CSC 174: Advanced Database Management Systems

Chapter 1

Databases and Database Users

Amarjot Biring

Introduction

- Traditional database applications
  - Numeric and Textual Databases
- Big data storage systems
  - Multimedia Databases
  - Geographic Information Systems (GIS)
  - Biological and Genome Databases
  - **Data Warehouses**
    - Data analytics
  - Mobile databases
  - Real-time and Active Databases

Introduction Continued

- Social Networks
  - Started capturing a lot of information
    - Facebook, Twitter (X), LinkedIn
- Search Engines
  - Google, Bing, Yahoo
  - Collect their own repository of web pages for searching purposes
- All the above constitutes data

Basic Definitions

- Data
  - Known facts that can be recorded and have an implicit meaning.
- Database
  - A collection of related data.
- Miniworld
  - A database represents some aspect of the real world.
    - For example, student grades and transcripts at a university.

Basic Definitions Continued

- Database Management System (DBMS)
  - A software package/ system to facilitate the creation and maintenance of a computerized database.
  - SQL Server, Oracle, MySQL, Access, MongoDB
  - Facilitates defining, constructing, manipulating, and sharing

- Database System
  - The DBMS software together with the data itself.

Databases

- Represents some aspect of the real world
- Logically coherent collection of data with some inherent meaning
  - A random assortment of data cannot correctly be referred to as a database
- Designed, built, and populated with data for a specific purpose
  - Has an intended group of users and some preconceived applications in which these users are interested

Databases Continued

- End users of a database may perform business transactions
  - A customer buys a camera
  - Events may happen
    - An employee has a baby
  - Information in the database is changed.

- Database accuracy and reliability
  - Depends on true reflection of the miniworld.
  - Changes must be reflected in the database as soon as possible

- Examples?

Database Management System (DBMS)

- Defining
  - Specifying the data types, structures, and constraints of the data to be stored in the database.
  - Database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary; it is called meta-data.

- Constructing
  - Process of storing the data on some storage medium that is controlled by the DBMS.

Database Management System (DBMS)

- Manipulating
  - Functions such as querying the database to retrieve specific data
  - Updating the database to reflect changes in the miniworld
  - Generating reports from the data

- Sharing
  - Allow multiple users and programs to access the database simultaneously.

Internet of Things (IoT)

- Computers and networks being built into everyday things*
  - Cars
  - Mattresses
  - Refrigerators
  - Thermostats
- Smart devices connected to each to create a Network of Things (NoT)
- Over 23 billion IoT devices today
  - Could grow to 75 billion by 2025

Big Data

- 2.5 quintillion bytes (2.5 exabytes) of data created daily
- 90% of the worlds data created in last five years
- By 2025, 175 zettabytes of data
  - 175 trillion gigabytes
  - 175 billion terabytes
- Context…
  - Megabytes (MB) – smartphone video = 350 MB
  - Gigabytes (GB) = 1,000 MB (2.85-minute smartphone video)
  - Terabytes (TB) = 1,000 GB (84 hours of video)
  - Petabytes, Exabytes, and Zettabytes = 5.5 million hours of video

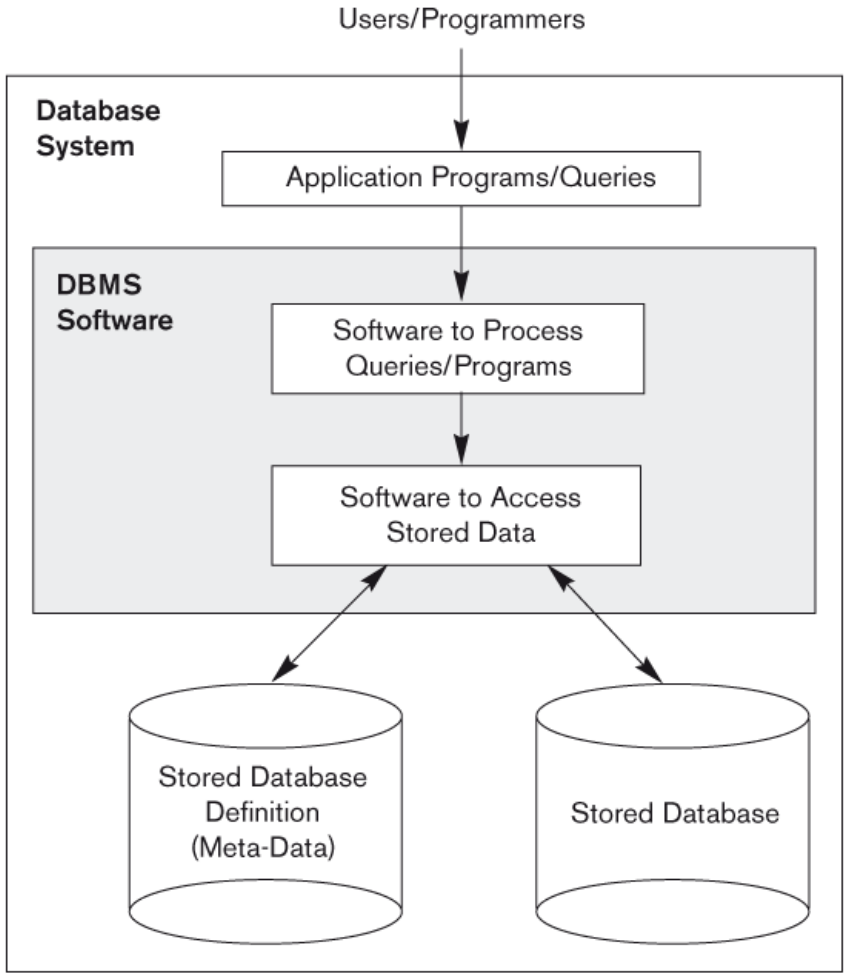
Big Data Continued

- Facebook
  - 1.49 billion active daily users
    - 6 new profiles every second
  - 250 billion photos uploaded (350 million/day) – 2019
- YouTube
  - 300 hours of video uploaded every minute
  - 5 billion videos watched every day
- Twitter
  - 350,000 tweets sent per minute (200 billion tweets per year)
- Google
  - 3.8 million searches per minute

Internet and World Wide Web

- Cloud
  - More and more computing today is done “in the cloud”—that is, distributed across the Internet worldwide
  - Many apps you use daily are dependent on cloud-based services
    - Use massive clusters of computing resources (computers, processors, memory, disk drives, etc.)
    - Databases that communicate over the Internet with each other and the apps you use
  - Huge data centers
    - Data Center video
  - What are some cloud services you use today?

Simplified Database System Environment



Example of a Database

- Mini-world for the example
  - University environment

- Some mini-world entities
  - **Student**s
  - **Course**s
  - **Section**s
  - **Department**s
  - **Instructor**s

Example of a Database Continued

- Some mini-world relationships
  - **Instructor**s teach **Course**s
  - **Course**s have prerequisite **Course**s
  - **Course**s have **Sections**
  - **Course**s belong to **Department**s
  - **Student**s enroll in **Section**s
  - **Student**s major in **Department**s

Example of a Database Continued

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

Database Users

- Database administrators
  - Authorize access to the database
  - Coordinate and monitor its use
  - Acquire software and hardware resources
  - Controlling database use and monitoring efficiency of operations
  - Responsible for security breaches and poor response time

- Database Designers
  - Define the content, structure, constraints, and functions or transactions against the database.
  - Must communicate with the end-users and understand their needs.

Database Users Continued

- End Users
  - Use the data for queries, reports and some of them update the database content.
  - Casual
    - Access database occasionally when needed
  - Naïve or Parametric
    - Make up a large section of the end-user population.
    - Use previously well-defined functions in the form of “canned transactions” against the database.
    - Users of Mobile Apps mostly fall in this category
    - Bank-tellers or reservation clerks are parametric users who do this activity for an entire shift of operations.
    - Social Media Users post and read information from websites

Database Users Continued

- End Users
  - Sophisticated
    - Business analysts, scientists, engineers, others thoroughly familiar with the system capabilities.
    - Many use tools in the form of software packages that work closely with the stored database.
  - Stand-alone
    - Mostly maintain personal databases using ready-to-use packaged applications.
    - Examples
      - User of a tax program that creates its own internal database.
      - User that maintains a database of personal photos and videos.

Database Users Continued

- End Users
  - System Analysts and Application Developers (large portion of IT worksforce)
    - System Analysts
      - Understand the user requirements of naïve and sophisticated users
      - Design applications including canned transactions to meet those requirements.
    - Application Programmers
      - Implement the specifications developed by analysts and test and debug them before deployment.
    - Business Analysts
      - Increasing need for such people who can analyze vast amounts of business data and real-time data (“Big Data”) for better decision making related to planning, advertising, marketing etc.

