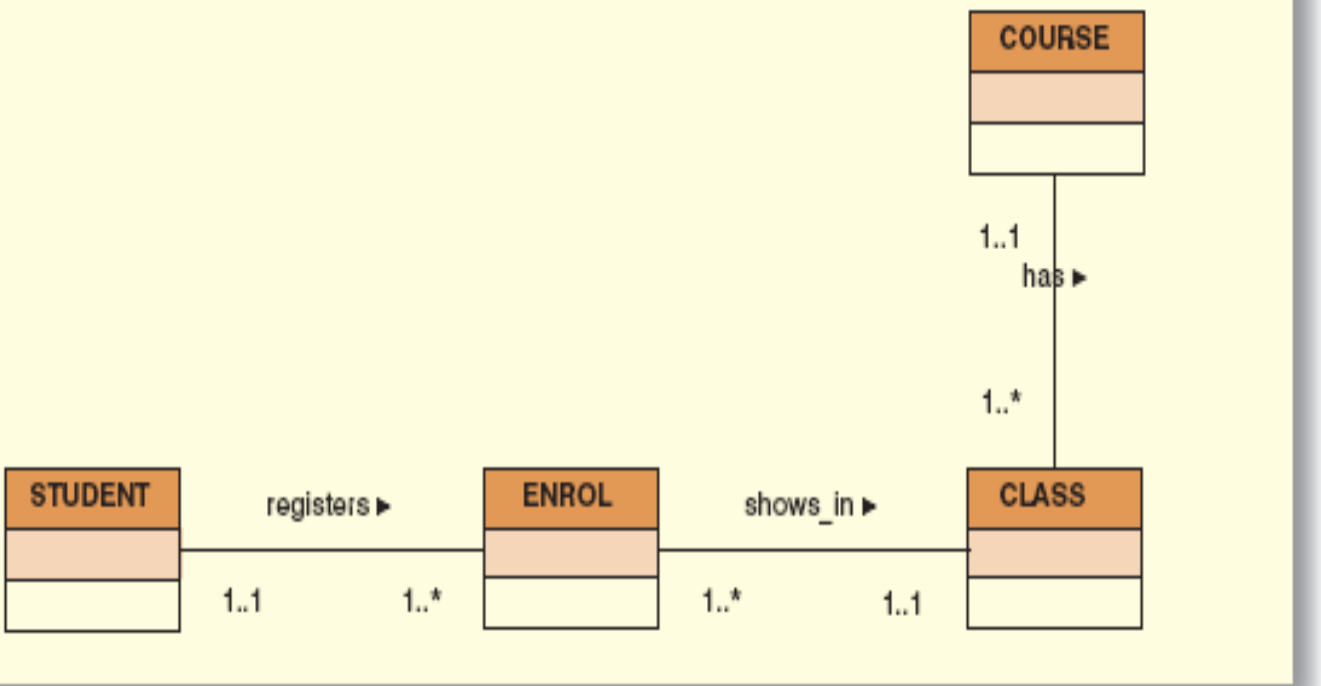
The *:* Relationship (continued)

FIGURE 3.16 Changing the *:* relationship to two 1:* relationships



The *:* Relationship (continued)

FIGURE 3.17 The expanded entity relationship model



Data Redundancy Revisited

- Data redundancy leads to data anomalies
  - Such anomalies can destroy the effectiveness of the database
- Foreign keys
  - Control data redundancies by using common attributes shared by tables
  - Crucial to exercising data redundancy control
- Sometimes, data redundancy is necessary

Data Redundancy Revisited (continued)

FIGURE 3.18 A small invoicing system

Database name: Ch03_SaleCo Table name: CUSTOMER

Primary key: CUS_CODE Foreign key: none

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE
10010	Ramas	Alfred	A	0181	844-2573
10011	Dunne	Leona	K	0161	894-1238
10012	Smith	Kathy	W	0181	894-2285
10013	Olowski	Paul	F	0181	894-2180
10014	Orlando	Myron		0181	222-1672
10015	O'Brian	Amy	B	0161	442-3381
10016	Brown	James	G	0181	297-1228
10017	Williams	George		0181	290-2556
10018	Farriss	Anne	G	0161	382-7185
10019	Smith	Olette	K	0181	297-3809

