BIG DATA: TECHNOLOGIES AND APPLICATIONS

2. Evolution of Database Technologies, Data Model, and Relational Databases

Il-Yeol Song, Ph.D.
College of Computing & Informatics
Drexel University
Philadelphia, PA 19104

DREXEL UNIVERSITY College of Computing & Informatics

Data, Information, and Knowledge

- **Data** are raw facts that are described, observed, and measures
  - Ex: name, height, weight, income, asset
- **Information** is data that has been organized or processed for easy consumption
  - Ex: a list of properties
- **Knowledge** is Information and Rules that are used for actual decision-making
  - Ex: approval/denial of loans

DREXEL UNIVERSITY College of Computing & Informatics

Data, Information, and Knowledge

- Raw data must be *formatted* for storage, processing, and presentation
  - Data must be managed
- Information should be accurate, relevant, and timely for good decision-making
- Knowledge is generalized, discovered, mined, and learned

Database and DBMS

- **Database**: a collection of shared, inter-related , and stored data as well as meta data
  - Data: raw facts of interest
  - **Metadata**: data about data; description about the data
    - Ex: Data size, data source, creation date
- **Database management system (DBMS)**: a collection of programs that define/maintain/manage data
- **Database System**:
  - The DBMS software together with the data itself.
  - Sometimes, the applications are also included.

A Simplified Database System Environment

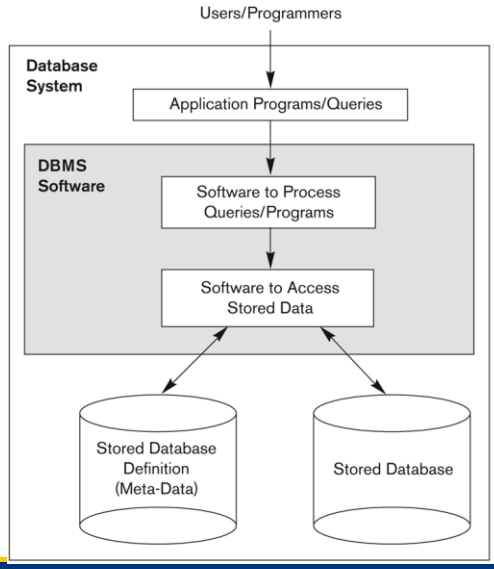
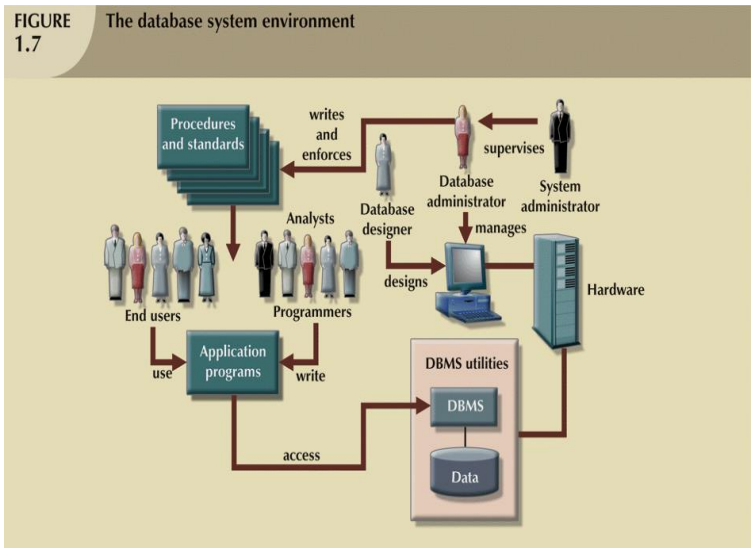


Figure 1.1
A simplified database system environment.