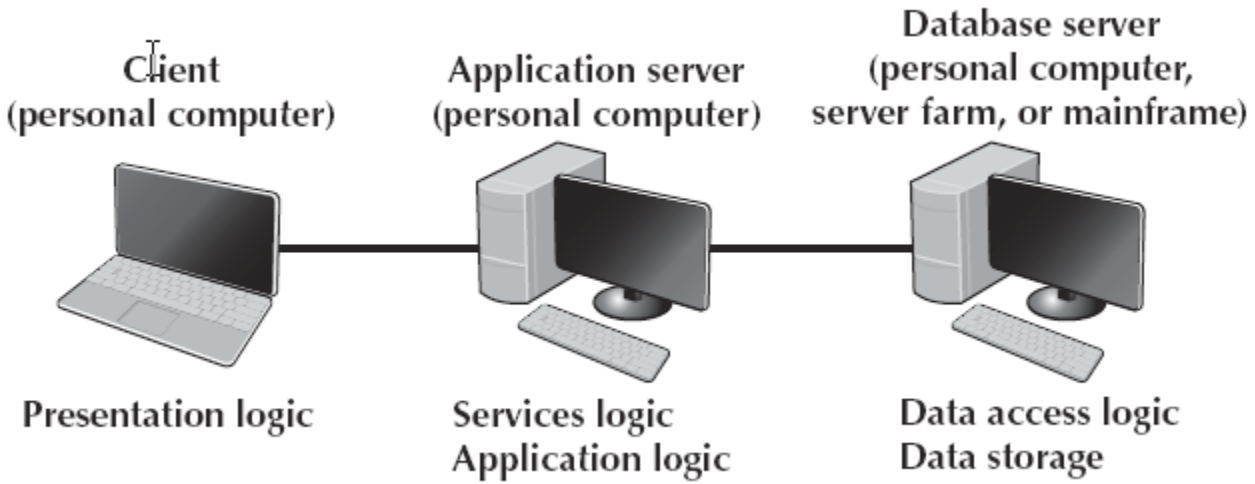
Database Considerations

- Controlling redundancy
  - Sharing of data among multiple users
- Securing access to data
- Efficient query processing
  - Indexes
- Enforcing integrity constraints
- Enforcing standards
  - Example
- Flexibility to change data structures
- Availability of up-to-date information

Three Tier Client-Server Architecture

- Client, Application/Web Server, and Database Server
- Common for Web applications
- The client is typically a PC or a mobile device connected to the Web
- Intermediate Layer called Application Server or Web Server
  - Business logic part of the application used to access the corresponding data from the database server
  - Conduit for sending data between the database server and the client.
- Database server only accessible via middle/intermediate layer
- Clients cannot directly access database server
- Clients contain user interfaces and Web browsers

Three Tier Client-Server Architecture Continued



CSC 174: Advanced Database Management Systems

Chapter 3

Data Modeling Using the Entity-Relationship Model

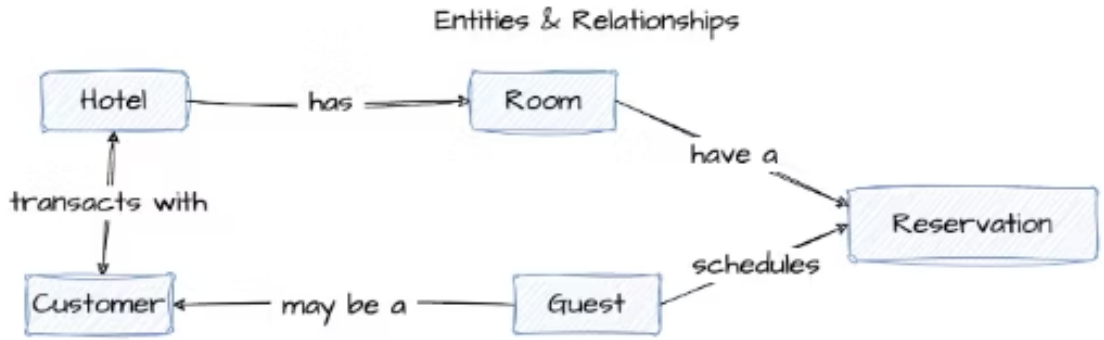
Amarjot Biring

Introduction

- Conceptual Data Model
  - Use entities and relationships
  - Very high-level
  - Entity – real-world object (employee, project)
  - Relationship – association among the entities

- Entity-Relationship (ER) model
  - High level conceptual data model
  - Used for conceptual design of database applications
  - Entities, Attributes, and Keys
    - Attribute – some property of interest that further describes an entity
      - No data types

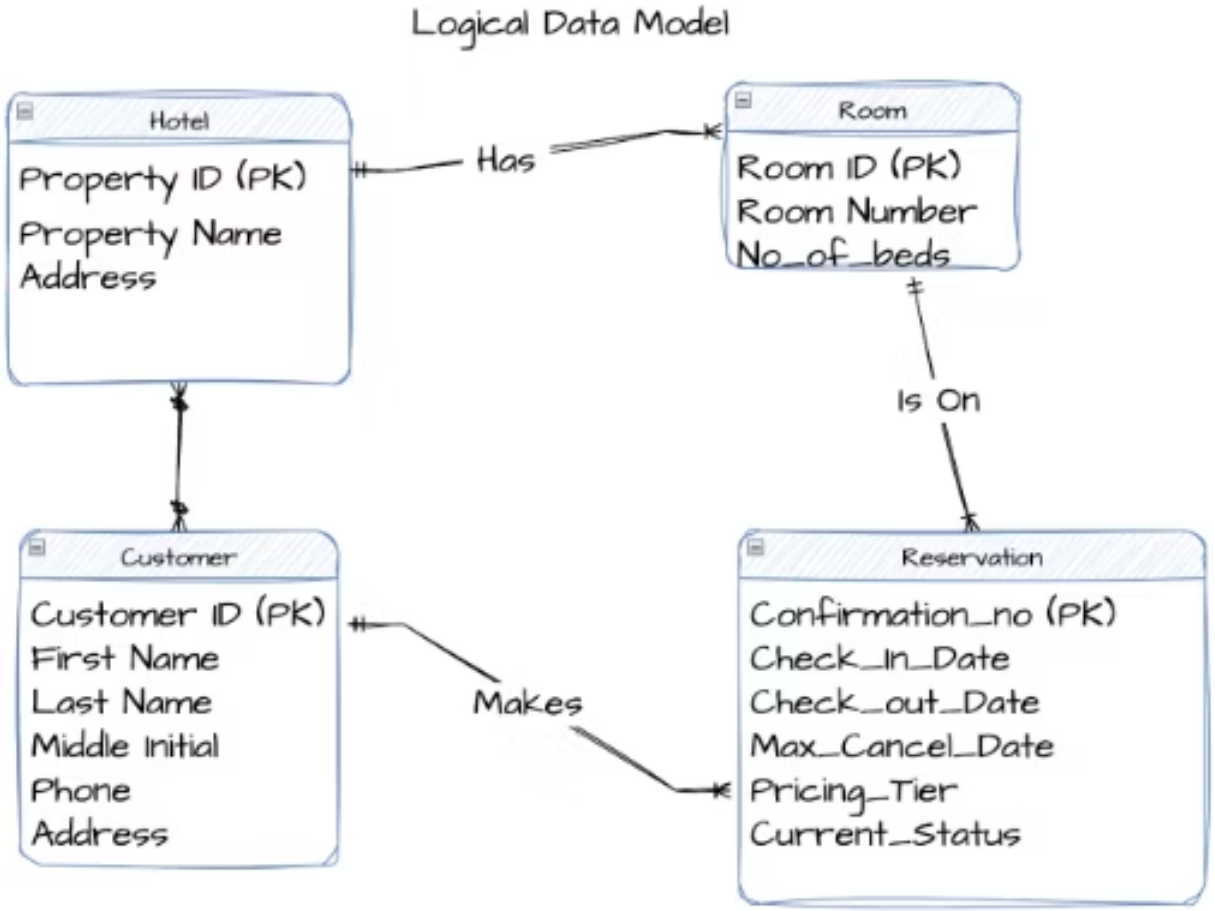
Introduction Continued



Definitions

Entity	Description	Relationships
Hotel	Physical location of the hotel with is key in identify a hotel. Hotels have many guest rooms. Reservations is for guest not conference rooms (not shown here).	Must have many rooms May have many customers
Room	Rooms are a sub-component of a hotel. They cannot exist outside of the hotel. Rooms are reserved by a guest.	Reserved by many guests
Guest	A guest reserves are room at the hotel. They can also be a customer. The difference between a guest and customer is that customers don't necessarily have to reserve a room to transact with the hotel.	Reserves a room Can be a customer
Customer	A customer transact business with the hotel but is not required to be a guest. Customers could purchase from the gift shop.	Transacts with the hotel
Reservation	A reservation reserves a room from the inventory of available rooms for a guest. It is uniquely identified by a confirmation number. Guest can make multiple reservations	Reserves a room for a guest.