Table name: INVOICE

Primary key: INV_NUMBER Foreign key: CUS_CODE

INV_NUMBER	CUS_CODE	INV_DATE
1001	10014	08-Mar-07
1002	10011	08-Mar-07
1003	10012	08-Mar-07
1004	10011	09-Mar-07

Data Redundancy Revisited (continued)

Table name: LINE

Primary key: INV_NUMBER + LINE_NUMBER

Foreign key: INV_NUMBER, PROD_CODE

INV_NUMBER	LINE_NUMBER	PROD_CODE	LINE_UNITS	LINE_PRICE
1001	1	123-21UUY	1	€150.09
1001	2	SRE-657UG	3	€2.36
1002	1	QER-34256	2	€14.72
1003	1	ZZX/3245Q	1	€5.36
1003	2	SRE-657UG	1	€2.36
1003	3	001278-AB	1	€10.23
1004	1	001278-AB	1	€10.23
1004	2	SRE-657UG	2	€2.36

Table name: PRODUCT

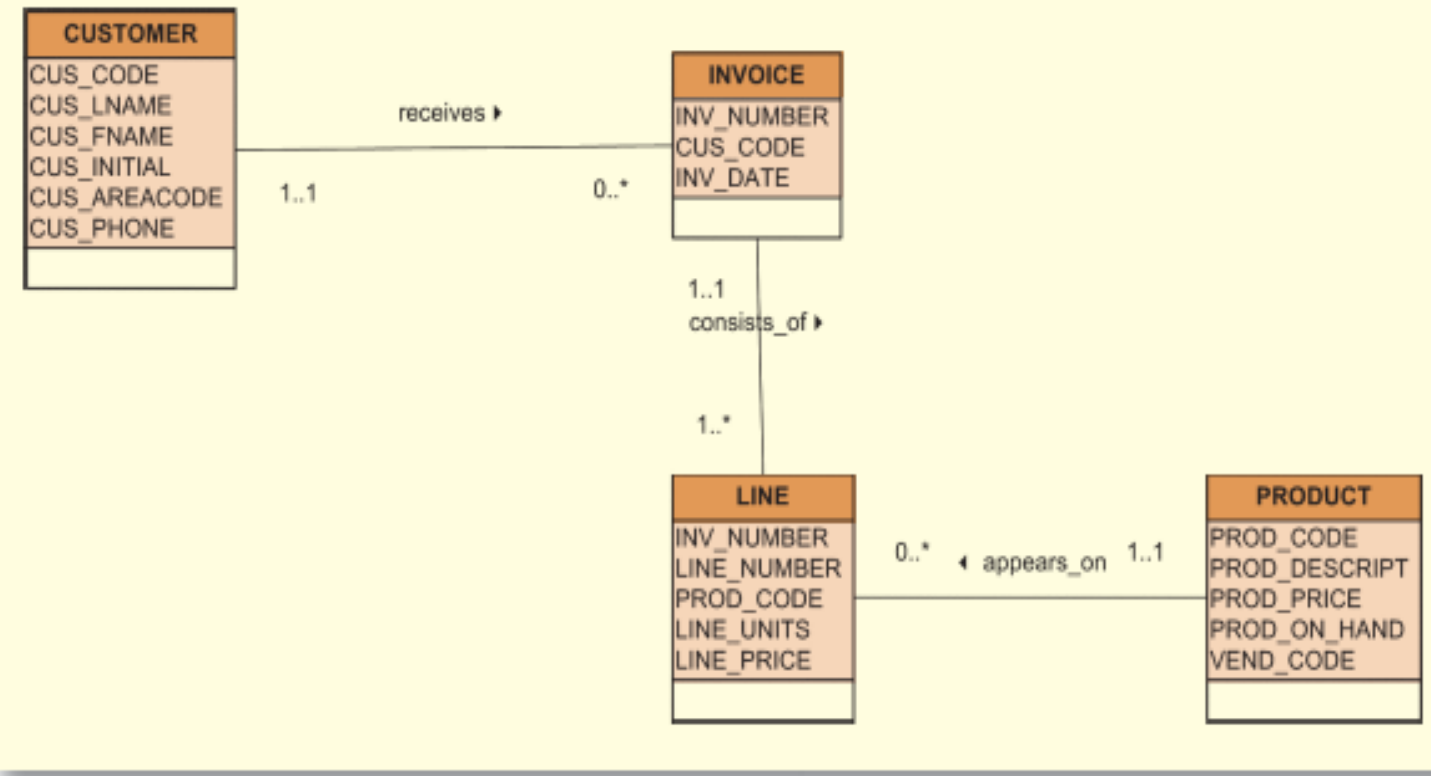
Primary key: PROD_CODE

Foreign key: none

PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	€10.23	23	232
123-21UUY	Houselite chain saw, 16-cm bar	€150.09	4	235
QER-34256	Sledge hammer, 16-kg head	€14.72	6	231
SRE-657UG	Rat-tail file	€2.36	15	232
ZZX/3245Q	Steel tape, 12-m length	€5.36	8	235