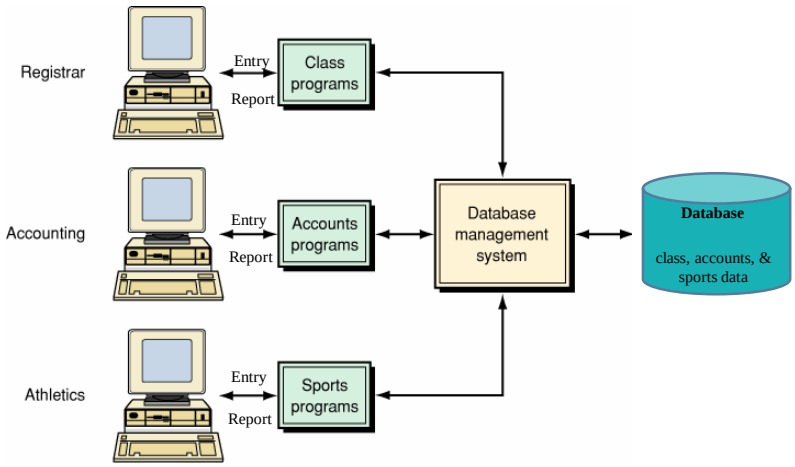
A DBMS needs Multiple People



Advantages of the DBMS

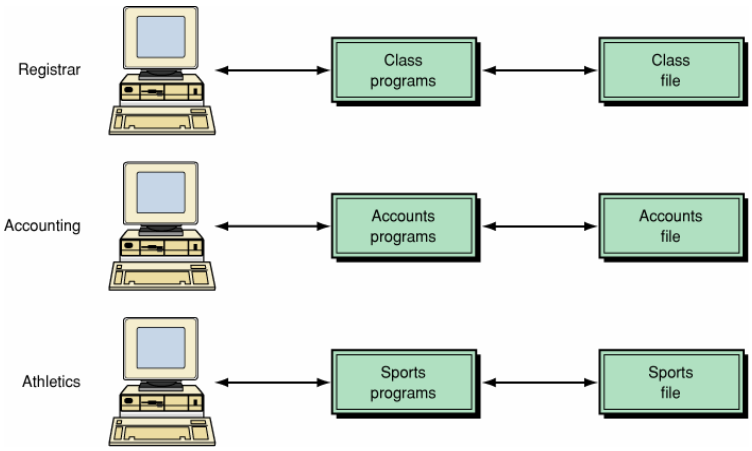
- Improved data sharing
- Improved data security
- Better data integration
- Minimized data inconsistency
- Improved data access
- Improved decision making
- Increased end-user productivity

Database Systems with Multiple Applications



DREXEL UNIVERSITY College of Computing & Informatics

Multiple Applications w/o Database



DBMS Functions

- Store the data
- *Manipulate* data (insert/delete/update)
- Maintain integrity of data
- Process queries, generate reports, optimize processing
- Maintain relationship among data in a data dictionary
- Share data among multiple users
- Handle multiple transactions
- Backup and recover data
- Database access languages and application programming interfaces
- Analyze/mine data

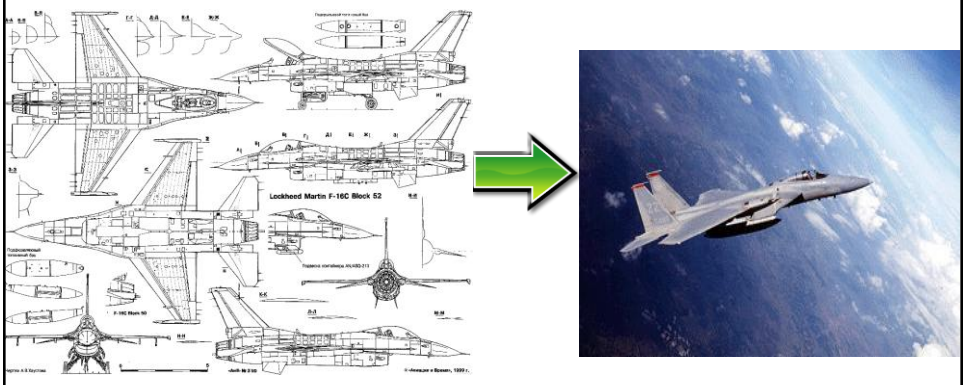
DREXEL UNIVERSITY College of Computing & Informatics

Data Model

- Data model
  - A simple representation of complex real-world data structures
    - Often graphical model is used
- Model: an abstraction of a real-world object or event
  - Useful in understanding complexities of the real-world environment
- Data modeling is iterative and progressive