Introduction Continued



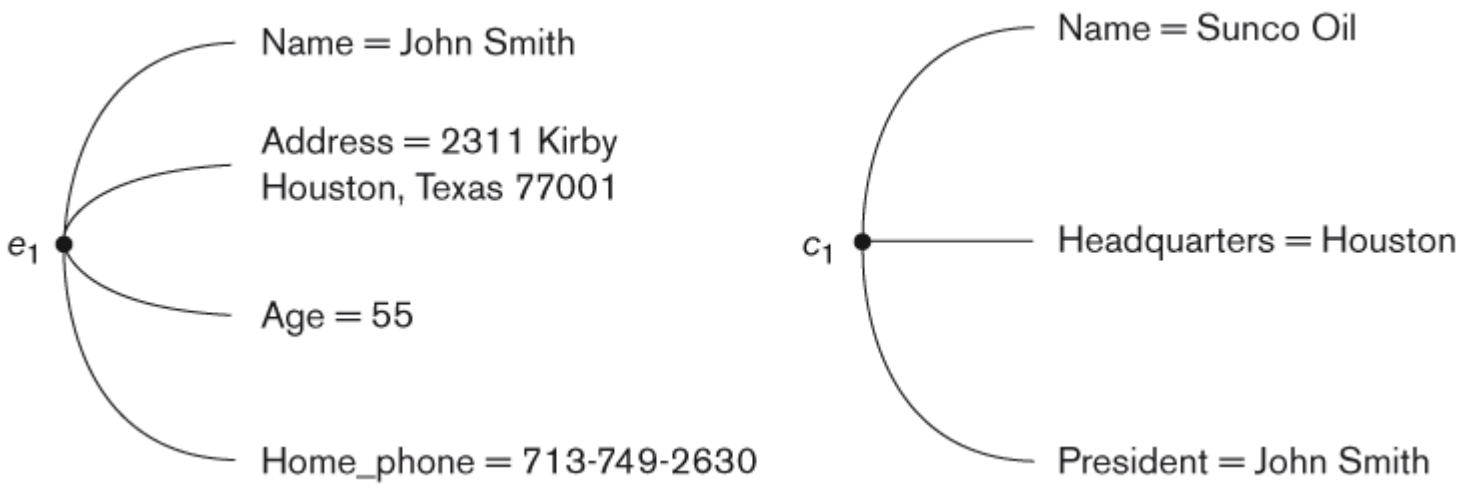
Entity

- A thing or object in the real world with an independent existence.
- May be an object with a physical existence
  - Person, car, house, or employee
- May be an object with a conceptual existence
  - Company, a job, or a university course
- Has Attributes

Attributes

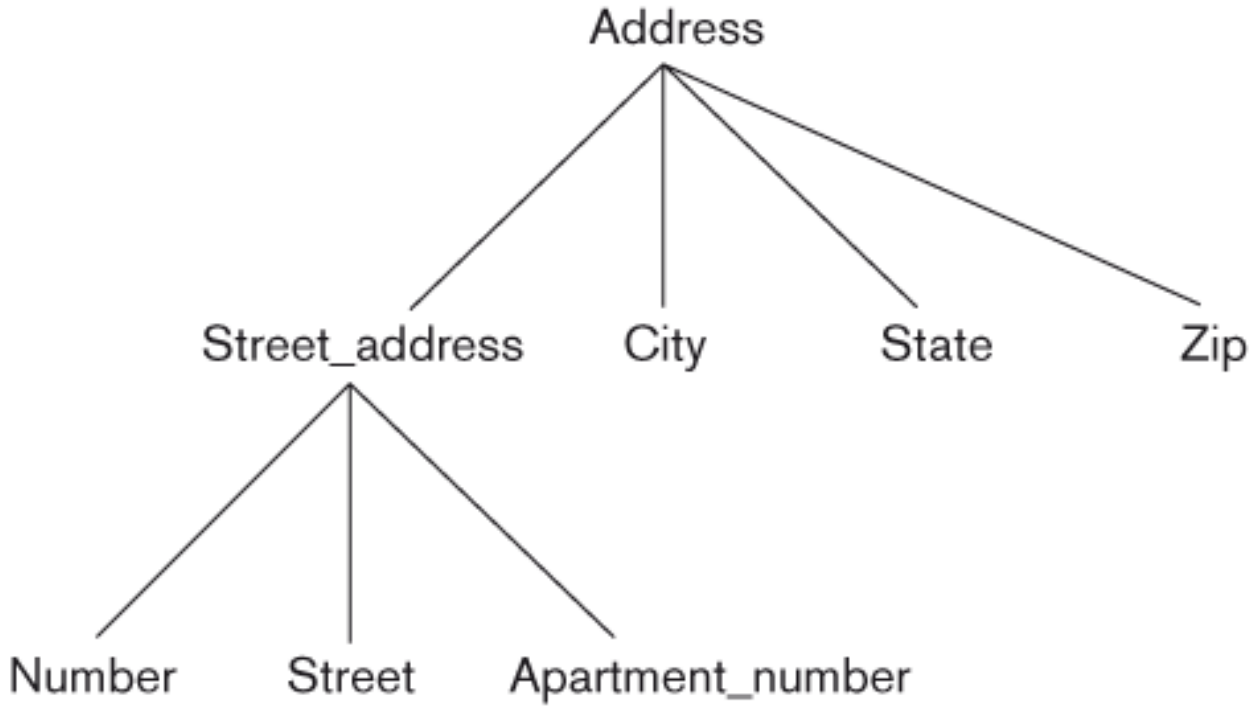
- Properties that describe an entity.
  - An EMPLOYEE entity may be described by Employee's
    - Name
    - Age
    - Address
    - Salary
    - Job.
- A particular entity will have a value for each of its attributes.
- The attribute values that describe each entity become a major part of the data stored in the database.

Entity and Attributes Example



Composite Attributes

- Can be divided into smaller subparts
- Attributes not divisible are called simple or atomic attributes



Single-Valued and Multivalued Attributes

- Single-valued
  - Attributes with one value for an entity
    - Entity → Person
    - Single-valued attribute → Age

- Multi-valued
  - Attributes that can have more than one value
    - Entity → Person
    - Multi-valued attribute → College degrees

Stored and Derived Attributes

- Derived
  - Attribute is derived/calculated from another attribute
    - Age is derived/calculated from BirthDate

- Stored
  - BirthDate is a stored attribute
  - A derived attribute is derivable from a stored attribute

NULL Values

- Attribute is not applicable
  - ApartmentNumber of an Address
- Attribute may not be known
  - HomePhoneNumber of a Person

Keys

- Uniqueness constraint on attributes
- An entity has one or more attributes whose values are distinct for each record in the entity
  - This attribute is called a **key attribute**
  - Values can be used to identify each record uniquely
- An entity with no key is a weak entity

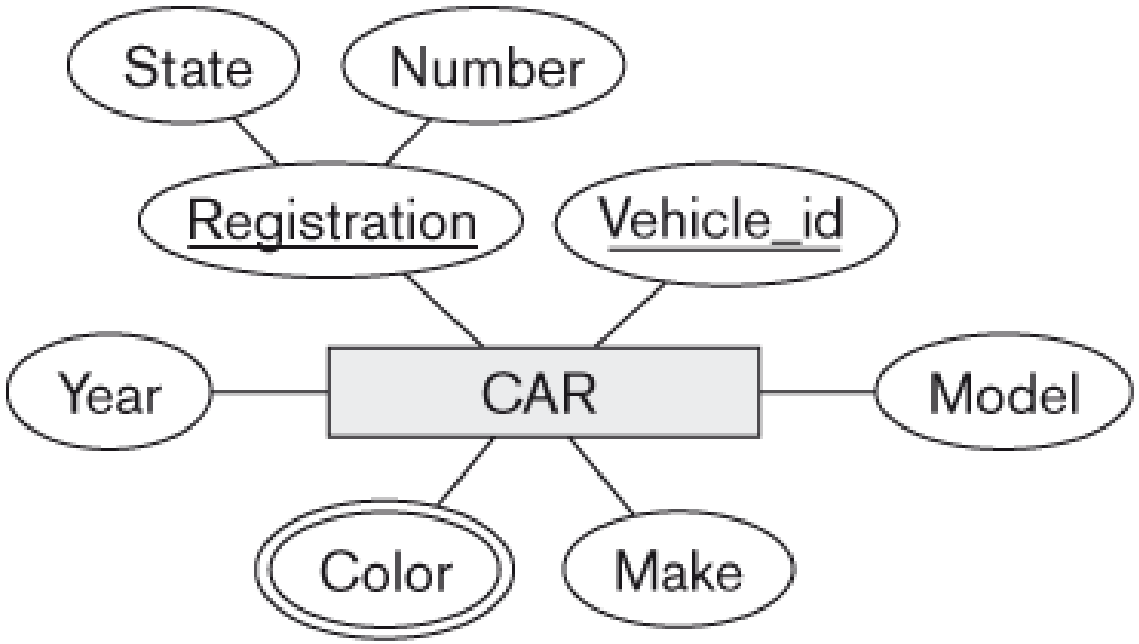
- Name attribute is a key of the COMPANY entity
  - No two companies are allowed to have the same name
- SSN is a key of the PERSON entity
  - SSN is unique for each person

ER Diagram Shapes

- Entity – entity names enclosed in rectangular box
- Attribute – attribute names enclosed in oval
- Composite Attribute – attached to their attributes with straight line
- Multivalued Attribute – displayed in double ovals
- Key – attribute is underlined

