Intro to Database

Sunday, 8 October 2023 10:56 am

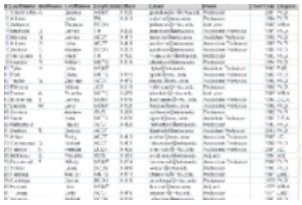
Data Versus Information

- Data
  - Raw facts
    - Raw indicates that facts have not yet been processed to reveal their meaning
  - Raw data must be properly formatted for
    - Storage
    - Processing
    - Presentation
- Information
  - Result of processing raw data to reveal its meaning
  - Data are the foundation of information
    - Bedrock of knowledge
    - Body of information and facts about a specific subject
- Key Points
  - Data constitute the building blocks of info
  - Information is produced by processing data
  - Information is used to reveal the meaning of data

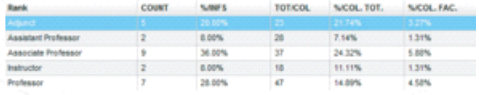
Transformation from Data to Information

1. Data Entry

2. Raw Data

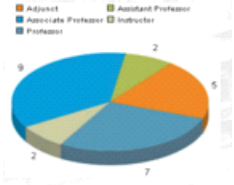


3. Information in Summary Format



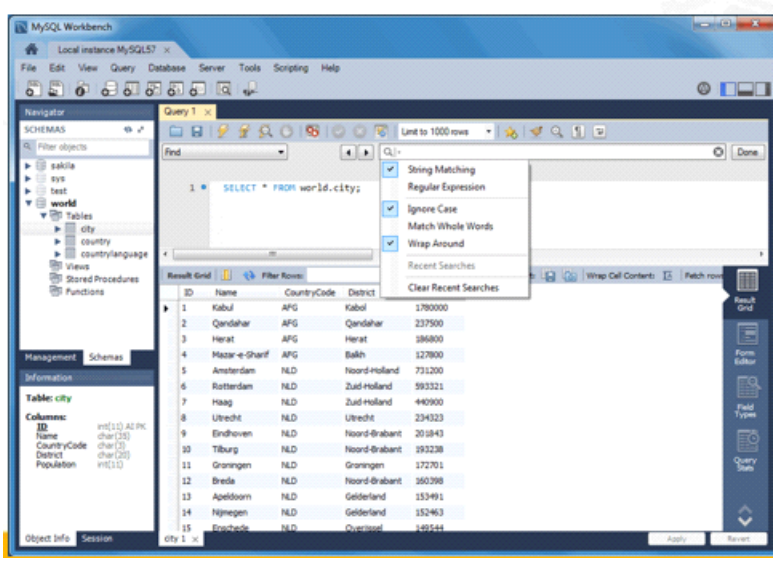
Rank	COUNT	%RF'S	TOT/COL	%COL. TOT.	%COL. FAC.
Adjunct	5	20.00%	25	21.74%	3.27%
Assistant Professor	2	8.00%	26	7.14%	1.31%
Associate Professor	9	36.00%	37	24.32%	5.88%
Instructor	2	8.00%	18	11.11%	1.31%
Professor	7	28.00%	47	14.89%	4.58%

4. Information in Visual Format



Introduction to Database

- **Database**
  - Collection of data typically describing activities of one or more related organizations



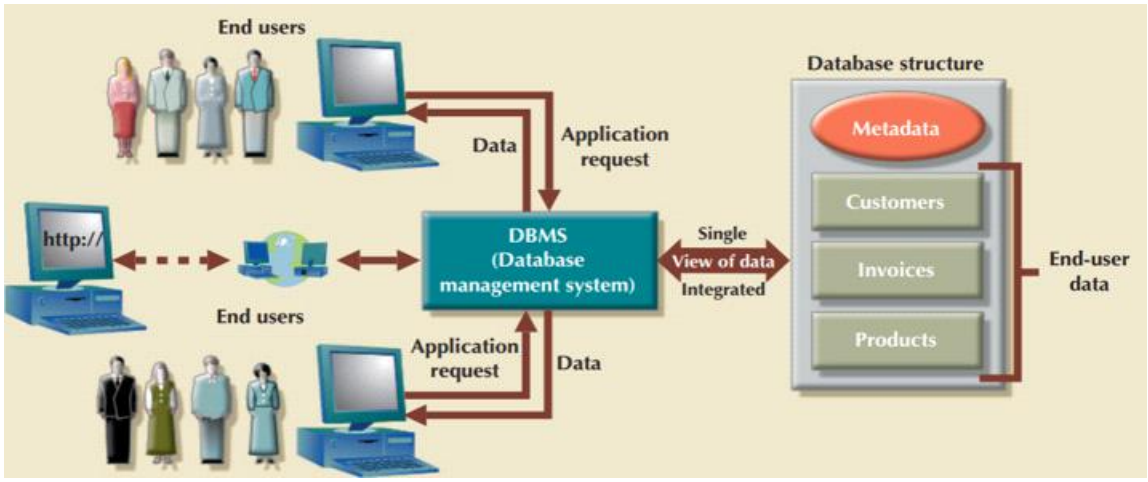
DBMS

- **Database Management System (DBMS)**
  - Collection of programs that
    - Manages database structure
    - Controls access to data stored in the database
    - Enables users to create and maintain a database

Historical Perspective of DBMS

- 1960s
  - First DBMS designed by **Charles Bachman** at General Electric
    - Was called **Integrated Data Store (IDS)**
- Late 1960s
  - IBM developed **Information Management System (IMS) DBMS**
    - Used even today in many major installations
  - SABRE system for making airline reservations was jointly developed by American Airlines and IBM
- 1970s
  - **Edgar Codd,** IBM's San Jose Research Laboratory
    - Proposed a new data representation framework called **relational data model**
- 1980s
  - **SQL**
    - Standardized in the late 1980s
    - Current standard, SQL-92 was adopted by the **ANSI (American National Standards Institute) and ISO (International Standards Organization)**
- Late 1980s and 1990s
  - Advances have been made in many areas of database systems
  - Popular vendors were IBM's DB2, Oracle 8, Informix UDS and others