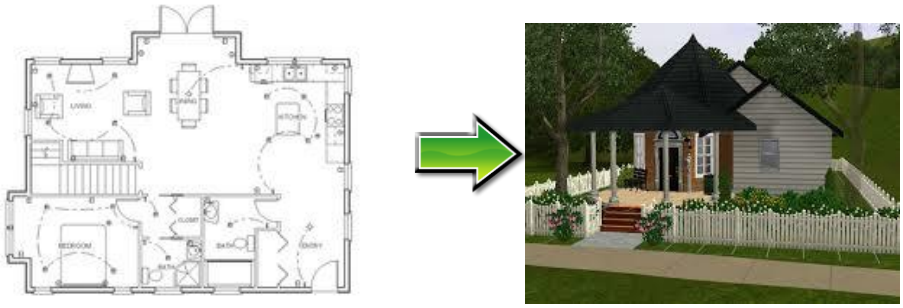
A Model

- A model is a simple representation of a real-world data structures

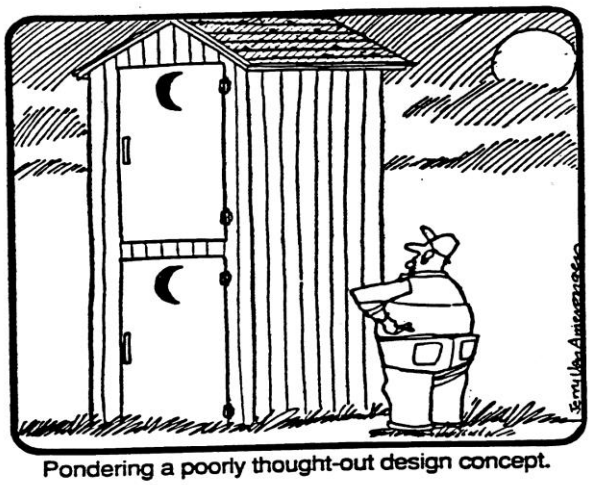


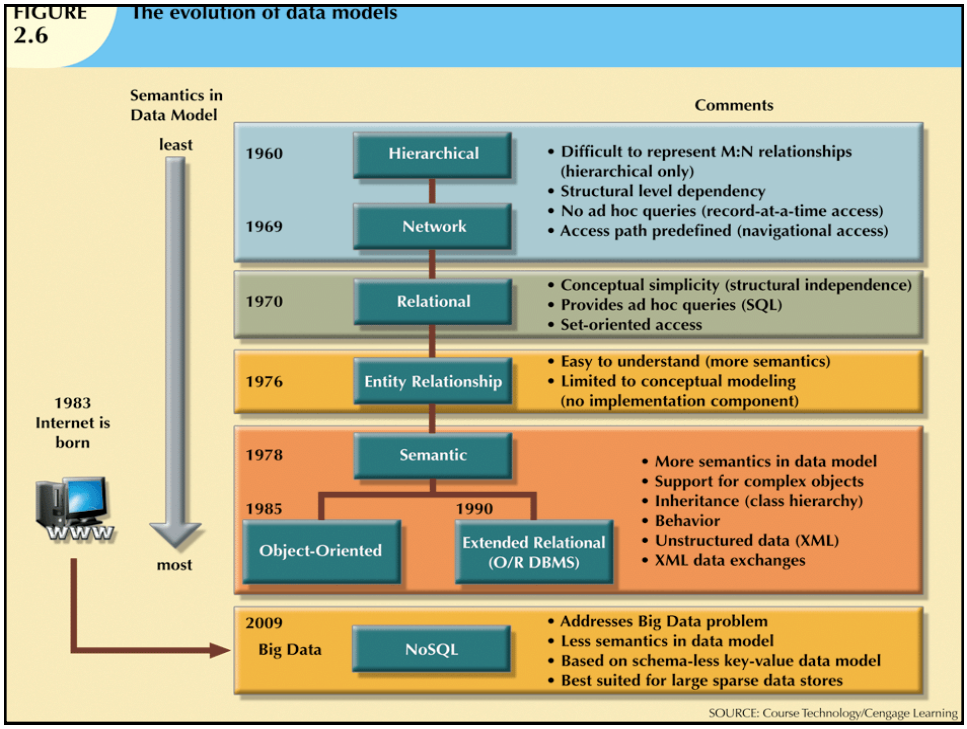
A Model

- A model is a simple representation of a real-world data structures



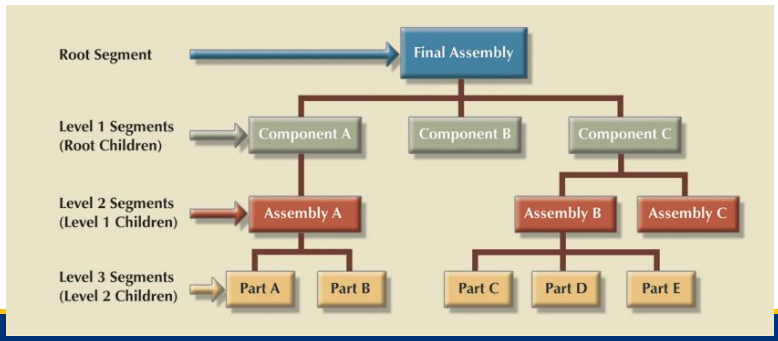
What is Modeling Important?





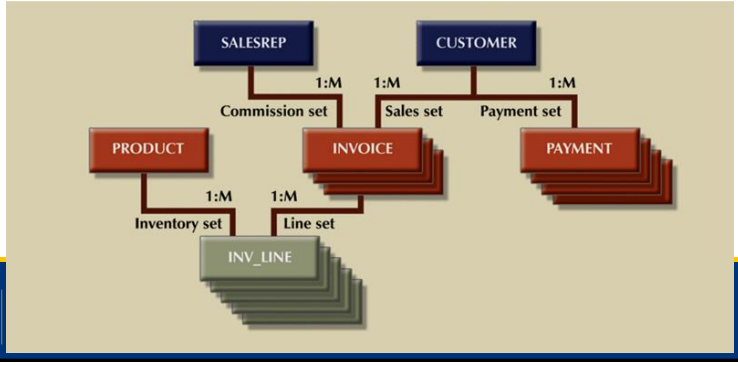
Hierarchical Model

- Data is arranged in an upside-down "tree"
  - Each segment has only one parent
- IMS (Information Management System) by IBM
  - First hierarchical DBMS