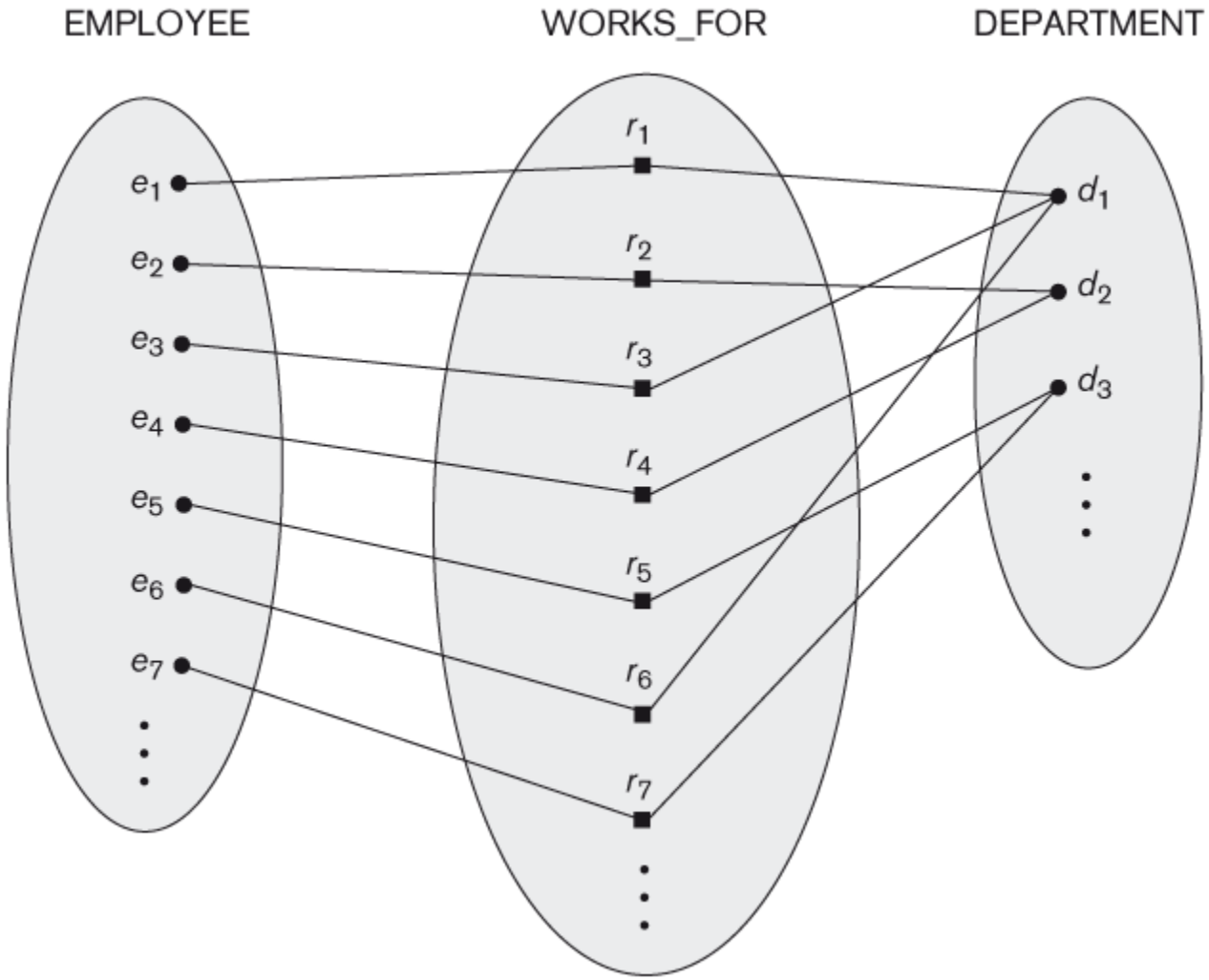
ER Diagram Shapes Continued

- Entity – entity names enclosed in rectangular box
- Attribute – attribute names enclosed in oval
- Composite Attribute – attached to their attributes with straight line
- Multivalued Attribute – displayed in double ovals
- Key – attribute is underlined



Relationships

- Sets of associations among entities
- Association includes exactly one record from each participating entity
- Each such relationship represents the fact that the entities participating in the relationship are related in some way in the corresponding miniworld situation



ER Diagram Shapes

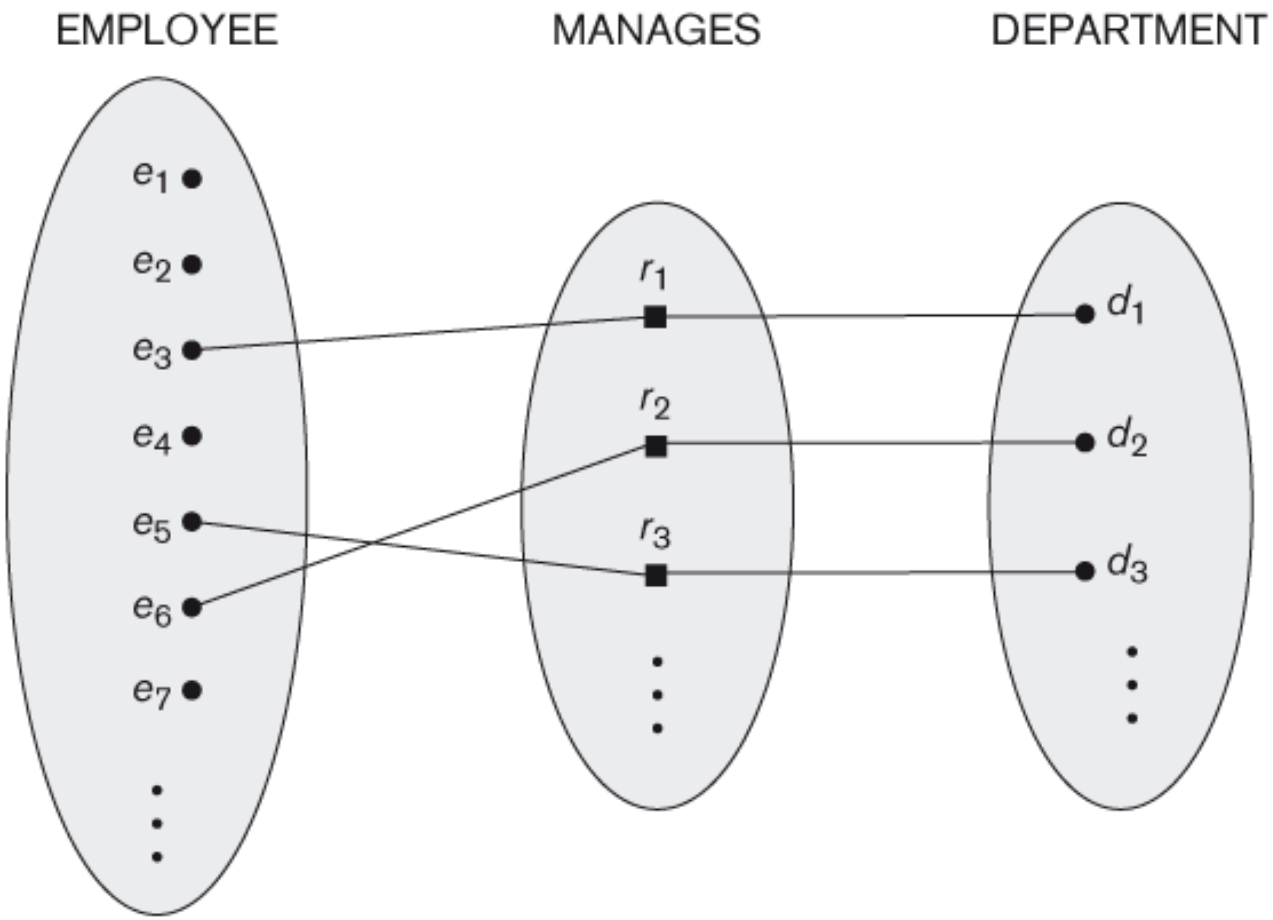
- Entity – entity names enclosed in rectangular box
- Attribute – attribute names enclosed in oval
- Composite Attribute – attached to their attributes with straight line
- Multivalued Attribute – displayed in double ovals
- Key – attribute is underlined
- **Relationships – diamond-shaped boxes**

Relationship Cardinality Ratios

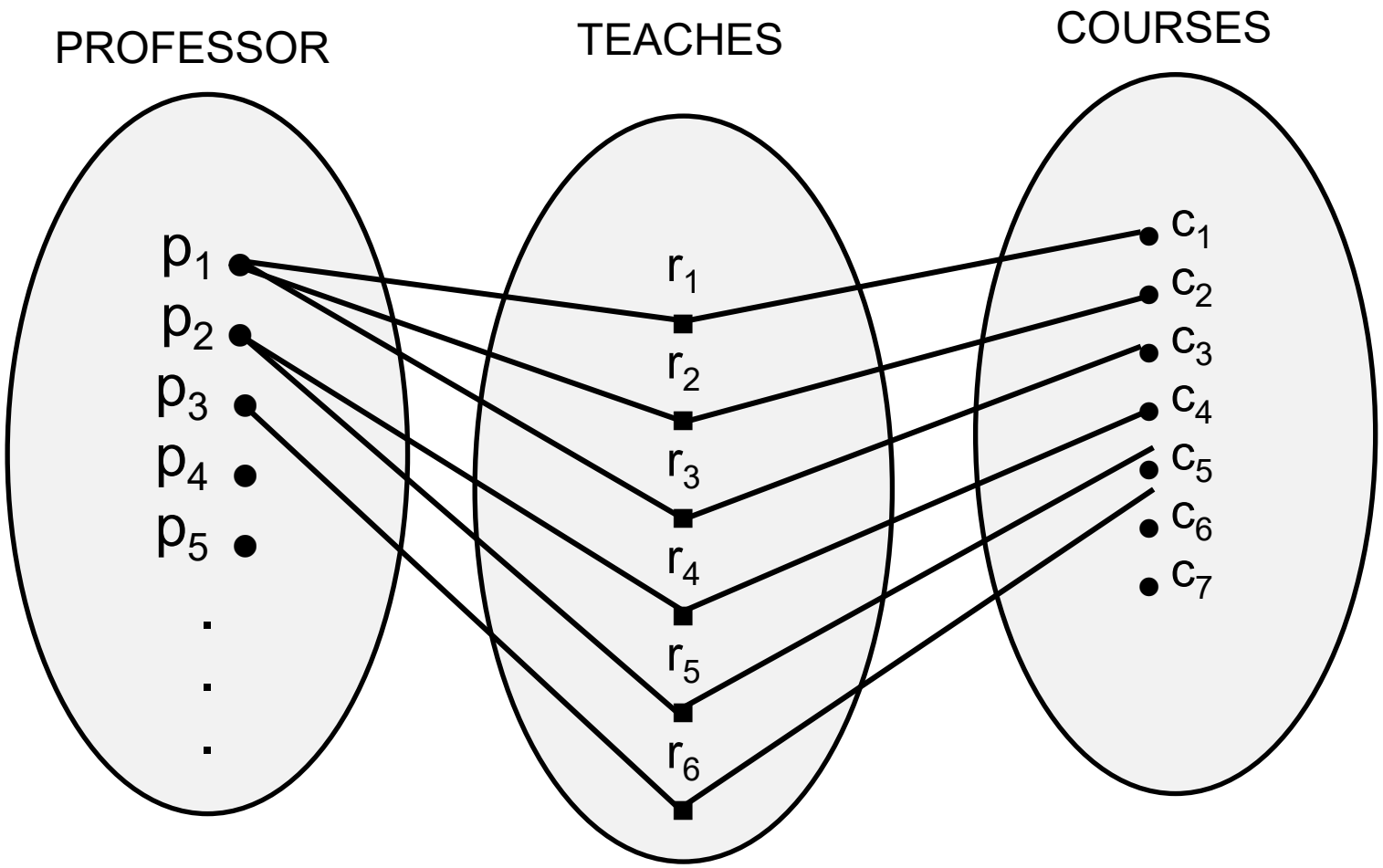
- Maximum number of relationship instances that an entity can participate in
- 1:1, 1:M, M:1, and M:M

