Computer Science 202

Database Systems:
Relational Database Model

Objectives

- To learn the basic relational database components and concepts.
- To become familiar with the relational table's components and characteristics.
- To learn how keys are used in the relational database environment.
- To introduce the relational database operators.
- To develop a simple data dictionary.
- To examine basic entity relationships.

Relational Database Model

- Relational model was a change over File based systems
- Focus on logical design
- A logical representation of data
- Not so tightly tied to physical storage and implementation

Entities and Attributes

- **Entity** – something about which data is to be stored
- **Attribute –** what information about an Entity is to be stored

STUDENT
Student Number
Date of Birth
First name
Last name

Relationships

- Relations with in a RDBMS are represented as a **TABLE**
- A Database table consists of:
  - Rows (Entity Instances) (also known as ***tuples***)
  - Columns (Entity Attributes)

Student Number	Last Name	First Name
g01b0023	Bloggs	Joe
g98m5639	Mcginnis	Wilbur

Relational Tables

- Two Dimensional structure (Rows & Columns)
- Each tuple represents a single instance of an entity
- Each column represents an attribute with a distinct name
- Each row/column intersection represents a **single** data value
- All values in a column must confirm to the same data type
- Each column has a range of values known as the attribute **domain**
- The order of tuples and columns is immaterial to the DBMS

See R&C p59

Tables - Naming Constraints

- Table names restricted to 8 characters
- Column names limited to 10 characters
- Column names may not begin with a digit
- Avoid the use of a – (dash) rather use an underscore (_)

These are generic limitations. Each RDBMS system will have its own limitations and constraints.

Tables – Attribute data types

There are 4 primary data types used in RDBM systems

- Numeric - 1 , 1880, 34.5 , - 145.29
- Character – alphanumeric and other characters
- Date – Specialised data format
- Logical – Boolean values: True/False

Keys

- A **key** consists of one or more attributes that determines other attributes
- Key –DETERMINES→ Attribute(s)

  If A → B then

  B is functionally Dependant on A
- **Composite key**
  - Key consisting of more than one attribute

R&C p 63

More on dependency

- Full Functional Dependency
  - If B is dependant on a composite key A, but not on any subset of A.
  - Key(A,A`) → B
  - Key(A) !→ B && Key(A`) !→ B

NULL

A **NULL** value can be stored in an attribute in the following cases:

- Unknown attribute value
- Missing value
- "Not Applicable" condition

- An attribute that has Null cannot be part of a key

- A **NULL** is a special case and is an empty value, is NOT equivalent to a space, although they may look the same on screen

Key Types

- Primary Key
  - Key used for identification of entity instances
- Secondary Key
  - Usually used for indexing for data retrieval
- Foreign Key
  - A Key with values must match the primary key in another table, or be NULL
- Superkey
  - Any key that defines a attribute uniquely
- Candidate Key
  - Superkey without redundancies

Key Selection

- Choosing the right key

Database integrity in the RDBMS Entity Integrity

Requirement	All primary keys are unique, and no part may be NULL
Purpose	Ensures that each entity can be uniquely defined and that foreign keys can correctly reference tables

After R&C p68

Database integrity in the RDBMS Referential Integrity

Requirement	A foreign key may have a null entry or an entry matching the primary key in the table to which it is related
Purpose	Enforcement prevents deletion of an entity from a table while the primary key has mandatory matching foreign keys.

After R&C p68

Use of Flags in Referential Integrity

- A flag can be used instead of a NULL value, when foreign keys are in use
- Help ensure referential integrity
- Generally only useful with attributes that will not have any kind of operation performed on them.

Relational Database Operators

- A fully relational database is expected to support the following 8 operations on tables
  - SELECT
  - PROJECT
  - JOIN
  - INTERSECT
  - UNION
  - DIFFERENCE
  - PRODUCT
  - DIVIDE

Relational Database Operators SELECT

- SELECT performs horizontal Cuts across a table
- Can be used to select ALL rows or rows matching certain criteria

Relational Database Operators SELECT

Original table

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

SELECT ALL will yield

New table or list

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

SELECT only PRICE less than 2.00 will yield

P_CODE	P_DESCRIPT	PRICE
213345	9v battery	1.92
254467	100W bulb	1.47

SELECT only P_CODE=311452 will yield

P_CODE	P_DESCRIPT	PRICE
311452	Powerdrill	34.99

R&C p 72 Figure 2.9

Relational Database Operators PROJECT

- Project yields all rows for an attribute(or group of attributes)
- Performs a Vertical cut of the table