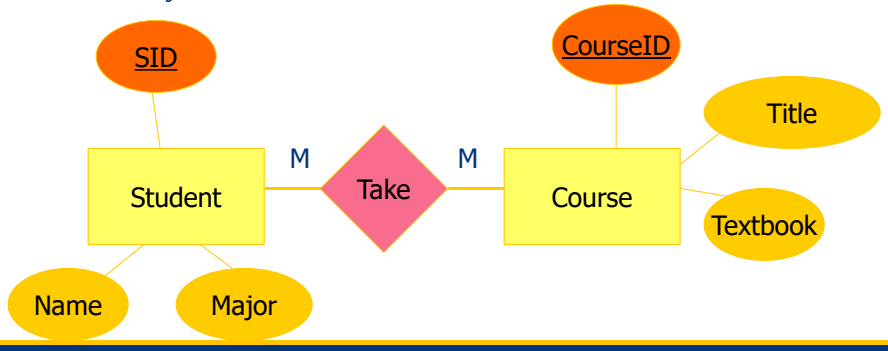
Network Model

- A record can have multiple parents
- Collection of records in 1:M relationships
- Integrated Data Store (IDS) by GE in 1964.
  - First commercial DBMS
  - Invented by Charles Bachman (1973 Turing Award Winner)



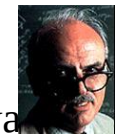
Entity-Relationship (ER) Model

- The most popular database modeling/design model
  - A high-level conceptual model to describe the real world
  - An ER diagram (ERD) represents a set of entities, relationships between entities and their attributes.
- Invented by Peter Chen at MIT in 1976