- 1:1 – one to one
- 1:M – one to many
- M:1 – many to one
- M:M – many to many

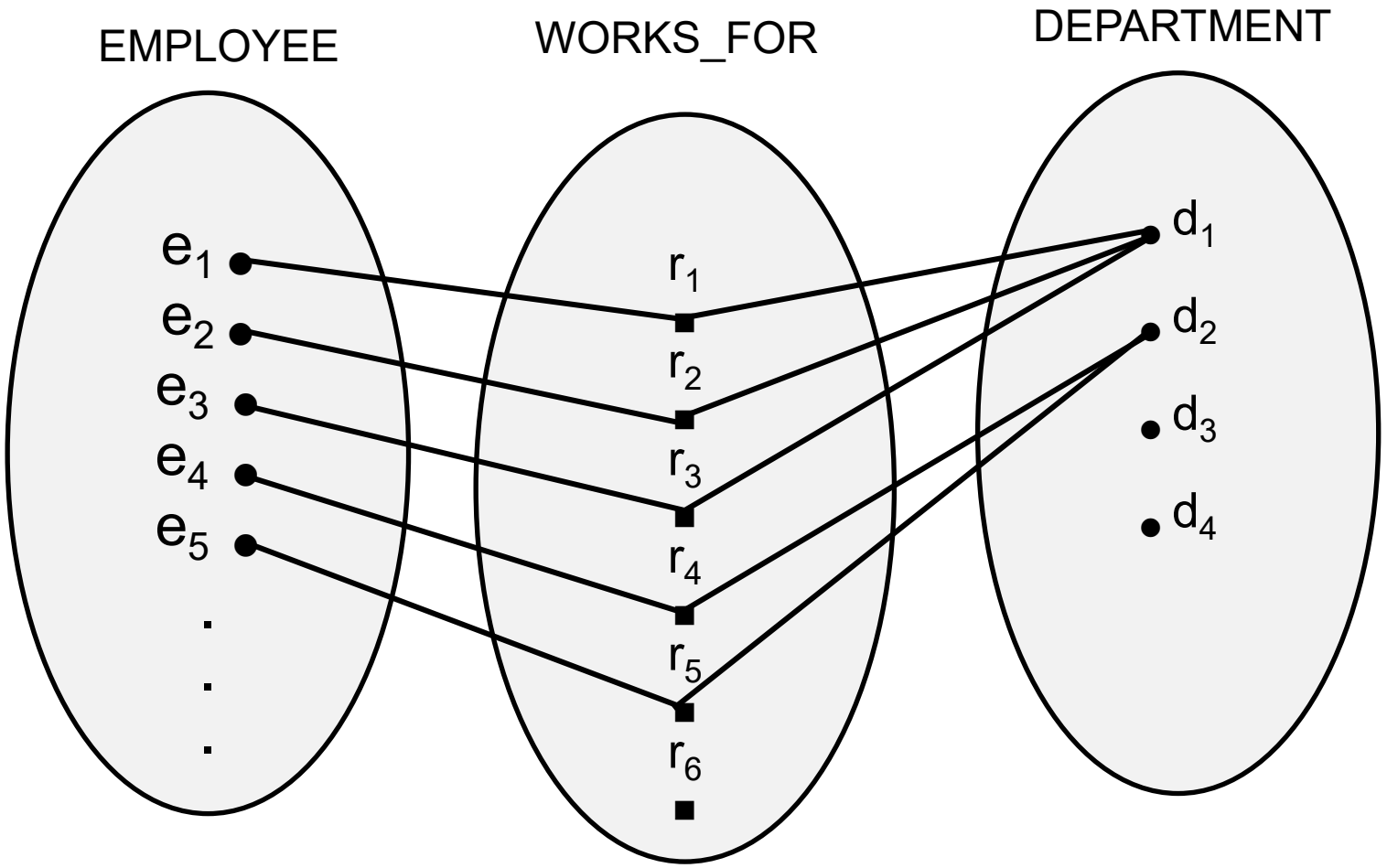
One to One (1:1) Relationship



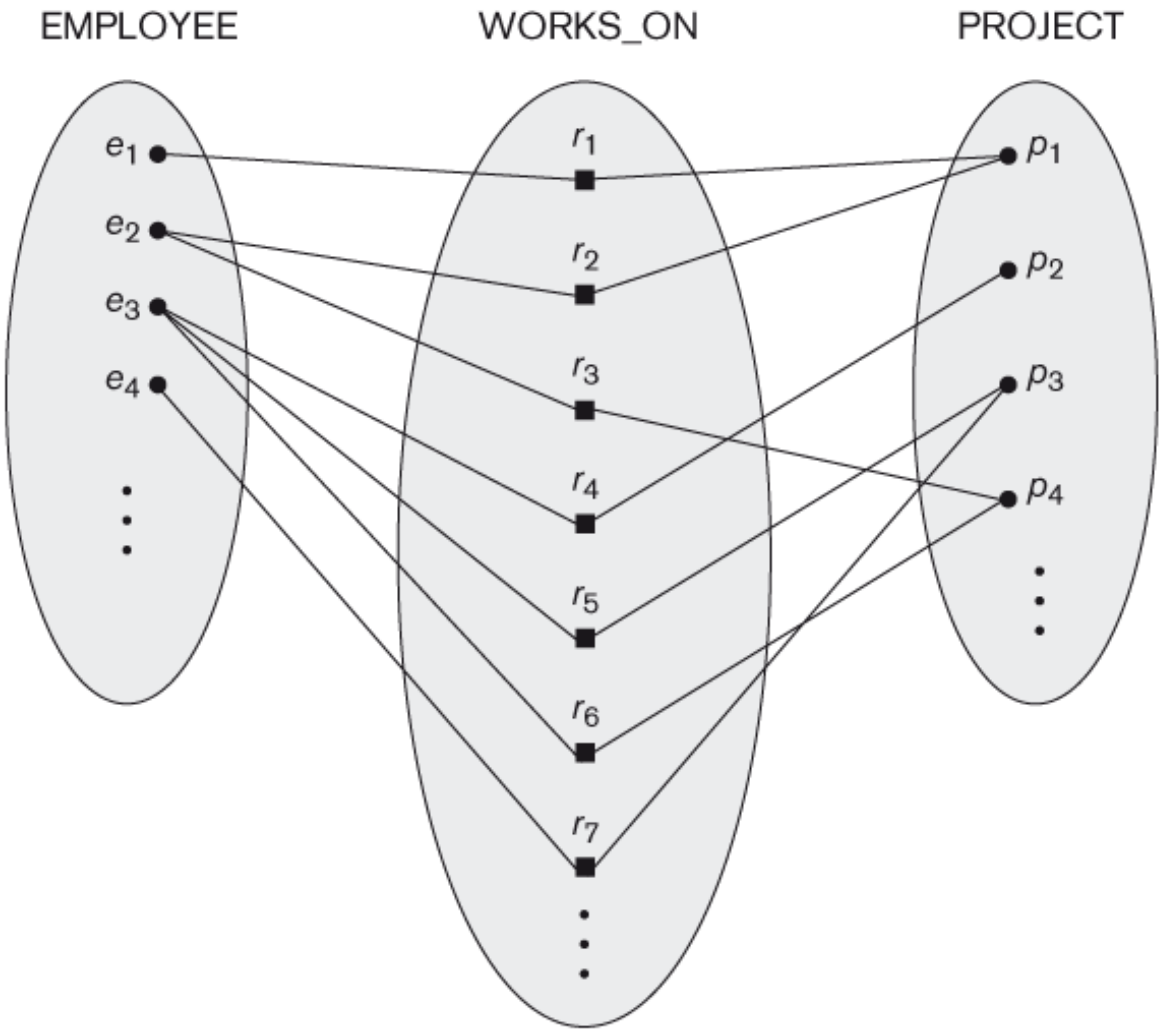
One to Many (1:M) Relationship



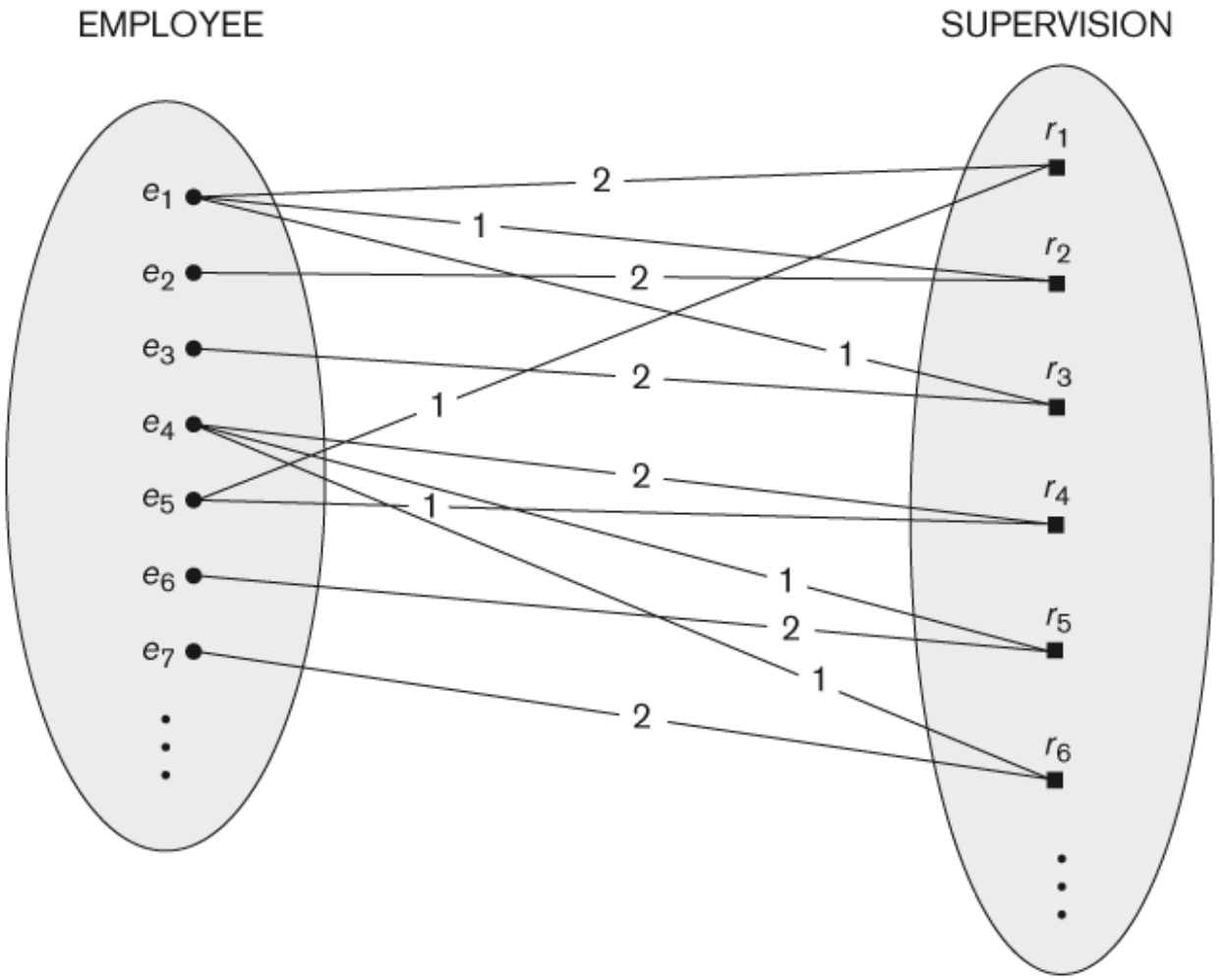
Many to One (M:1) Relationship



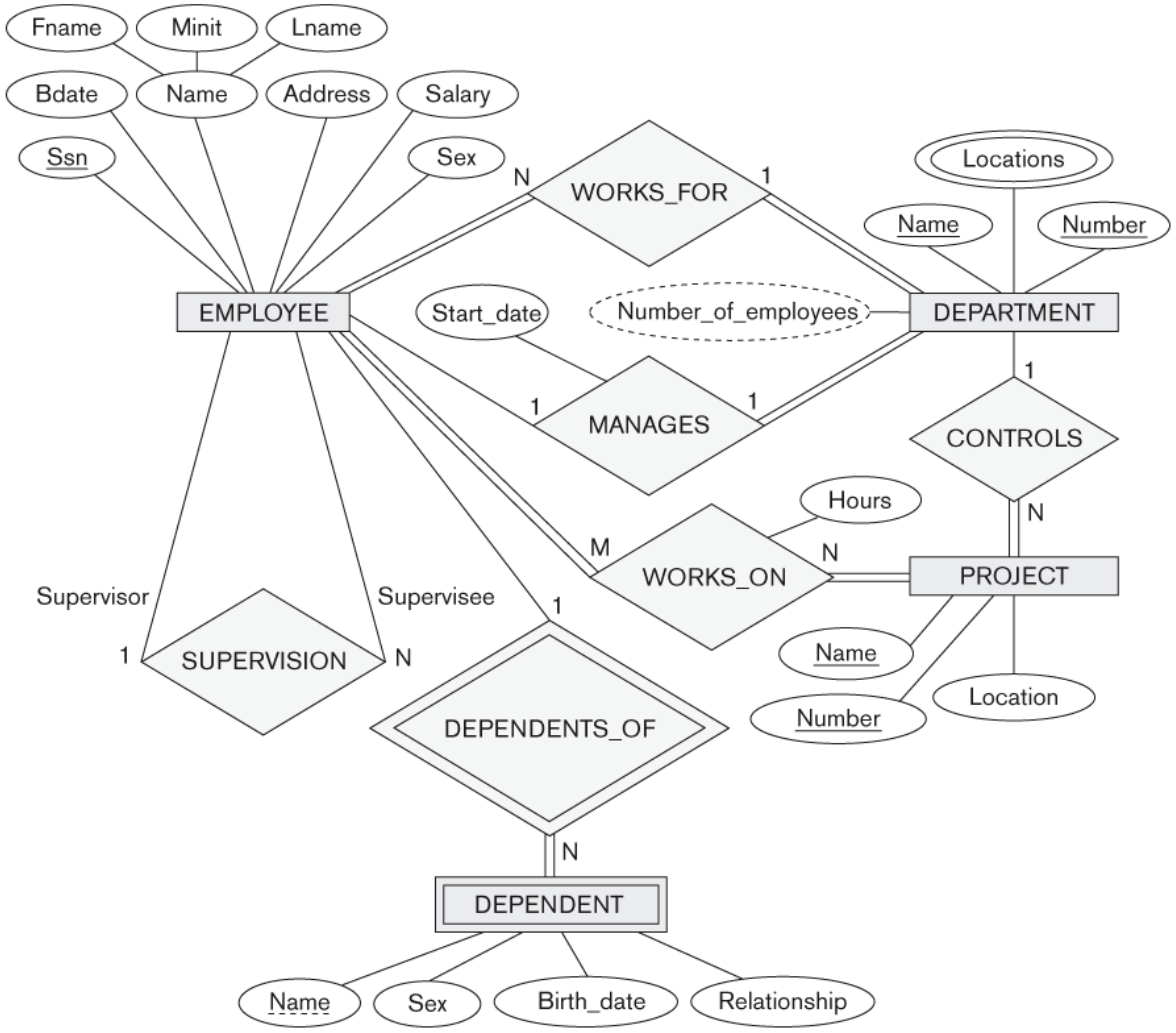