Relational Database Operators PROJECT

Original table

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

PROJECT PRICE yields

New table or list

PRICE
5.26
25.15
10.99
1.92
1.47
34.99

PROJECT P_DESCRIPT and PRICE yields

P_DESCRIPT	PRICE
Flashlight	5.26
Lamp	25.15
Box Fan	10.99
9v battery	1.92
100W bulb	1.47
Powerdrill	34.99

PROJECT P_CODE and PRICE yields

P_CODE	PRICE
123456	5.26
123457	25.15
123458	10.99
213345	1.92
254467	1.47
311452	34.99

R&C p 72 Figure 2.10

Relational Database Operators JOIN

- Join allows information from two or more tables to be combined
- Performs joins on independent tables using common attributes
- Used with foreign and primary keys
- One of the more powerful operations in the Relational database

Relational Database Operators JOIN

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE
1132445	Walker	32145	231
1217782	Adares	32145	125
1312243	Rakowski	34129	167
1321242	Rodriguez	37134	125
1542311	Smithson	37134	421
1657399	Vanloo	32145	231

Table name: AGENT

AGENT_CODE	AGENT_PHONE
125	6152439887
167	6153426778
231	6152431124
333	9041234445

Two Tables to be Joined

R&C p 73 Figure 2.11

Relational Database Operators JOIN

CUS_CODE	CUS_LNAME	CUS_ZIP	CUSTOMER.AGENT_CODE	AGENT.AGENT_CODE	AGENT_PHONE
1132445	Walker	32145	231	125	6152439887
1132445	Walker	32145	231	167	6153426778
1132445	Walker	32145	231	231	6152431124
1132445	Walker	32145	231	333	9041234445
1217782	Adares	32145	125	125	6152439887
1217782	Adares	32145	125	167	6153426778
1217782	Adares	32145	125	231	6152431124
1217782	Adares	32145	125	333	9041234445
1312243	Rakowski	34129	167	125	6152439887
1312243	Rakowski	34129	167	167	6153426778
1312243	Rakowski	34129	167	231	6152431124
1312243	Rakowski	34129	167	333	9041234445
1321242	Rodriguez	37134	125	125	6152439887
1321242	Rodriguez	37134	125	167	6153426778
1321242	Rodriguez	37134	125	231	6152431124
1321242	Rodriguez	37134	125	333	9041234445
1542311	Smithson	37134	421	125	6152439887
1542311	Smithson	37134	421	167	6153426778
1542311	Smithson	37134	421	231	6152431124
1542311	Smithson	37134	421	333	9041234445
1657399	Vanloo	32145	231	125	6152439887
1657399	Vanloo	32145	231	167	6153426778
1657399	Vanloo	32145	231	231	6152431124
1657399	Vanloo	32145	231	333	9041234445

R&C p 71 Figure 2.6

Relational Database Operators JOIN

CUS_CODE	CUS_LNAME	CUS_ZIP	CUSTOMER.AGENT_CODE	AGENT.AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	125	6152439887
1321242	Rodriguez	37134	125	125	6152439887
1312243	Rakowski	34129	167	167	6153426778
1132445	Walker	32145	231	231	6152431124
1657399	Vanloo	32145	231	231	6152431124

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	6152439887
1321242	Rodriguez	37134	125	6152439887
1312243	Rakowski	34129	167	6153426778
1132445	Walker	32145	231	6152431124
1657399	Vanloo	32145	231	6152431124

Joined tables before and after the removal of duplicate values

R&C p 74 Figure 2.13, 2.14

Relational Database Operators JOIN Advanced

- Previous example illustrated a NATURAL join
- EquiJoin – Join based on comparison of column values. Duplicates are not removed. Only applicable to the use of an equals (=) sign
- Theta Join – Same as Equijoin, but for other operators

Relational Database Operators JOIN – Outer Joins

- An outer join joins two tables with the following result
  - Matched pairs are retained
  - Unmatched values are filled with a NULL
- LEFT OUTER Join
  - Includes all the rows in the table on the left of the join statement, including those with no matches in the right table
- RIGHT OUTER Join
  - Includes all the rows in the table on the RIGHT of the join statement, including those with no matches in the LEFT table