DREXEL UNIVERSITY College of Computing & Informatics

Relational Model



- Developed by E. F. Codd (1981 Turing Awa Winner) from IBM in 1970
- A relational model contains a collection of related tables (relations)
  - Each column in a relation represents an attribute
  - Each row in a relation is called a tuple
- Allows data to be represented independently of its structure
- More flexible in handling and processing data

Terminology in Relational Model

Relational schema → STUDENT (**stud#,** sname, address, deptno)
DEPARTMENT (**deptno,** dname, chair, phone)

Table (Relation) Name

Attribute Names

Columns

student

stud#	sname	address	deptno
100	John	Philadelphia, PA	10
101	Smith	Norristown, PA	10
102	Borg	Philadelphia, PA	20
103	Jacob	Bryn Mawr, PA	30

PK referencing relation FK

department

deptno	dname	chair	phone
10	Computer Science	Joyce	x.3985
20	Electrical Eng	James	x.6879
30	Physics	Alicia	x.1669

PK referenced relation

Rows (Tuples)

A **Primary key (PK)** uniquely identifies each row.
A **foreign key (FK)** is an attribute that are primary key from another table.

FIGURE 2.3 Linking relational tables

Table name: AGENT (first six attributes) Database name: Ch02_InsureCo

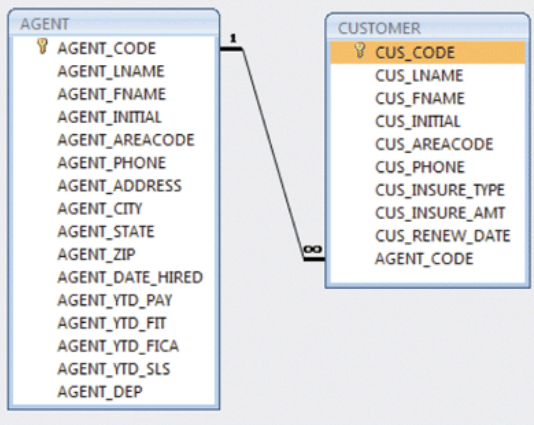
AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

Link through AGENT_CODE

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_INSURE_TYPE	CUS_INSURE_AMT	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	T1	100.00	05-Apr-2008	502
10011	Dunne	Leona	K	713	894-1238	T1	250.00	16-Jun-2008	501
10012	Smith	Kathy	W	615	894-2285	S2	150.00	29-Jan-2009	502
10013	Olowski	Paul	F	615	894-2180	S1	300.00	14-Oct-2008	502
10014	Orlando	Myron		615	222-1672	T1	100.00	28-Dec-2008	501
10015	O'Brian	Amy	B	713	442-3381	T2	850.00	22-Sep-2008	503
10016	Brown	James	G	615	297-1228	S1	120.00	25-Mar-2009	502
10017	Williams	George		615	290-2556	S1	250.00	17-Jul-2008	503
10018	Farriss	Anne	G	713	382-7185	T2	100.00	03-Dec-2008	501
10019	Smith	Olette	K	615	297-3809	S2	500.00	14-Mar-2009	503

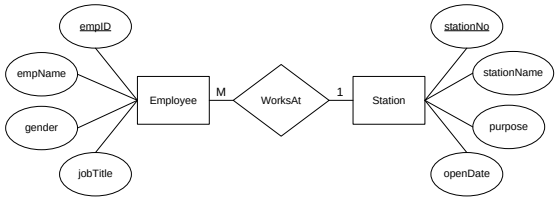
FIGURE 2.2 A relational diagram



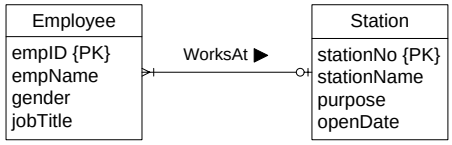
DREXEL UNIVERSITY College of Computing & Informatics

Notations of ER Modeling

- There is no standard notation for ER models
  - Chen



  - Crow's foot



  - UML

Relational Schema

- A logical structure of a database is called a (database) *schema*.
- A logical structure of a relational database is called a ***relational schema***.
  - Always assume each column is ***atomic*** (one value at a time)
- Example: Department relational schema with two tables

  STUDENT (**stud#,** sname, address, deptNO)

  DEPARTMENT (**deptNO,** dname, chair, phone)

  – *Primary key* is underlined

Primary Key (PK)

A **Primary key (PK)** uniquely identifies each row in a table.