Many to Many (M:M) Relationship



Recursive Relationships



Cardinality on ER Diagram

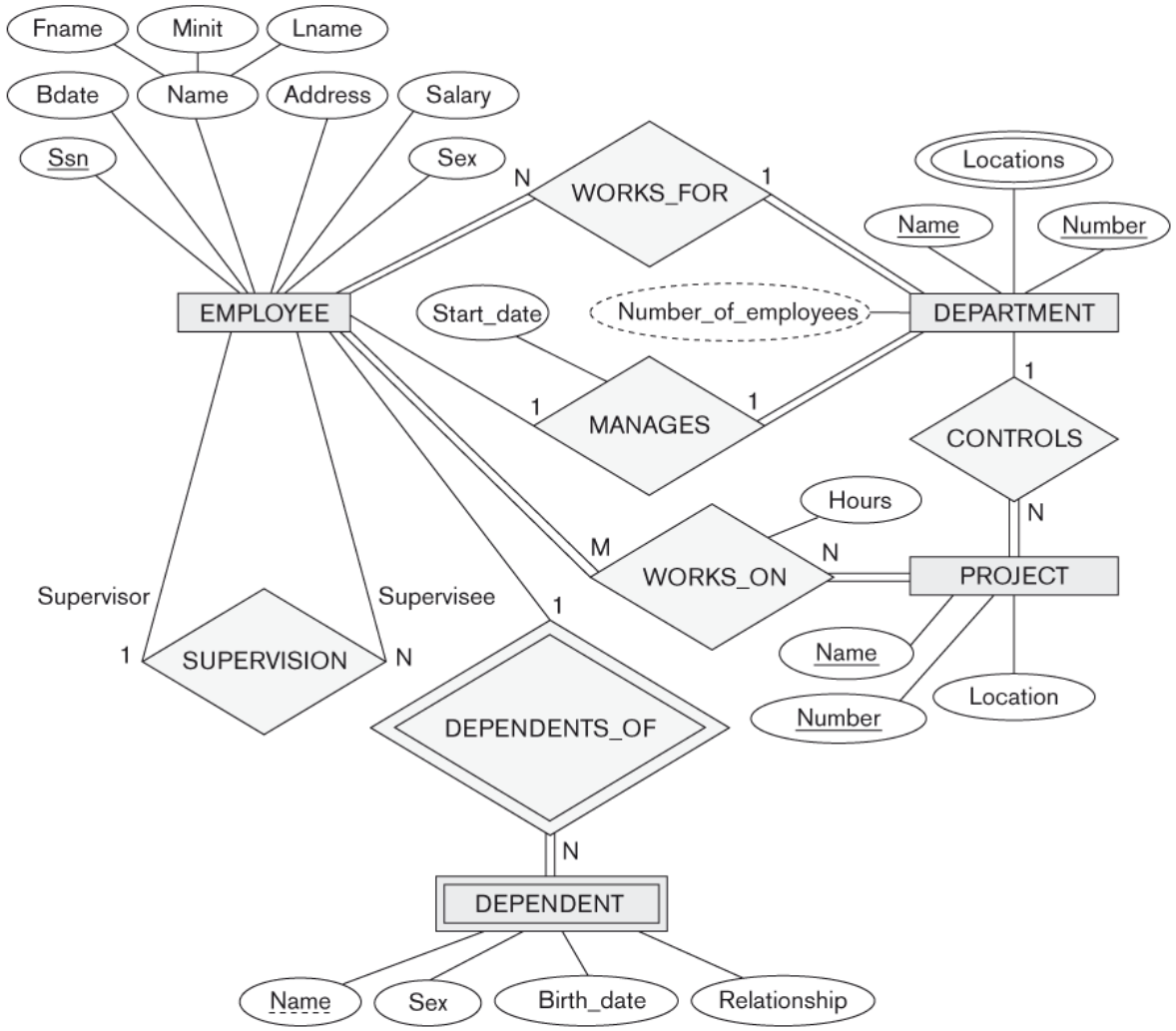


Attributes of Relationship Types

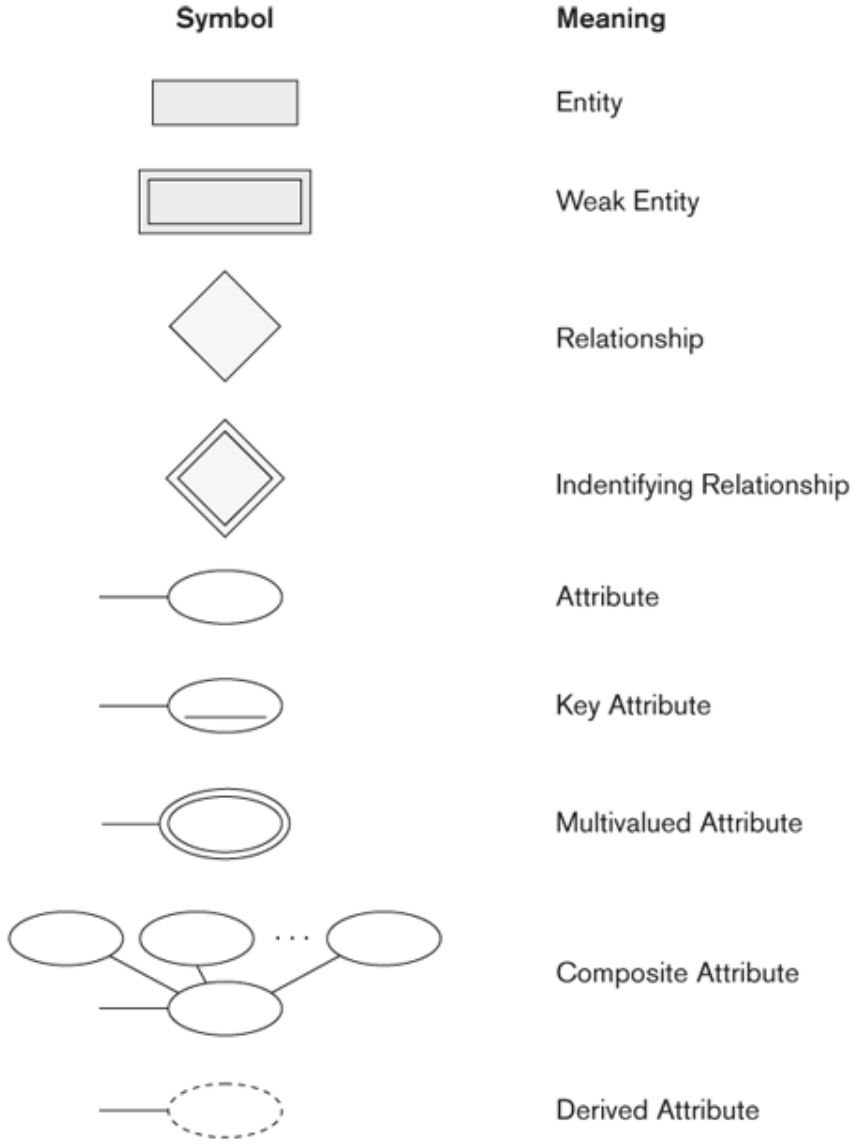
- Number of hours per week that an employee works on a project
  - Employee → works_on → Project
