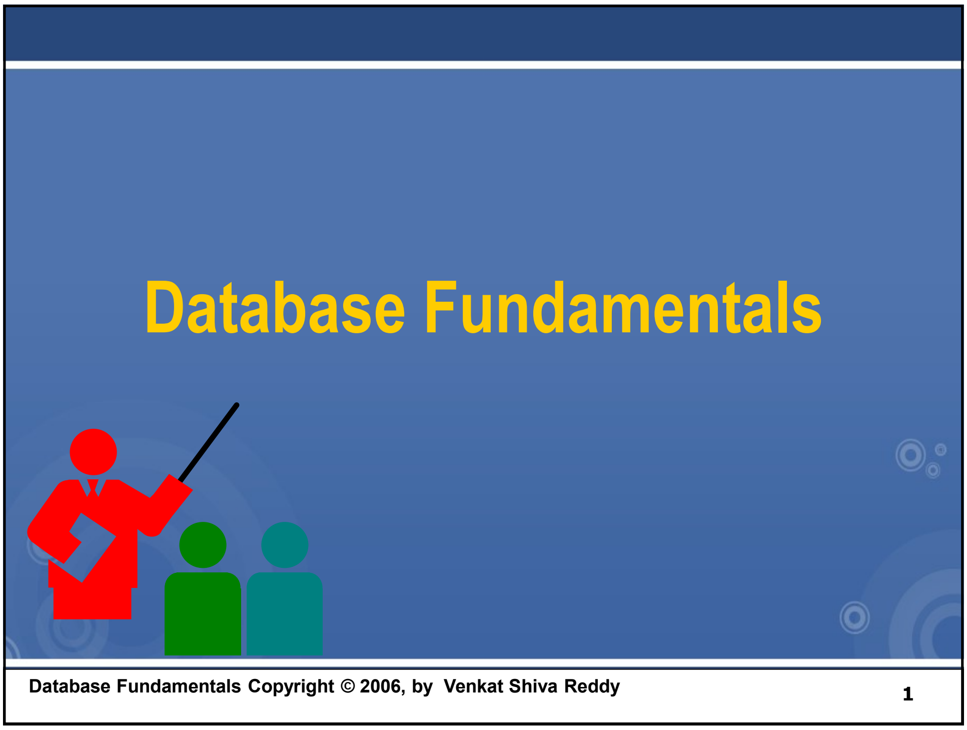
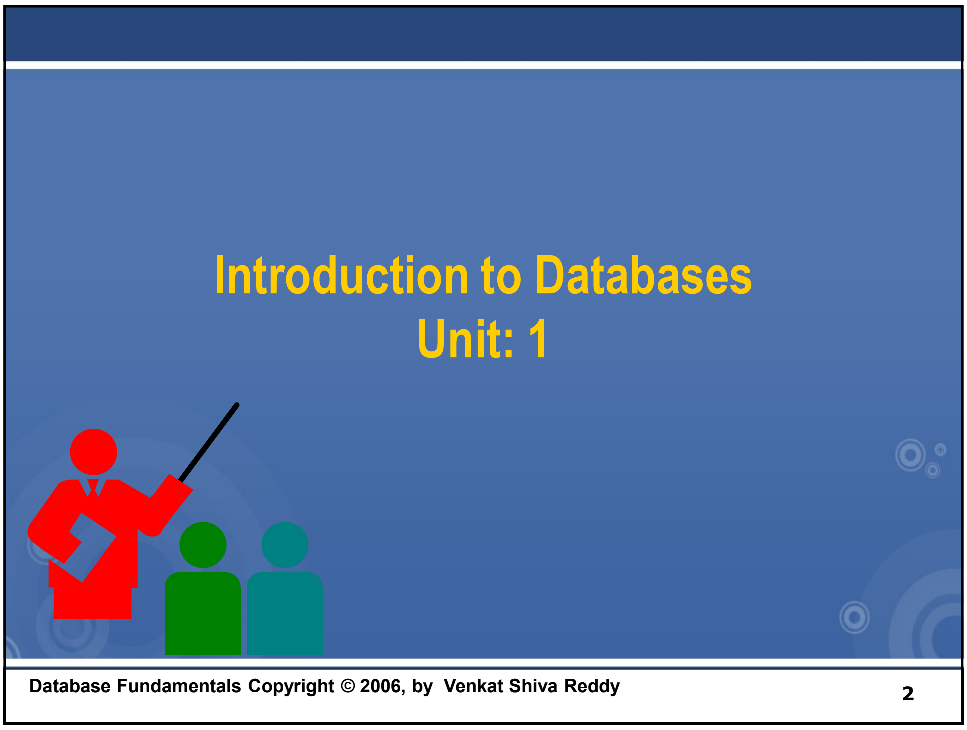