LEFT OUTER Join

- LEFT OUTER Join
  - Includes all the rows in the table on the left of the join statement, including those with no matches in the right table

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	6152439887
1321242	Rodriguez	37134	125	6152439887
1312243	Rakowski	34129	167	6153426778
1132445	Walker	32145	231	6152431124
1657399	Vanloo	32145	231	6152431124
1542311	Smithson	37134	421	

RIGHT OUTER JOIN

- RIGHT OUTER Join
  - Includes all the rows in the table on the RIGHT of the join statement, including those with no matches in the LEFT table

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	6152439887
1321242	Rodriguez	37134	125	6152439887
1312243	Rakowski	34129	167	6153426778
1132445	Walker	32145	231	6152431124
1657399	Vanloo	32145	231	6152431124
			333	9041234445

Relational Database Operators INTERSECT

- INTERSECT outputs only those rows common to both tables
- Tables must be compatible, with the attributes the same data types in both

Relational Database Operators
INTERSECT

F_NAME
George
Jane
Elaine
Wilfred
Jorge

INTERSECT

F_NAME
Jane
William
Jorge
Dennis

yields

F_NAME
Jane
Jorge

R&C p 71 Figure 2.16

Relational Database Operators
UNION

- UNION combines all rows from two tables
- Tables must be compatible
- Attributes must have the same data type

Relational Database Operators
UNION

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

UNION

P_CODE	P_DESCRIPT	PRICE
345678	Microwave	160
345679	Dishwasher	500

yields

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99
345678	Microwave	160
345679	Dishwasher	500

R&C p 70 Figure 2.5

Relational Database Operators
DIFFERENCE

- Difference outputs all the rows in Table A not in table B
- Subtraction of table B from table A
- Tables must be compatible
- Attribute data types must be the same

Relational Database Operators
DIFFERENCE

F_NAME
George
Jane
Elaine
Wilfred
Jorge

DIFFERENCE

F_NAME
Jane
William
Jorge
Dennis

yields

F_NAME
George
Elaine
Wilfred

R&C p 71 Figure 2.7

Relational Database Operators
PRODUCT

- Product produces all possible combinations of the rows from Table A and B
- Also known as the Cartesian Product
- Table A is multiplied by Table B
- **A** (5 Rows) **PRODUCT B** (3 Rows)
  → Table **C** (15 Rows)

Relational Database Operators PRODUCT

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

PRODUCT

STORE	AISLE	SHELF
23	W	5
24	K	9
25	Z	6

yields

P_CODE	P_DESCRIPT	PRICE	STORE	AISLE	SHELF
123456	Flashlight	5.26	23	W	5
123456	Flashlight	5.26	24	K	9
123456	Flashlight	5.26	25	Z	6
123457	Lamp	25.15	23	W	5
123457	Lamp	25.15	24	K	9
123457	Lamp	25.15	25	Z	6
123458	Box Fan	10.99	23	W	5
123458	Box Fan	10.99	24	K	9
123458	Box Fan	10.99	25	Z	6
213345	9v battery	1.92	23	W	5
213345	9v battery	1.92	24	K	9
213345	9v battery	1.92	25	Z	6
311452	Powerdrill	34.99	23	W	5
311452	Powerdrill	34.99	24	K	9
311452	Powerdrill	34.99	25	Z	6
254467	100W bulb	1.47	23	W	5
254467	100W bulb	1.47	24	K	9
254467	100W bulb	1.47	25	Z	6

R&C p 71 Figure 2.8

Relational Database Operators DIVIDE

- Specific operation
- Requires a two column table which is divided by a single column table
- Table A and B must have a column of the same type
- Result contains values from table A which appear for all values in table B
- Not commonly used

Relational Database Operators DIVIDE

CODE	LOC
A	5
A	9
A	4
B	5
B	3
C	6
D	7
D	8
E	8

DIVIDE

CODE
A
B

yields

LOC
5

R&C p 76 Figure 2.17

The Data Dictionary

- Provides information about the data stored in the database – **MetaData**
- Holds information about table structures, attribute data types, relationships
- Usually used by the DB designer as part of the design process to provide a plan for implementation.