    - Works_on has attribute of NumberOfHours

Weak Entity – "Dependent"

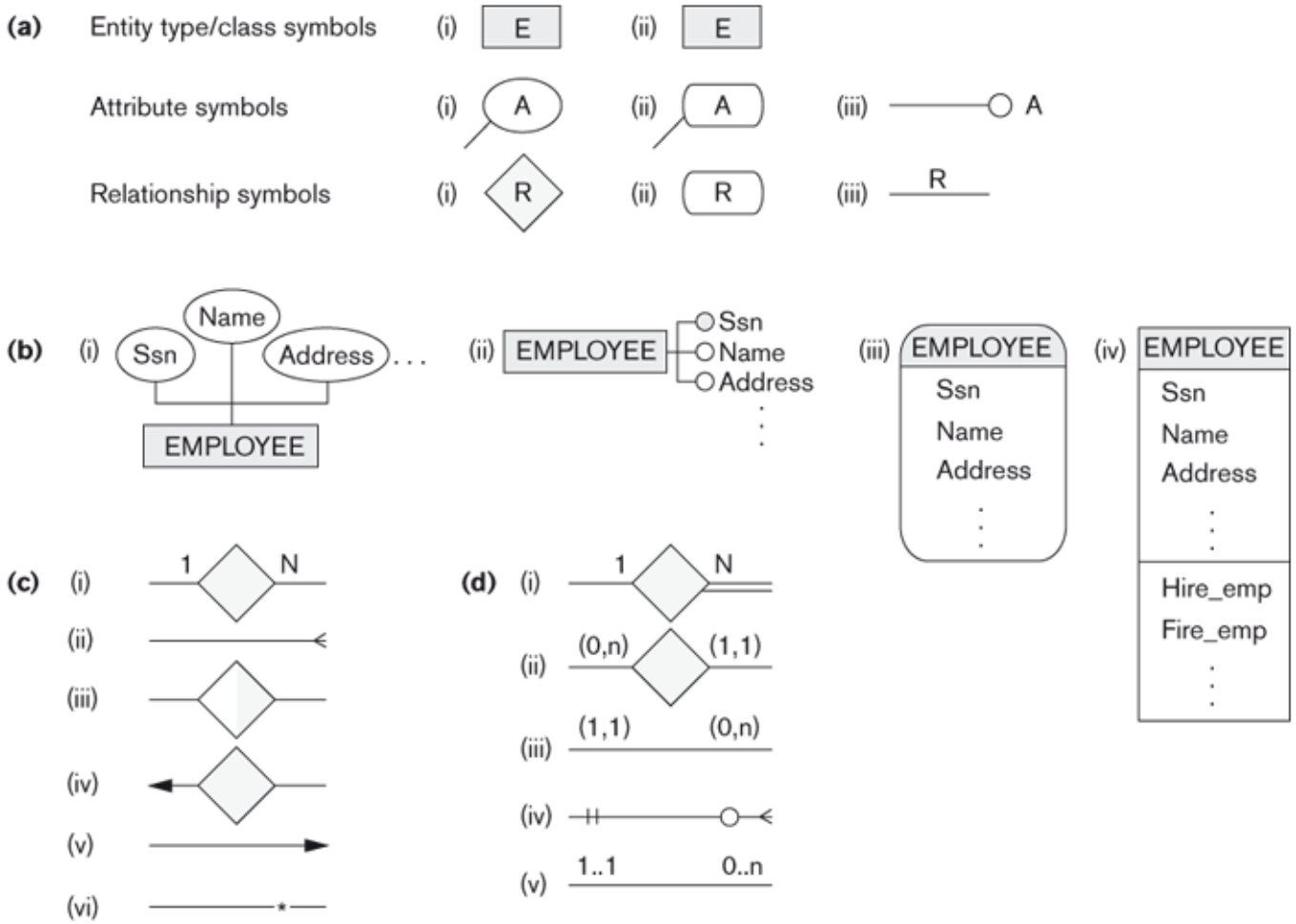
- Entity that cannot be uniquely identified by its attributes alone
- It must use a foreign key in conjunction with its attributes to create a primary key.
- The foreign key is typically a primary key of an entity it is related to.