Roles and Advantages of DBMS



Advantages of DBMS

- Improved data sharing
- Improved data security
- Better data integration
- Minimized data inconsistency
- Improved decision making
- Increased end-user productivity

Actors of the DBMS Scene

- Database Administrators
  - DBA
  - Responsible for authorizing access to the database, for coordinating and monitoring use, and for acquiring software and hardware resources as needed
  - Accountable for problems such as breach of security and poor system response time
- Database Designers
  - Responsible for identifying the data to be store in the database
  - Choosing appropriate structures to represent and store data
- End Users
  - People whose jobs require access to the database for querying, updating and generating reports
  - Database primarily exists for their use
  - Several categories of end users
    - Casual End-Users
      - Occasionally access database
      - May need different information each time
    - Naïve End-Users
      - Main job function revolves around constantly querying and updating the database, using types of queries and updates called canned transactions that have been carefully programmed and tested
    - Sophisticated End-Users
      - Include engineers, scientists, business analysts, and others who thoroughly familiarize themselves with the facilities of DBMS so as to implement applications to meet their complex requirements
    - Stand-alone End-Users
      - Maintain personal databases by using ready-made program packages that provide easy-to-use menu-based or graphics-based interfaces
- System Analysts and Applications Programmers
  - System analyst
    - Determine requirements of end users, especially naïve and parametric end users, and develop specifications for canned transactions that meet these requirements
  - Application programmers
    - Implement these specifications as programs

        - Test, debug, document, and maintain canned transactions
- Operators and Maintenance Personnel
    - System administration personnel who are responsible for actual running and maintenance of the hardware and software environment for the database system

Types of Database

- DBMS can support different types of databases
- Database can be classified according to
    - Number of users
    - Database location(s)
    - Expected type
    - Extent of use
    - Structure of data
- According to number of Users:
    - Single-user database
        - Supports only one user at a time
    - Multiuser database
        - Supports multiple users at the same time
    - Workgroup database
        - Multiuser database that supports relatively small number of users
        - Usually fewer than 50
    - Enterprise database
        - Used by entire organization
        - Supports many users
        - More than 50, usually hundreds
- According to location
    - Centralized database
        - Supports data located at a single site
    - Distributed database
        - Support data distributed across several different sites
- According to the Extent of use
    - Operational database
        - Designed primarily to support company's day to day operation
        - Sometimes referred to as transactional or production database
    - Data Warehouse
        - Focuses primarily on storing data used to generate information required to make tactical or strategic decisions

Database Design

- Refers to activities that focus on design of database structure that will be used to store and manage end-user data

File System

- Manual File System
    - Manual, paper-and-pencil systems
    - Typically, this was accomplished through a system of file folders and filing cabinets
- Computerized File System
    - DP (Data Processing) specialist was hired to create a computer base system that would track data and produce required reports

Basic File Terminology

- Data

    - "Raw" Facts
    - Telephone number, birth date, customer name, year-to-date (YTD) sale value
- Field
    - Character or group of characters (alphabetic or numeric) that has specific meaning
    - Used to define and store data
- Record
    - Logically connected set of one or more fields that describes person, place, or thing
- File
    - Collection of related records

Problem with File System Data Processing
- Lengthy development times
- Difficulty of getting quick answers
- Complex system administration
- Lack of security and limited data sharing
- Extensive programming

Data Redundancy
- Exists when same data are stored unnecessarily at different places
- Uncontrolled data redundancy sets stage for
    - Poor data security
    - Data inconsistency
    - Data-entry errors
    - Data integrity problems

Data Anomalies
- Develop when not all of required changes in redundant data are made successfully
    - Update anomalies
    - Insertion anomalies
    - Deletion anomalies

Data Integrity
- Defined as condition in which all of data in database are consistent with real world events and conditions
- Data are accurate
    - No data inconsistencies
- Data are verifiable
    - Data will always yield consistent results

Database System
- Consists of logically relate data stored in a single logical data repository

Components of Database Environment
- Hardware
- Software (OS, DBMS, App Program)
- People (Actors, BTS-people)
- Procedure
- Data

Disadvantages of Database
- Increased cost
- Management complexity
- Maintaining cost
- Vendor dependence
- Requires technical skills

Relational Database

Sunday, 8 October 2023 11:08 am

Logical View of Data

- Database stores and manages both data and metadata
- Relational data model allows designer to focus on logical representation of data and its relationships, rather than on physical storage details
- In short, relational model enables you to view logically rather than physically

Table Characteristics

- Table
  - Two dimensional structure composed of rows and column
  - Also called as relation by Edgar Codd
- Characteristics of a table (relation)
  - Perceived as two-dimensional structure
  - Composed of rows and columns
  - Entries in table are single valued
  - Rows are sometimes referred to as records
  - Columns are sometimes labeled as fields
  - Each table column represents an attribute
  - Each column has distinct name
  - Each row/column intersection represents a single value
  - All values in column must conform to the same data format
  - Each table must have an attribute or combination of attributes that uniquely identifies each row