- **Uniqueness**: No two instances have the same value in the table.
- Underline the PK

STUDENT (**stud#,** sname, address, deptNO)

DEPARTMENT (**deptNO,** dname, chair, phone)

Exercise: Primary Key

- What is the good PK for the following tables?

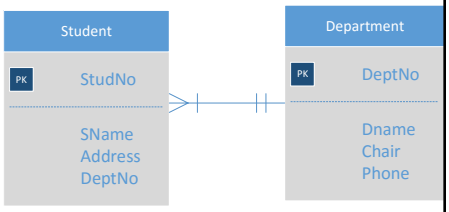
- DOCTOR (SSN, doctorName, license#, specialty, hospitalName, address)
- PRODUCT (productName, brand, price, color, size, barcode)
- ORDER (orderDate, orderAmount, customerNameWhoOrdered, billingAddress, shippingAddress)

Foreign Key (FK)

- An attribute that is the **primary key** in another table.
- Example: Department schema

  STUDENT (**stud#,** sname, address, deptNO)

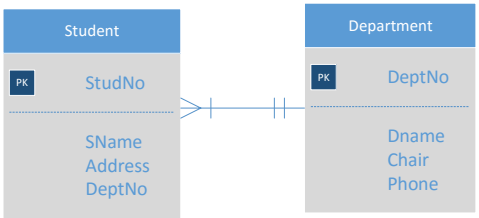
  DEPARTMENT (**deptNO,** dname, chair, phone)
  - Primary key is underlined
  - DeptNO is the foreign key



- Where should we put a FK?

Foreign Key (FK)

- *The Many-side table has the FK, coming from the PK of one-side entity.*
- A FK is used to connect one table to another via *1:N relationship*



Foreign Key & Referential Integrity (RI)

The value of a Foreign Key (FK), if not null, must exist in the original relation as a primary key value.

student

stud#	sname	address	deptno
100	John	Philadelphia, PA	10
101	Smith	Norristown, PA	10
102	Borg	Philadelphia, PA	20
103	Jacob	Bryn Mawr, PA	30

PK FK

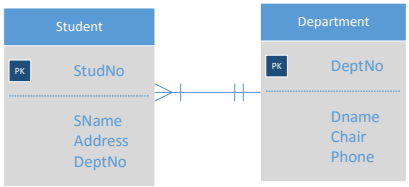
referencing relation

department

deptno	dname	chair	phone
10	Computer Science	Joyce	x.3985
20	Electrical Eng	James	x.6879
30	Physics	Alicia	x.1669

PK

referenced relation



DREXEL UNIVERSITY College of Computing & Informatics

29

What's wrong with the following db?

Student

stud#	sname	address	deptno
100	John	Philadelphia, PA	10
101	Smith	Norristown, PA	10
102	Borg	Philadelphia, PA	20
103	Jacob	Bryn Mawr, PA	40

PK FK

Referencing relation

Department

deptno	dname	chair	phone
10	Compute r Science	Joyce	3985
20	Electrical Eng	James	6879
30	Physics	Alicia	1669

PK

Referenced relation

DREXEL UNIVERSITY College of Computing & Informatics

30

To Which Table must we add a FK?