ER Diagram Conventions



Other Notations for ER Diagrams



CSC 174: Advanced Database Management Systems

Chapter 5

The Relational Data Model and Relational Database Constraints

Amarjot Biring

Introduction

- Tables
- Primary Key
- Foreign Key
- Constraints
- Update Operations

Tables

- Entity = Table
- Store information about related objects
- Rows and columns - think of a spreadsheet

Tables - Rows

- Run horizontally
- Represent each record
- Smallest unit of data that can be inserted into a database.
- Span multiple columns, and therefore, the definition of a column applies to the cell where the row intersects with that column.
- Each row (tuple) in each table is a record with a unique identifier (key)

Tables - Columns

- Run vertically.
- Contain the definition of each field.
- Give each column a name
  - It describes the data that is stored
- The columns of the table hold values of data

- Each record/row usually has a value for each attribute/column, but not always

Tables – Primary Key (PK)

- A special relational database table column (or combination of columns) designated to uniquely identify each table record.
- Used as a unique identifier to quickly parse data within the table.
- A table cannot have more than one primary key.
- Main features
  - It must contain a unique value for each row of data.
  - It cannot contain null values.
  - Every row must have a primary key value.

Tables – Primary Key (PK) Continued

- Primary keys are required for relationship databases!
  - The tables are linked with one another to create connections between the data.
  - The backbone of any relational database is the primary and foreign keys.

- Primary keys implement a technical mechanism to check if every row has a unique, non-empty value in the primary key column.
  - If you try to insert a new row with a duplicate value in the primary key column, this row will be rejected.

- NULL values are not allowed in the primary key column.

Tables – Primary Key (PK) Continued

- Since a table can easily contain thousands of records (including duplicates), a primary key is necessary to ensure that a table record can always be uniquely identified.

- Natural PK vs. Surrogate PK
  - Natural PK
    - Customer's Name
    - SSN
  - Surrogate PK
    - Generated by database system

Tables - Foreign Key (FK)

- Column (or group of columns) used in a relational database to link data between tables.
  - Foreign keys link together two or more tables in relational database.

- References the primary key of another existing table.
  - A foreign key must always reference the primary key of another table.
  - The original table is called the parent table or referenced table, and the referencing table with a foreign key is called a child table.

Tables - Foreign Key (FK) Continued

Constraints

- General constraints
- Entity integrity constraints
- Referential integrity constraints

Constraints – General Constraints

- Salary of an employee should not exceed the salary of the employee's supervisor
- Maximum number of hours an employee can work on all projects per week is 56

- Such constraints can and should be specified and enforced within the application programs that update the database
- It is more common to check for these types of constraints within the **application programs** than to use **constraint specification languages** because the latter are sometimes difficult and complex to use.

Constraints – Entity Integrity Constraints

- Entity integrity constraint
  - No primary key value can be NULL.
  - Primary key is used to identify individual records.
  - Having NULL values for the primary key implies we cannot identify records.
  - For example, if two or more records had a NULL primary key, we may not be able to distinguish them if we try to reference them from other relations.

EmployeeId	FirstName	LastName	City
1	Ken	Adams	New York
2	Ken	Adams	New York
	Michael	Scott	Scranton
	Michael	Scott	Scranton

Constraints – Referential Integrity Constraints

- Specified between two tables
- Used to maintain the consistency among records in the two tables.
- Ensures that all data in a database remains consistent and up to date.
- Prevents incorrect records from being added, deleted, or modified.
- If referential integrity is not enforced, then you may encounter data redundancy and inconsistencies.

Constraints – Referential Integrity Constraints

- Data quality concept that ensures that when you make changes to data in one place, those changes are reflected in other related records
- Informally, the referential integrity constraint states that
  - A record in one table that refers to another table must refer to an existing record in that table.

EmployeeId	FirstName	LastName	DepartmentId
1	Ken	Adams	1
2	Ken	Adams	3
3	Michael	Scott	4
4	Michael	Scott	2

DepartmentId	Name
1	Marketing
2	Information Technology
3	Human Resources
4	Finance

Constraints – Referential Integrity Constraints

- A foreign key can refer to its own table.

EmployeeId	FirstName	LastName	SupervisorId
1	Jim	Halpert	3
2	Dwight	Schrute	3
3	Michael	Scott	4
4	Jan	Levinson	

Update Operations

- Three basic operations that can change the states of tables in the database
  - Insert
    - Insert new data
    - Inserts one or more records into a table(s)
  - Delete
    - Delete old/existing data
    - Deletes one or more records in a table(s)
  - Update (or Modify)
    - Modify existing data
    - Changes the values of some attributes in one or more records in a table(s)
- Whenever these operations are applied, the integrity constraints specified on the relational database schema should not be violated.

The Insert Operation

- Provides a list of attribute values for a new record that is to be inserted into a table.
- Insert can violate constraints
  - Key constraints can be violated if a key value in the new record already exists in another record in the table
  - Entity integrity can be violated if any part of the primary key of the new record is NULL.
  - Referential integrity can be violated if the value of any foreign key in new record refers to a record that does not exist in the referenced table.
- If an insertion violates one or more constraints, the default option is to reject the insertion.

The Insert Operation Continued

- Insert <NULL, ‘Cecilia’, ‘Kolonsky’> into EMPLOYEE
  - Result: This insertion violates the entity integrity constraint (NULL for the primary key EmployeeId), so it is rejected.

- Insert <1,‘Alicia’, ‘Zelaya’> into EMPLOYEE
  - Result: This insertion violates the key constraint because another record with the same EmployeeId value already exists in the EMPLOYEE table, and so it is rejected.

EmployeeId	FirstName	LastName
1	Jim	Halpert
2	Dwight	Schrute
3	Michael	Scott
4	Jan	Levinson

The Insert Operation Continued

- Insert <5,‘Cecilia’, ‘Kolonsky’, 7> into EMPLOYEE
  - Result: This insertion violates the referential integrity constraint specified on DepartmentId in EMPLOYEE because no corresponding referenced record exists in the Department table with DepartmentId = 7.

- Insert <1, ‘Cecilia’, ‘Kolonsky’,4> into EMPLOYEE.
  - Result: This insertion satisfies all constraints, so it is acceptable.

EmployeeId	FirstName	LastName	DepartmentId
1	Jim	Halpert	1
2	Dwight	Schrute	3
3	Michael	Scott	4
4	Jan	Levinson	2

DepartmentId	Name
1	Marketing
2	Information Technology
3	Human Resources
4	Finance

The Delete Operation

- Can violate only referential integrity.

- This occurs if the record being deleted is referenced by foreign keys from other records in the database.

The Delete Operation Continued

- Delete the Employee record with EmployeeId = ‘1’
  - Result: This deletion is acceptable and deletes exactly one record.

- Delete the Department record with DepartmentId = ‘1’.
  - Result: This deletion is not acceptable, because there are records in Employee that refer to this record. Hence, if the record in Department is deleted, referential integrity violations will result.

EmployeeId	FirstName	LastName	DepartmentId
1	Jim	Halpert	1
2	Dwight	Schrute	3
3	Michael	Scott	4
4	Jan	Levinson	2

DepartmentId	Name
1	Marketing
2	Information Technology
3	Human Resources
4	Finance

The Update Operation

- The Update (or Modify) operation is used to change the values of one or more attributes in a record (or records) of some table.
- Updating an attribute that is neither part of a primary key nor part of a foreign key usually causes no problems.
  - The DBMS needs to only check to confirm that the new value is of the correct data type.
- Modifying a primary key value is similar to deleting one record and inserting another in its place because we use the primary key to identify records.
- If a foreign key attribute is modified, the DBMS must make sure that the new value refers to an existing record in the referenced table.

The Update Operation Continued

- Update the first name of the EMPLOYEE record with EmployeeId = ‘1’ to Jimmy.
  - Result: Acceptable

- Update the DepartmentId of the EMPLOYEE record with EmployeeId = ‘1’ to 2.
  - Result: Acceptable.

EmployeeId	FirstName	LastName	DepartmentId
1	Jim	Halpert	1
2	Dwight	Schrute	3
3	Michael	Scott	4
4	Jan	Levinson	2

DepartmentId	Name
1	Marketing
2	Information Technology
3	Human Resources
4	Finance

The Update Operation Continued

- Update the DepartmentId of the EMPLOYEE record with EmployeeId = ‘1’ to 7.
  - Result: Unacceptable, because it violates referential integrity.

- Update the EmployeeId of the EMPLOYEE record with EmployeeId = ‘1’ to ‘2’.
  - Result: Unacceptable, because it violates primary key constraint by repeating a value that already exists as a primary key in another record.

EmployeeId	FirstName	LastName	DepartmentId
1	Jim	Halpert	1
2	Dwight	Schrute	3
3	Michael	Scott	4
4	Jan	Levinson	2

DepartmentId	Name
1	Marketing
2	Information Technology
3	Human Resources
4	Finance

Transactions

- A database application program running against a relational database typically executes one or more transactions.

- A transaction is an executing program that includes some database operations, such as reading from the database, or applying insertions, deletions, or updates to the database.

- At the end of the transaction, it must leave the database in a valid or consistent state that satisfies all the constraints specified on the database schema.

Transactions

- A single transaction may involve any number of retrieval operations and any number of update operations.

- These retrievals and updates will together form an atomic unit of work against the database.

- For example, a transaction to apply a bank withdrawal will typically read the user account record, check if there is a sufficient balance, and then update the record by the withdrawal amount.