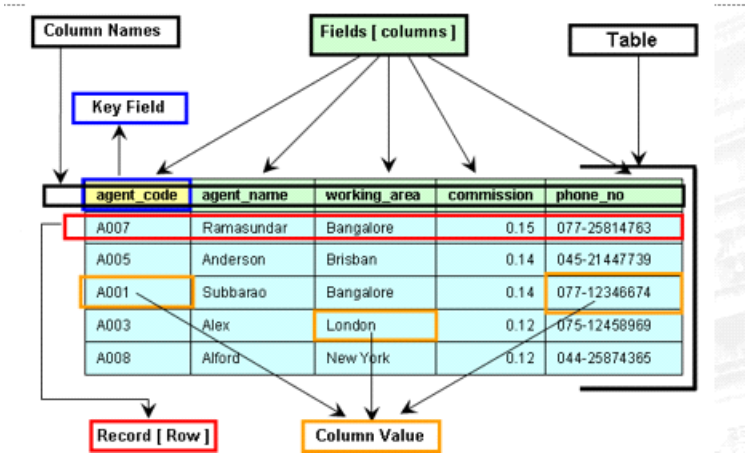


FIGURE 3.1 STUDENT TABLE ATTRIBUTE VALUES

Table name: STUDENT — Database name: Ch03_TinyCollege

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS	STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
321452	Bowser	William	C	12-Feb-1985	42	So	2.84	No	BIOL	2134	205
324257	Smithson	Anne	K	15-Nov-1991	81	Jr	3.27	Yes	CIS	2256	222
324258	Brewer	Juliette		23-Aug-1979	36	So	2.26	Yes	ACCT	2256	228
324269	Oblonski	Walter	H	16-Sep-1986	66	Jr	3.09	No	CIS	2114	222
324273	Smith	John	D	30-Dec-1968	102	Sr	2.11	Yes	ENGL	2231	199
324274	Katinga	Raphael	P	21-Oct-1989	114	Sr	3.15	No	ACCT	2267	228
324291	Robertson	Gerald	T	08-Apr-1983	120	Sr	3.87	No	EDU	2267	311
324299	Smith	John	B	30-Nov-1996	15	Fr	2.92	No	ACCT	2315	230

STU_NUM = Student number
STU_LNAME = Student last name
STU_FNAME = Student first name
STU_INIT = Student middle initial
STU_DOB = Student date of birth
STU_HRS = Credit hours earned
STU_CLASS = Student classification
STU_GPA = Grade point average
STU_TRANSFER = Student transferred from another institution
DEPT_CODE = Department code
STU_PHONE = 4-digit campus phone extension
PROF_NUM = Number of the professor who is the student's advisor

Keys

- Important in relational model
- Used to ensure that each row in a table is uniquely identifiable
- Key consists of one or more attributes that determine other attributes

Dependencies

- Determination
  - State in which knowing value of one attribute makes it possible to determine the value of another

STU_NUM→ STU_NAME

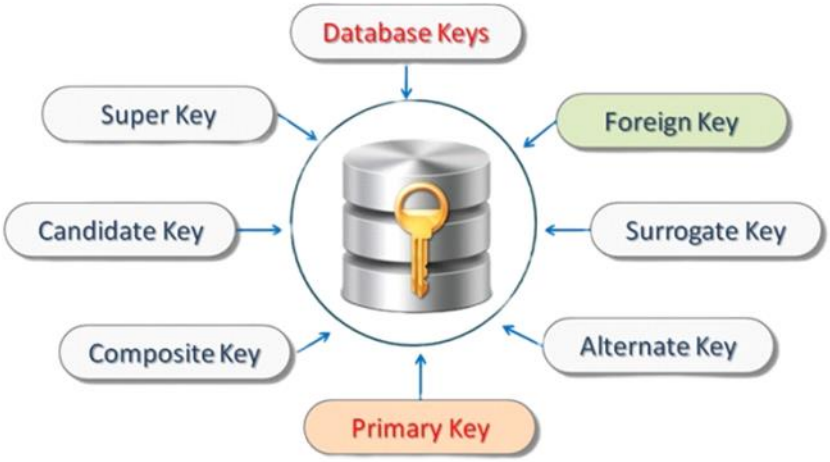
STU_NUM → **Determinant**

STU_NAME → **Dependent**

- Full Functional Dependence
  - Functional dependencies in which the entire collection of attributes in the determinant is necessary for the relationship

(STU_NUM, STU_LNAME) → STU_GPA

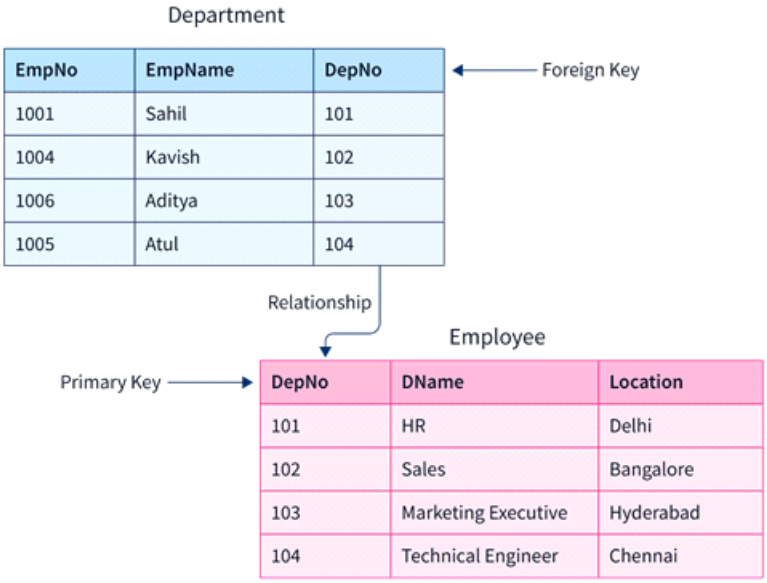
Types of Keys



- Primary Key
  - Candidate key that is most appropriate to be main reference key for the table
  - Rule of Thumb
    - UNIQUE & NOT NULL