FIGURE 3.18 The 1:M relationship between PAINTER and PAINTING

The MANY-side table keeps the FK

FIGURE 3.19 The implemented 1:M relationship between PAINTER and PAINTING

Table name: PAINTER
Primary key: PAINTER_NUM
Foreign key: none

Database name: Ch03_Museum

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

Table name: PAINTING
Primary key: PAINTING_NUM
Foreign key: PAINTER_NUM

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

FIGURE 3.28 The expanded entity relationship model

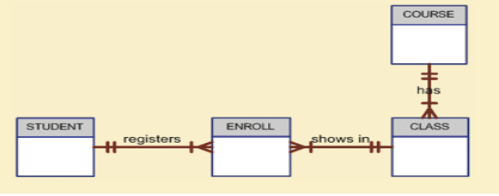
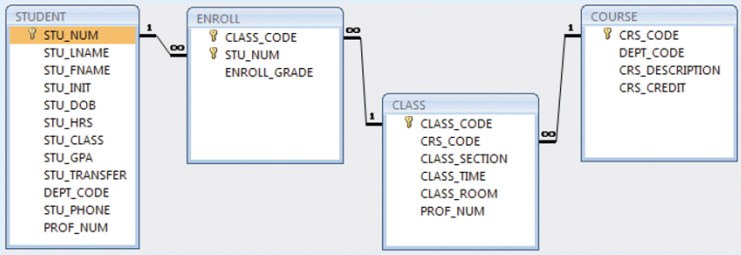


FIGURE 3.29 The relational diagram for the Ch03_TinyCollege database



College of Computing & Informatics

Example of SQL Implementing a Relational Schema

student

stud#	sname	address	deptno
100	John	Philadelphia, PA	10
101	Smith	Norristown, PA	10
102	Borg	Philadelphia, PA	20
103	Jacob	Bryn Mawr, PA	30

referencing relation

department

deptno	dname	chair	phone
10	Computer Science	Joyce	x.3985
20	Electrical Eng	James	x.6879
30	Physics	Alicia	x.1669

referenced relation

```
CREATE TABLE Department (
        DeptNo          NUMBER(5)       PRIMARY KEY,
        DName           VARCHAR2(15)    NOT NULL        ,
        Chair           VARCHAR2 (20)                   ,
        Phone           CHAR (10)                       ,
        );

CREATE TABLE Student (
        Stud#           NUMBER(8)                       ,
        SName           VARCHAR2(20)    NOT NULL        ,
        Address         VARCHAR2 (40)                   ,
        DeptNo          NUMBER (5)                      ,
        CONSTRAINT Stud_PK  PRIMARY KEY (Stud#)         ,
         CONSTRAINT Stud_FK FOREIGN KEY (DeptNo)
                REFERENCES Department (DeptNo)
        );
```

DREXEL UNIVERSITY College of Computing & Informatics

Database Transaction

- Database transaction:
  - A transaction is a set of operations that are treated ***as a logical unit***
  - Must be either entirely completed or aborted
  - No intermediate states are acceptable

- Ex: Transfer $100 from Account A to B:
  - Debit $100 from Account A
  - Credit $100 to Account B

  Note: Performing only one operation is not acceptable Both operations must be executed.

The ACID Properties in RDBMSs

- ***Atomicity*** – A transaction is atomic– it completes entirely or not at all, despite failures.
- ***Consistency*** –Transactions must maintain consistency, despite crashes or concurrent access or power failure.
- ***Isolation*** – transactions must be executed sequentially in some order even if there are multiple concurrent users
- ***Durability*** – Once a transaction has completed, its changes are made permanent even if there are system failures.

Why Relational Database is so Popular?

- RM is the most widely used data model in real-world databases:
  - **Simple structure** – a set of inter-related tables
    - Easy to understand the structure
  - Easy use and high-level DB language: **SQL**
    - Easy to implement
  - RDBMSs support **ACID** properties
    - Reliably use in mission-critical business

Two Most Important Traditional Data Models in Practice

- **Relational Model (E.F. Codd, 1971)**
  - Used for implementation
  - Popular due to simplicity and SQL
  - Most commercial database systems
  - Oracle, IBM DB 2, MS SQL Server, Sybase, Teradata, MS Access
- **Entity-Relationship (ER) Model (P. Chen, 1976)**
  - Used for modeling and design
  - Popular due to easy to learn/use/model/translate into RDB schema

Review

- What is a data model?
- What are the purposes of Entity-relationship (ER) model and relational model?
- What does a relational database look like?
- What is a relational schema?
- What is the primary key?
- What is a foreign key?
- What is the referential integrity?
- What is a database transaction
- What is the significance of ACID properties?
- Why relational database is the most widely used?

Points to Think about Relational Database

- Think about a database application you may want to develop.
- What are the limitations of storing big data to relational databases?
  - How RDB can handle Volume?
  - How RDB can handle Velocity?
  - How RDB can handle Variety?
  - How RDB can handle Veracity?