Database Fundamentals

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy 1

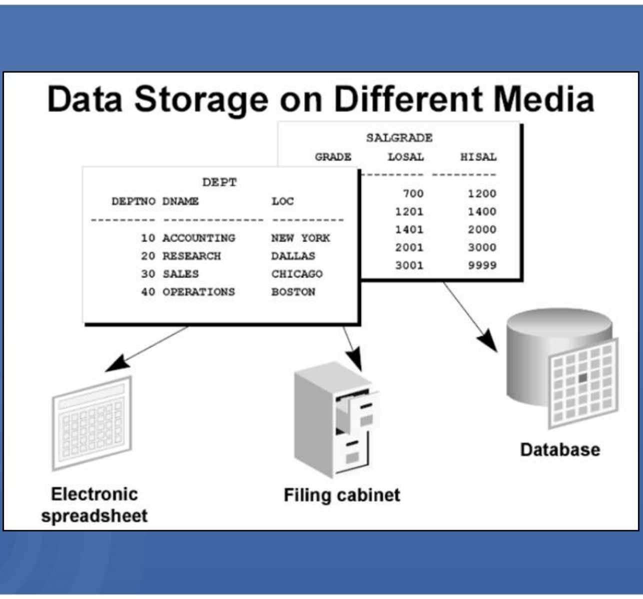


Introduction to Databases
Unit: 1

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy 2

What is a Database?

- A database is a computerized record keeping system
- Databases are designed to offer an organized mechanism for
  - storing,
  - managing and
  - retrieving information.



Data Storage on Different Media

```
                SALGRADE
     GRADE      LOSAL      HISAL
             --------- ---------
                   700      1200
                  1201      1400
                  1401      2000
                  2001      3000
                  3001      9999
```

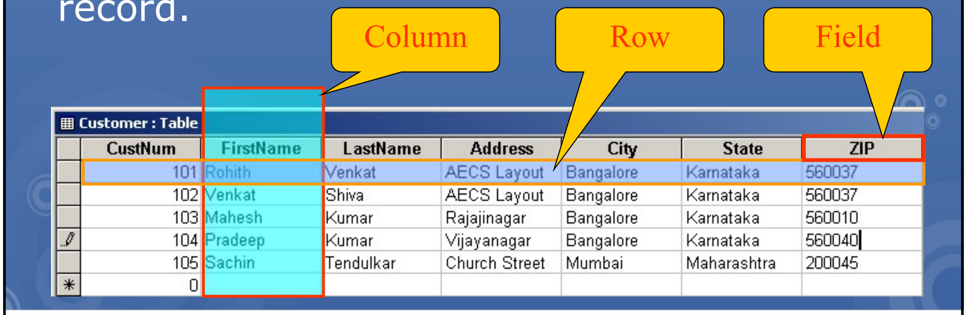
```
                DEPT
   DEPTNO DNAME          LOC
--------- -------------- ----------
       10 ACCOUNTING     NEW YORK
       20 RESEARCH       DALLAS
       30 SALES          CHICAGO
       40 OPERATIONS     BOSTON
```

Functions of a Database

- Retrieve all records that match certain criteria
- Update records in bulk
- Cross-reference records in different tables
- Perform complex aggregate calculations

Database Tables

- Database tables consist of
  - ➢columns and
  - ➢rows.
- Each column contains a different type of attribute and each row corresponds to a single record.