StudentId	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042
L0023487	Peter	Murray	P301
L0018453	Anne	Norris	S042

- Foreign Key
  - Attribute or combination of attribute match a primary key in another relation
  - Dependent Table
    - Table in which foreign key is created
  - Parent Table
    - Table to which foreign key refers to



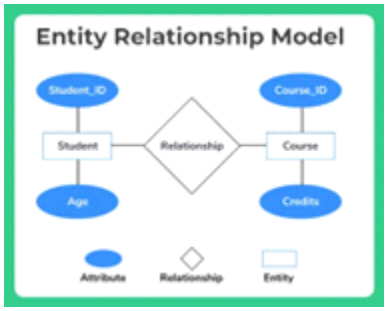
- Candidate Key/Alternate Key
  - Attributes or combination of attributes that are not used as primary key
  - Characteristics
    - Must contain unique values
    - It must not contain null values
    - Contains minimum number of fields to ensure uniqueness
    - Must uniquely identify each record
- Super Key
  - Key that can uniquely identify any row in the table
  - Remember: not all keys are super keys
  - **STU_UNITS → STU_LEVEL**
- Composite Key
  - Composed of more than one attribute
  - Attribute that is part of a key is called a key attribute
  - STU_NUM → STU_GPA
  - (STU_LNAME, STU_FNAME, STU_INIT) → STU_HRS

Entity-Relationship Model

Sunday, 8 October 2023 8:35 pm

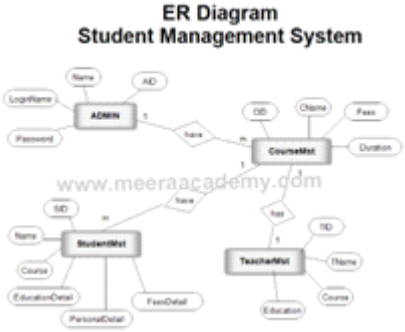
The Entity-Relationship Model

- Conceptual framework used in database design
- Describe the structure of a database
- Focuses on representing entities (objects, concepts, things) and the relationships between them



ERD (Entity-Relationship Diagram)

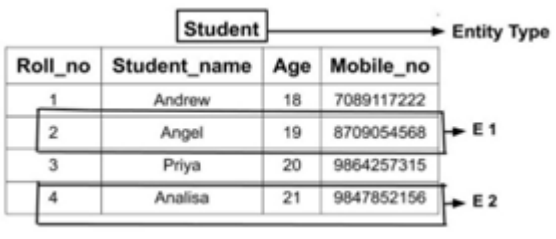
- Visual representation of the Entity-Relationship model
- Diagrammatic way to illustrate entities, attributes and relationships within a database system

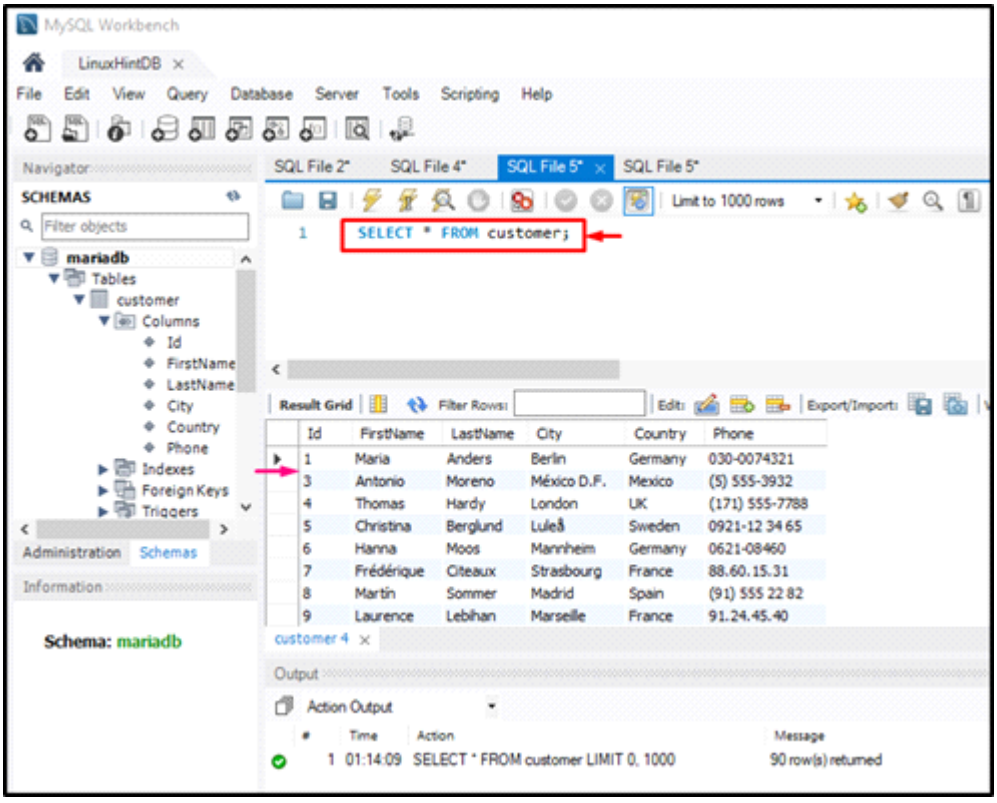


Entity Relationship Model

- ER Model
- Approach to data modeling
- Proposed by P. Chen in 1976
- Says that you divide your database into two logical parts
  - Entities (customer, product)
  - Relations (buys, pays for)