Data Redundancy Revisited (continued)

FIGURE 3.19 The Class ERD for the invoicing system

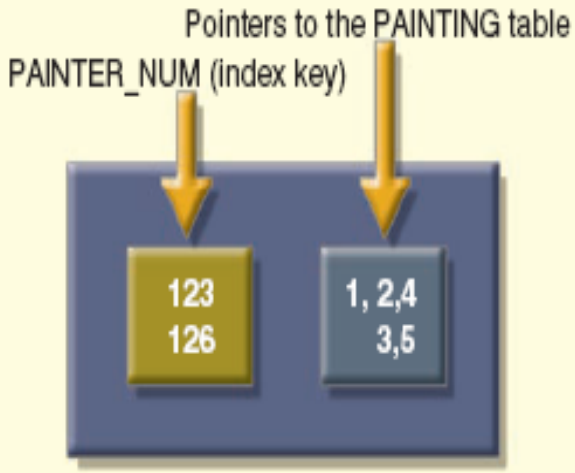


Indexes

- Arrangement used to logically access rows in a table
- Index key
  - Index's reference point
  - Points to data location identified by the key
- Unique index
  - Index in which the index key can have only one pointer value (row) associated with it
- Each index is associated with only one table

Indexes (continued)

FIGURE 3.20 Components of an index



Codd's Relational Database Rules

- In 1985, Codd published a list of 12 rules to define a relational database system
- The reason was the concern that many vendors were marketing products as "relational" even though those products did not meet minimum relational standards

Codd's Relational Database Rules (Continued)

TABLE 3.8 Dr. Codd's 12 relational database rules

Rule	Rule Name	Description
1	Information	All information in a relational database must be logically represented as column values in rows within tables.
2	Guaranteed Access	Every value in a table is guaranteed to be accessible through a combination of table name, primary key value and column name.
3	Systematic Treatment of Nulls	Nulls must be represented and treated in a systematic way, independent of data type.
4	Dynamic On-Line Catalogue Based on the Relational Model	The metadata must be stored and managed as ordinary data, that is, in tables within the database. Such data must be available to authorized users, using the standard database relational language.
5	Comprehensive Data Sublanguage	The relational database may support many languages. However it must support one well-defined declarative language with support for data definition, view definition, data manipulation (interactive and by program), integrity constraints, authorization and transaction management (begin, commit and rollback).
6	View Updating	Any view that is theoretically updatable must be updatable through the system.
7	High-Level Insert, Update and Delete	The database must support set-level inserts, updates and deletes.
8	Physical Data Independence	Application programs and ad hoc facilities are logically unaffected when physical access methods or storage structures are changed.
9	Logical Data Independence	Application programs and ad hoc facilities are logically unaffected when changes are made to the table structures that preserve the original table values (changing order of column or inserting columns).
10	Integrity Independence	All relational integrity constraints must be definable in the relational language and stored in the system catalog, not at the application level.
11	Distribution Independence	The end users and application programs are unaware and unaffected by the data location (distributed vs. local databases).
12	Nonsubversion	If the system supports low-level access to the data, there must not be a way to bypass the integrity rules of the database.
	Rule Zero	All preceding rules are based on the notion that in order for a database to be considered relational, it must use its relational facilities exclusively to manage the database.

Summary

- Relations are basic building blocks of a relational database.
- Keys are central to the use of relational tables
- Keys define functional dependencies
  - Superkey
  - Candidate key
  - Primary key
  - Secondary key
  - Foreign key

Summary (continued)

- Each table row must have a primary key which uniquely identifies all attributes
- Tables can be linked by common attributes. Thus, the primary key of one table can appear as the foreign key in another table to which it is linked
- Good design begins by identifying appropriate entities and attributes and the relationships among the entities. Those relationships (1:1, 1:*, and *:*) can be represented using ERDs.