Customer : Table

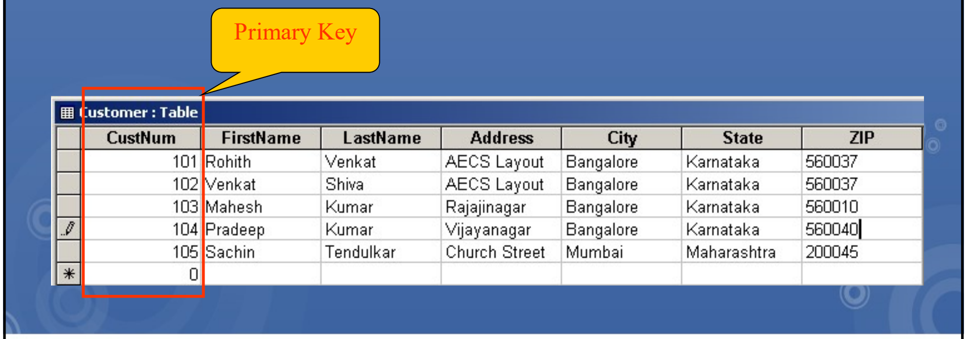
	CustNum	FirstName	LastName	Address	City	State	ZIP
	101	Rohith	Venkat	AECS Layout	Bangalore	Karnataka	560037
	102	Venkat	Shiva	AECS Layout	Bangalore	Karnataka	560037
	103	Mahesh	Kumar	Rajajinagar	Bangalore	Karnataka	560010
✎	104	Pradeep	Kumar	Vijayanagar	Bangalore	Karnataka	560040
	105	Sachin	Tendulkar	Church Street	Mumbai	Maharashtra	200045
*	0						

Database Keys

- Databases use tables to organize data.
- Each table consists of number of rows and columns
- Each table have
  - Primary Key
  - Foreign Key

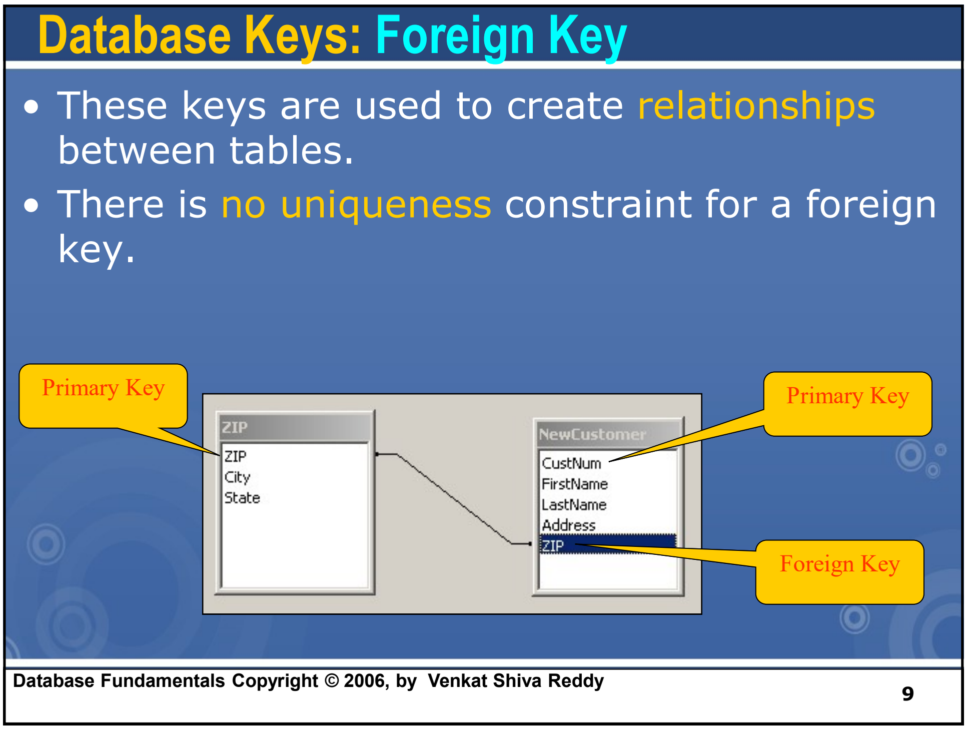
Database Keys: Primary Key

- Every database table should have one or more columns designated as the primary key.
- The value this key holds should be unique for each record in the database.