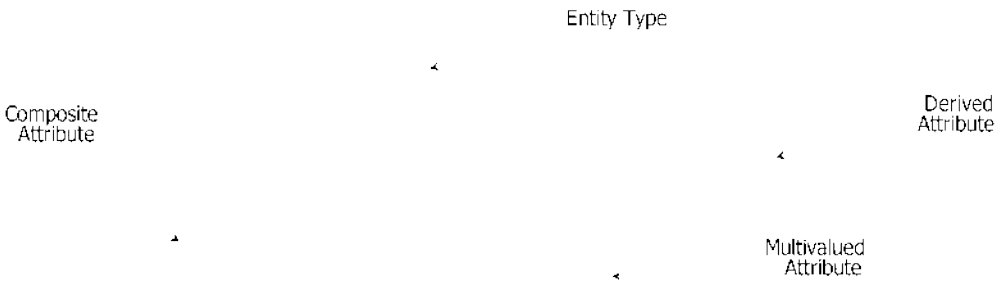
Entity Relationship Modelling

- Entity
  - 
  - Collection of things that share common properties or characteristics
  - In database, anything about which information can be stored
  - For example, person, concept, physical object or event
  - Typically refers to a record structure
  - 



Entity Relationship Model

- Entity Instances
  - A single occurrence of entity
  - There can be many instances of data stored in a database
- Attribute
  - 
  - Properties or characteristics of an entity
- Types of Attributes
  - Composite
    - Can be divided into simple attributes
    - Address, Street , Barangay, City
  - Simple or Atomic
    - Indivisible
    - Age
  - Single-valued
    - Single value attribute
    - Age=27
  - Multi-Valued
    - Attribute can have multiple values
    - College Degree = BSCS, BSBA, BEED
  - Derived
    - Value of an attribute is derived from another attribute
    - Age is derived from Birthday and Current Date
      - Birthday and Current Date are related attributes

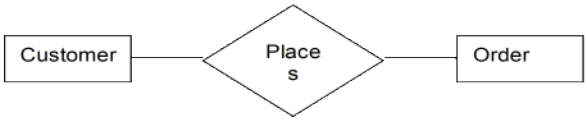


Relationship

- Symbol
- 
- Association between entities
- Tells about some connection between entities

Relationship (*- symbol for relationship* **)**

- association between entities
- tells about some connection between entities



- PLACES is a relationship
  - ✓ labeled with verb phrases
- Relationships are shown in a diamond
- The diagram can be read in both directions

  - Places is a relationship
  - Labeled with verb phrases
- Relationship are shown in a diamond
- Diagram can be read in both directions

Types of Relationship

- One-to-one relationship (1:1)

1. **one-to-one relationship (1:1)**

In a company, each division is managed by only one manager and each manager manages only one division

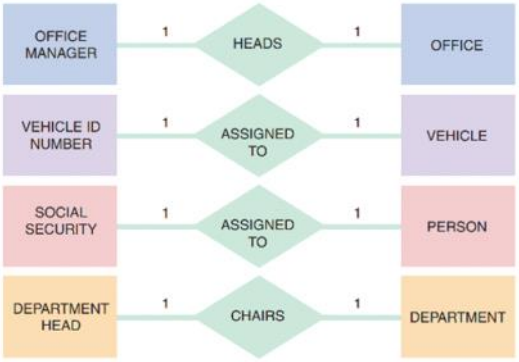
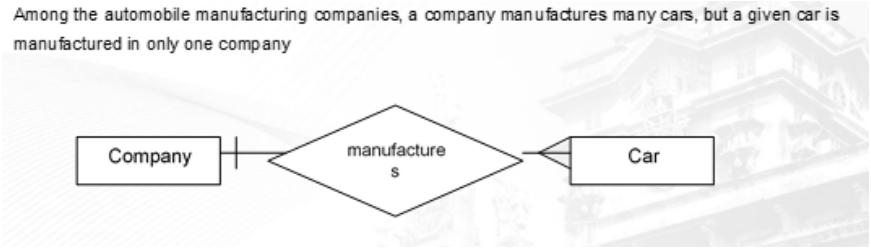


FIGURE 9-13 Examples of one-to-one (1:1) relationships.

- One-to-many relationship (1:N)



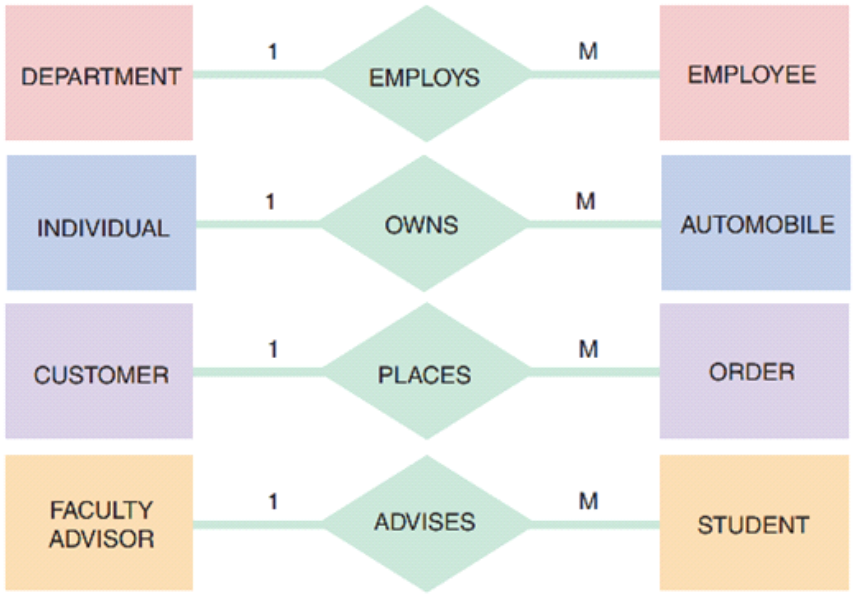
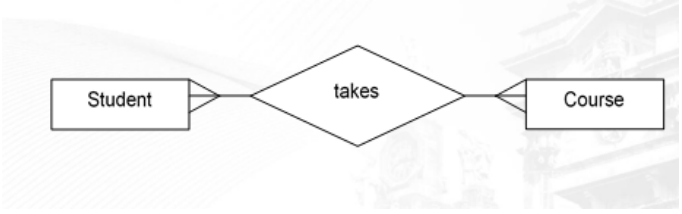


FIGURE 9-14 Examples of one-to-many (1:M) relationships.