The Data Dictionary

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	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	VCHAR(20)	Xxxxxxxxxxx		Y		
	CUS_FNAME	Customer first name	VCHAR(20)	Xxxxxxxxxxxx		Y		
	CUS_INITIAL	Customer middle initial	CHAR(1)	X				
	CUS_RENEW_DATE	Customer insurance renewal date	DATE	DD_MM_YYYY				
	AGENT_CODE	Insurance agent code	CHAR(3)	999	100-999		FK	AGENT
AGENT								
	AGENT_CODE	Insurance agent code	CHAR(3)	999		Y	PK	
	AGENT_AREACODE	Agent's area code	CHAR(3)	999		Y		
	AGENT_PHONE	Agent's telephone number	CHAR(8)	999 – 9999		Y		
	AGENT_LNAME	Agent's last name	VARCHAR(20)			Y		
	AGENT_YTD_SALES	Agent's 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 to 255 characters.
VARCHAR = Variable character length data, 1 to 2,000 characters. May also be labeled VARCHAR2.
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 a CURRENCY data type.

R&C p 77 Table2.6

System Catalog

- Closely related to a data dictionary
- Holds detailed information about all objects in a database
  - Creation time
  - Owner
  - Access Control
  - Index data
- Most modern systems only have a System catalog

Relationships

- Three Classes of relationship within an RDBMS
  - 1:1 → one to one
  - 1:M → one to many
  - M:M or M:N → many to many

Entity Relation Diagrams (ERD)

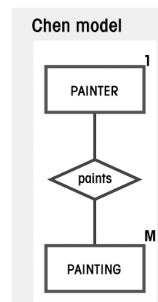
- The Entity Relation Diagram is used to graphically represent the Relational model of the data.
- Two Models are used
  - CHEN
  - Crow's Feet

Entity Relation Diagrams (ERD)

- Entity names are nouns
- Entity name is usually capitalised
- Relationships are described by passive verbs
- Relationships written in lower case

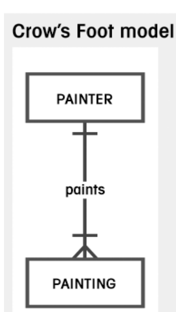
ERD – CHEN Model

- Entities contained in a rectangle
- Relationships are contained in a diamond
- '1' side of a relationship uses a 1
- Many side uses M or N

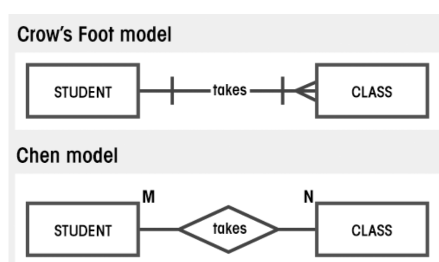


ERD – Crow's Foot Model

- Entities contained in a rectangle
- Relationships are written on the connecting line
- '1' side of a relationship uses a bar
- Many side uses a Crows Foot ←



ERD Many to Many (M:N) relation



Problems with M:N relations

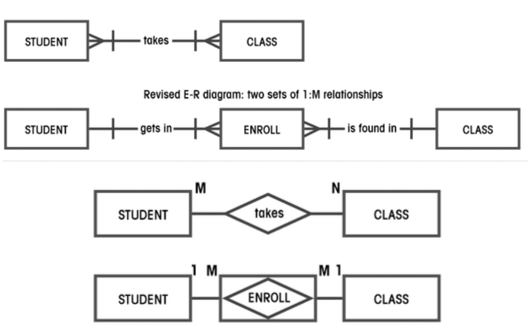
- M:N tables contain many redundancies
- Difficult to work with
- Difficult to implement

- See Rob and Cornel p 83-87 for a full worked example

Simplifying M:N relations

- M:N problem can be resolved by the introduction of an artificial entity known as a **composite** or **bridging** entity
- Linking table holds the primary keys of the two tables to be linked, as foreign keys
- Reduces the M:N relation to two 1:M relationships

Simplified M:N model



Data Redundancy

- Use of Foreign Keys can reduce redundancy
- Limited controlled redundancy can be desirable
  - For speed in lookups
  - For holding further information without the need for a join

Indexing

- Indexes are used for improved performance
- Index is a set of pointers in a lookup table, which allows rapid referencing of table rows
- Allows the DBMS to go directly to relevant rows, rather than having to do a linear search

Indexing Example

Value	Pos
123	1
456	6
789	4

123		
451		
231		
789		
456		