Primary Key

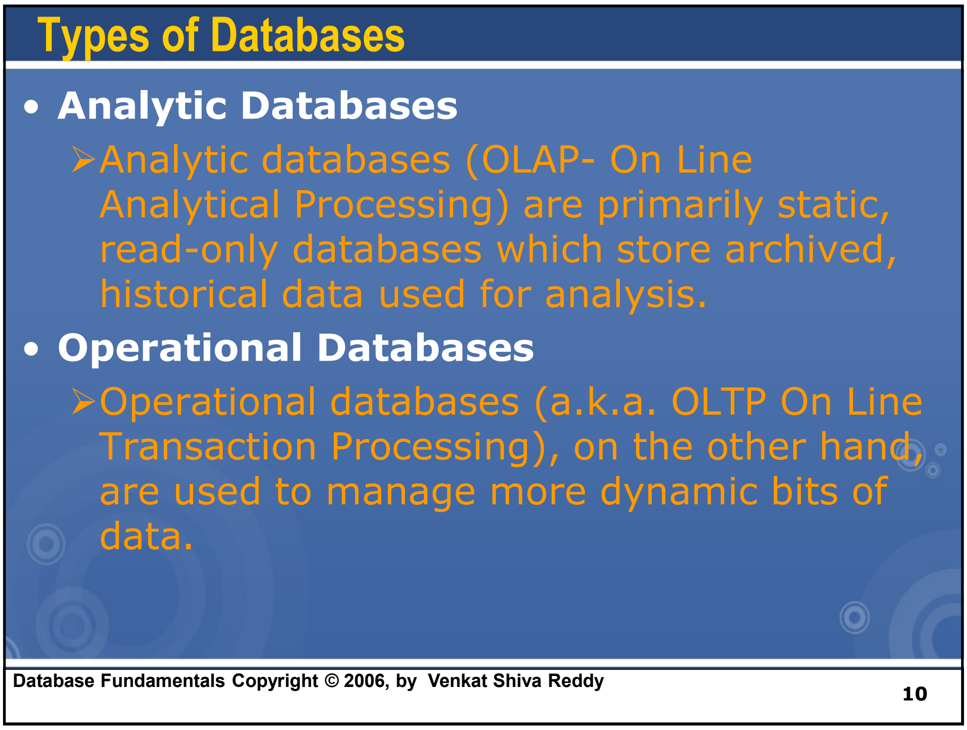
CustNum	FirstName	LastName	Address	City	State	ZIP
101	Rohith	Venkat	AECS Layout	Bangalore	Karnataka	560037
102	Venkat	Shiva	AECS Layout	Bangalore	Karnataka	560037
103	Mahesh	Kumar	Rajajinagar	Bangalore	Karnataka	560010
104	Pradeep	Kumar	Vijayanagar	Bangalore	Karnataka	560040
105	Sachin	Tendulkar	Church Street	Mumbai	Maharashtra	200045
0						



Database Keys: Foreign Key

- These keys are used to create relationships between tables.
- There is no uniqueness constraint for a foreign key.

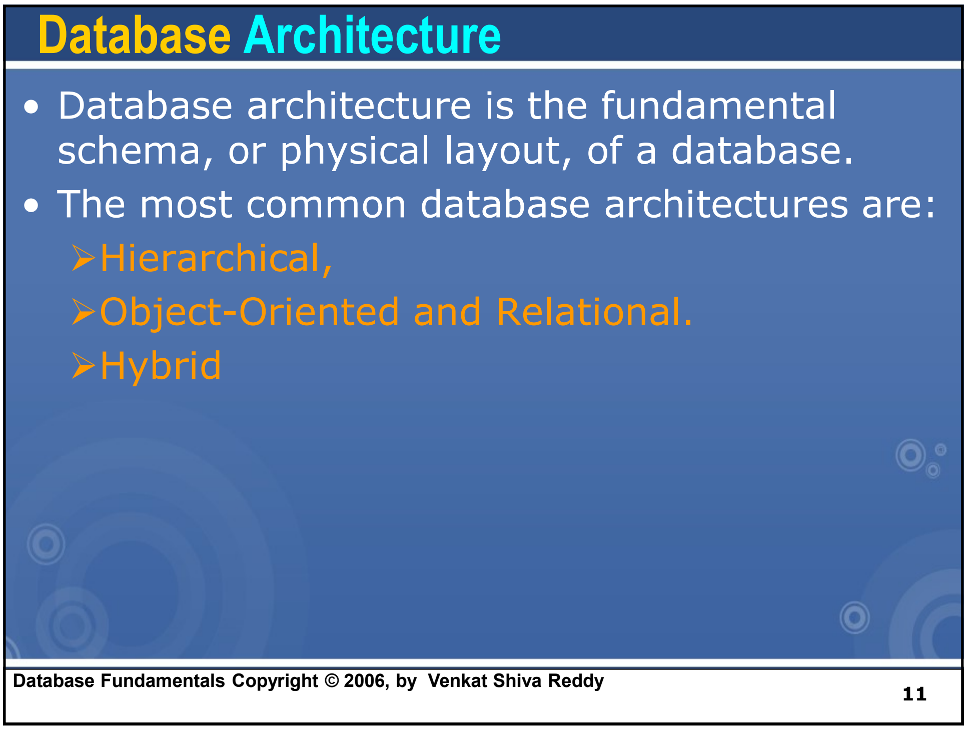
Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy 9