- Many-to-many relationship



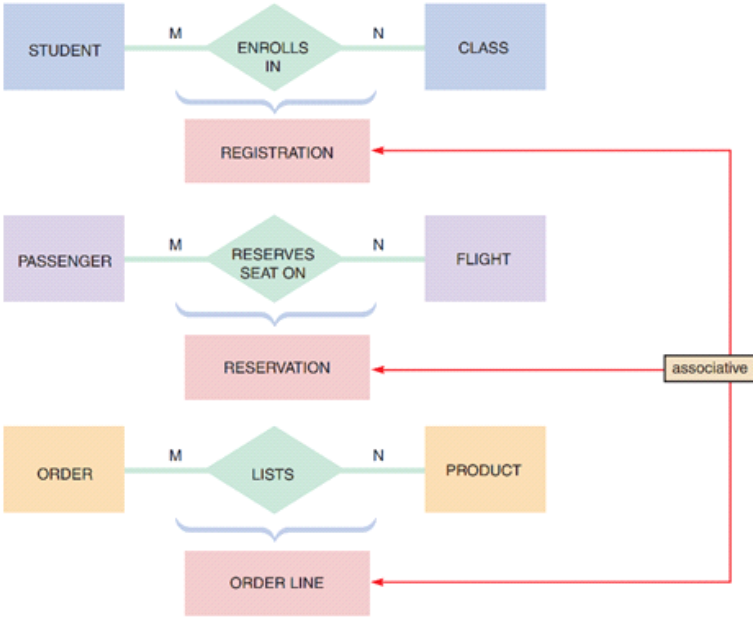
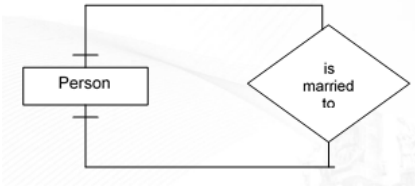
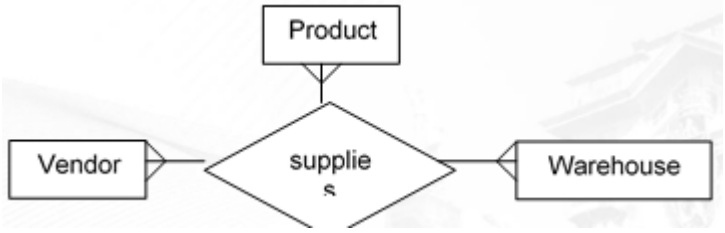
- 

FIGURE 9-15 Examples of many-to-many (M:N) relationships. Notice that the event or transaction that links the two entities is an associative entry with its own set of attributes and characteristics.

Degree of Relationship

- Unary
  - Relationship between instances of single entity type
  - Also called as recursive relationship
  - 

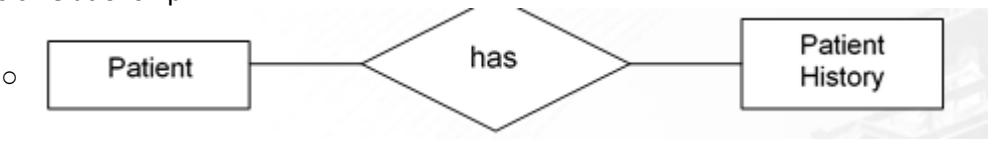
- Binary
  - Relationship between instances of two entity types
  - Most common type of relationship encountered in data modeling
  - 

- Ternary
  - Simultaneous relationship among instances of three entity types
  - 

  - One instance of supplies might record the fact that vendor x can supply to product C to warehouse Y

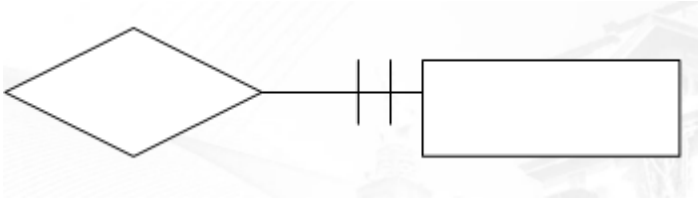
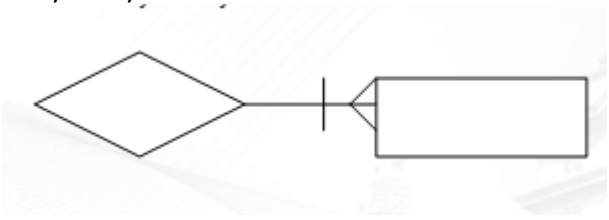
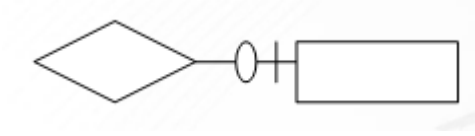
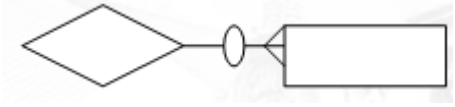
Cardinality Constraints

- Minimum Cardinality
  - Minimum number of instances of one entity that may be associated with each instance of another entity
- Maximum Cardinality

    - Maximum number of instances of one entity that may be associated with single occurrence of another entity
- Basic Relationship
    - 

- Relationship with cardinality constraints
    - 

Relationship Cardinality

- Mandatory One
    - 
- Mandatory Many
    - 
- Optional One
    - 
- Optional Many
    - 

Connectivity and Cardinality

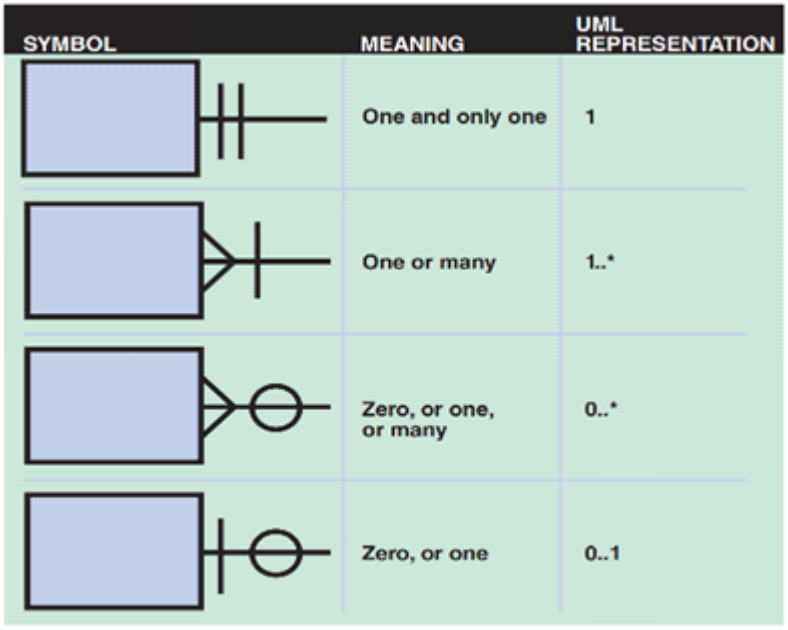
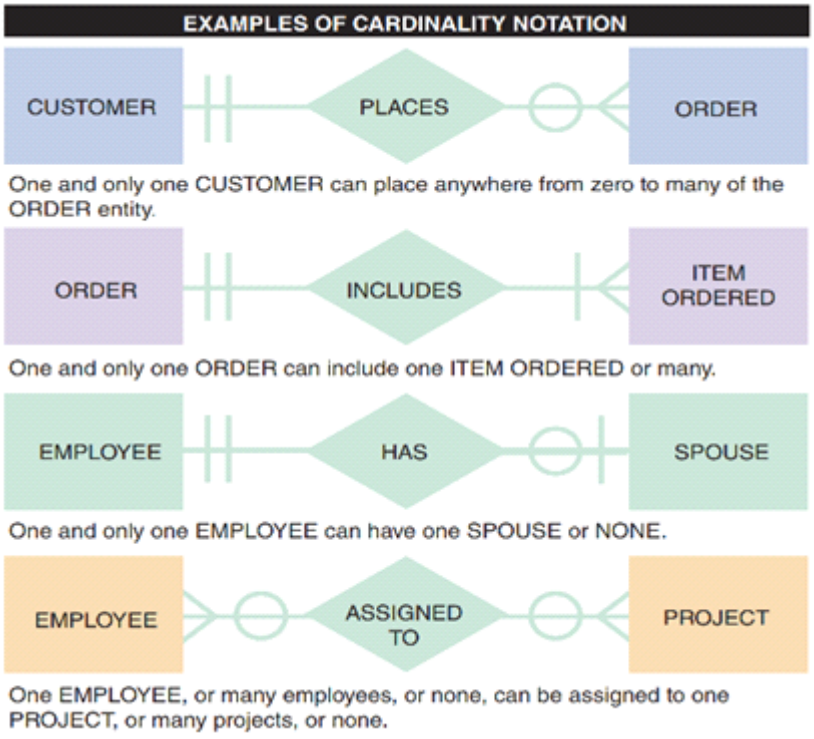


FIGURE 9-17 Crow's foot notation is a common method of indicating cardinality. The four examples show how various symbols can be used to describe the relationships between entities.



Normalization Notes

Monday, 2 October 2023 8:12 am

Normalization Notes
-Process for evaluating and correcting table structures to minimize data redundancies, thereby reducing likelihood of data anomalies
-method of organizing data elements in a database into tables
-summarizing ng columns sa tables to remove anomalies

Need for Normalization
-when designing a new database structure based on the business requirements of the end users, the database designer will construct a data model using technique such as Crow's Foot Notation ERDs
-after initial design is complete, designer can use normalization

Normalization Goals
-Goal is to avoid anomalies

Insertion Anomaly
-Adding new rows forces user to create duplicate data
Deletion Anomaly
-deleting rows may cause loss of data that would be needed for other future rows
Modification Anomaly
-changing data in a row forces changes to other rows because of duplication

Process of Normalization
-

First Normal Form
-term INF describes the tabular format in which
-all of key attributes are defined
-there are no repeating groups in the table, in other words, each row/column in intersection contains one and only one value, not set of values
-all attributes are dependent on the primary key

A TABLE IN FIRST NORMAL FORM

PROJ NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	ASSIGN HOURS
15	Evergreen	103	E. Arbough	Elect. Engineer	$84.50	23.8
15	Evergreen	101	G. News	Dbase Designer	$105.00	19.4
15	Evergreen	105	K. Johnson	Dbase Designer	$105.00	35.7
15	Evergreen	106	W. Smithfield	Programmer	$35.75	12.6
15	Evergreen	102	D. Senior	System Analyst	$96.75	23.8
18	Amber Wave	114	A. Jones	Applications Designer	$48.10	24.6
18	Amber Wave	118	J. Frommer	General Support	$18.36	45.3
18	Amber Wave	104	A. Ramoras	Systems Analyst	$96.75	32.4
18	Amber Wave	112	D. Smithson	DSS Analyst	$45.95	44.0
22	Rolling Tide	105	A. Johnson	Dbase Designer	$105.00	64.7
22	Rolling Tide	104	A. Ramoras	Systems Analyst	$96.75	48.4
22	Rolling Tide	113	D. Joenbrood	Applications Designer	$48.10	23.6
22	Rolling Tide	111	G. Wabash	Clerical Support	$26.87	22.0
22	Rolling Tide	106	W. Smithfield	Programmer	$35.75	12.8
25	Starflight	107	M. Alonzo	Programmer	$35.75	24.6
25	Starflight	115	T. Bawangi	System Analyst	$96.75	45.8
25	Starflight	101	G. News	Dbase Designer	$105.00	56.3
25	Starflight	114	A. Jones	Applications Designer	$48.10	33.1
25	Starflight	108	R. Washington	Systems Analyst	$96.75	23.6
25	Starflight	118	J. Frommer	General Support	$18.36	30.5
25	Starflight	112	D. Smithson	DSS Analyst	$45.95	41.4

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY
1	RAM	9716271721, 9871717178	HARYANA	INDIA
2	RAM	9898297281	PUNJAB	INDIA
3	SURESH		PUNJAB	INDIA

Table 1

Conversion to first normal form

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY
1	RAM	9716271721	HARYANA	INDIA
1	RAM	9871717178	HARYANA	INDIA
2	RAM	9898297281	PUNJAB	INDIA
3	SURESH		PUNJAB	INDIA

Table 2

Second Normal Form (2NF)

- Relation is in first normal form and every non-primary-key attribute is fully functionally dependent on the primary key
- Any partial functional dependencies have been removed
- Functional dependency
  - Constraint between two attributes
  - Relationship between primary key and a key active in the table
  - For every relation R, attribute B is functionally dependent on attribute A
  - Example
    - SSN -> Name, Address, Birthdate
    - VIN -> Make, Model, Color
    - ISBN -> Title, Author_Name
    - Determinants
      - Attribute on the left-hand side of the arrow in a functional dependency (e.g. SSN, VIN, ISBN)
  - Partial Functional Dependency
    - Functional dependency in which one or more non-key attributes are functionally dependent on part (but not all) of the primary key

PROJ_NUM	PROJ_NAME
15	Evergreen
18	Amber Wave
22	Rolling Tide
25	Starflight

EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR
101	G. News	Dbase Designer	$105.00
102	D. Senior	Systems Analyst	$96.75
103	E. Arbough	Elect. Engineer	$84.50
104	A. Ramoras	Systems Analyst	$96.75
105	K. Johnson	Dbase Designer	$105.00
106	W. Smithfield	Programmer	$35.75
107	M. Alonzo	Programmer	$35.75
108	R. Washington	Systems Analyst	$96.75
111	G. Wabash	Clerical Support	$26.87
112	D. Smithson	DSS Analyst	$45.95
113	D. Joenbrood	Applications Designer	$48.10
114	A. Jones	Applications Designer	$48.10
115	T. Bawangi	Systems Analyst	$96.75
118	J. Frommer	General Support	$18.36

PROJ_NUM	EMP_NUM	ASSIGN_HOURS
15	103	23.8
15	101	19.4
15	105	35.7
15	106	12.6
15	102	23.8
18	114	24.6
18	118	45.3
18	104	32.4
18	112	44.0
22	105	64.7
22	104	48.4
22	113	23.6
22	111	22.0
22	106	12.8
25	107	24.6
25	115	45.8
25	101	56.3

PROJ_NUM	EMP_NUM	ASSIGN_HOURS
15	103	23.8
15	101	19.4
15	105	35.7
15	106	12.6
15	102	23.8
18	114	24.6
18	118	45.3
18	104	32.4
18	112	44.0
22	105	64.7
22	104	48.4
22	113	23.6
22	111	22.0
22	106	12.8
25	107	24.6
25	115	45.8
25	101	56.3
25	114	33.1
25	108	23.6
25	118	30.5
25	112	41.4

Third Normal Form

- Transitive dependencies have been removed
- Transitive dependency
    - Functional dependency between two or more non-key attributes
- Creating tables for transitive dependencies

PROJ_NUM	PROJ_NAME
15	Evergreen
18	Amber Wave
22	Rolling Tide
25	Starflight

EMP_NUM	EMP_NAME	JOB_CLASS
101	G. News	Dbase Designer
102	D. Senior	Systems Analyst
103	E. Arbough	Elect. Engineer
104	A. Ramoras	Systems Analyst
105	K. Johnson	Dbase Designer
106	W. Smithfield	Programmer
107	M. Alonzo	Programmer
108	R. Washington	Systems Analyst
111	G. Wabash	Clerical Support
112	D. Smithson	DSS Analyst
113	D. Joenbrood	Applications Designer
114	A. Jones	Applications Designer
115	T. Bawangi	Systems Analyst
118	J. Frommer	General Support

JOB_CLASS	CHG_HOUR
Applications Designer	$48.10
Clerical Support	$26.87
Dbase Designer	$105.00
DSS Analyst	$45.95
Elect. Engineer	$84.50
General Support	$18.36
Programmer	$35.75
Systems Analyst	$96.75

PROJ_NUM	EMP_NUM	ASSIGN_HOURS
15	103	23.8
15	101	19.4
15	105	35.7
15	106	12.6
15	102	23.8
18	114	24.6
18	118	45.3
18	104	32.4
18	112	44.0
22	105	64.7
22	104	48.4
22	113	23.6
22	111	22.0
22	106	12.8
25	107	24.6
25	115	45.8
25	101	56.3
25	114	33.1
25	108	23.6
25	118	30.5
25	112	41.4