Types of Databases

- **Analytic Databases**
  - ➢Analytic databases (OLAP- On Line Analytical Processing) are primarily static, read-only databases which store archived, historical data used for analysis.
- **Operational Databases**
  - ➢Operational databases (a.k.a. OLTP On Line Transaction Processing), on the other hand, are used to manage more dynamic bits of data.

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy 10

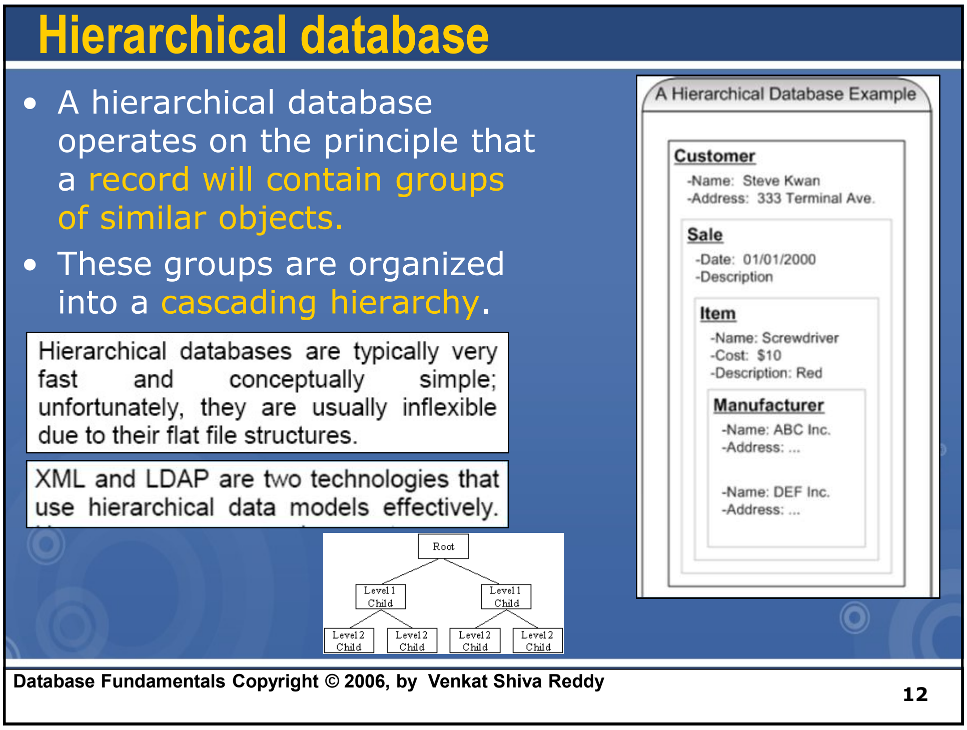


Database Architecture

- Database architecture is the fundamental schema, or physical layout, of a database.
- The most common database architectures are:
  - ➢Hierarchical,
  - ➢Object-Oriented and Relational.
  - ➢Hybrid

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

11



Hierarchical database

- A hierarchical database operates on the principle that a record will contain groups of similar objects.
- These groups are organized into a cascading hierarchy.

Hierarchical databases are typically very fast and conceptually simple; unfortunately, they are usually inflexible due to their flat file structures.

XML and LDAP are two technologies that use hierarchical data models effectively.

A Hierarchical Database Example

Customer
-Name: Steve Kwan
-Address: 333 Terminal Ave.

Sale
-Date: 01/01/2000
-Description

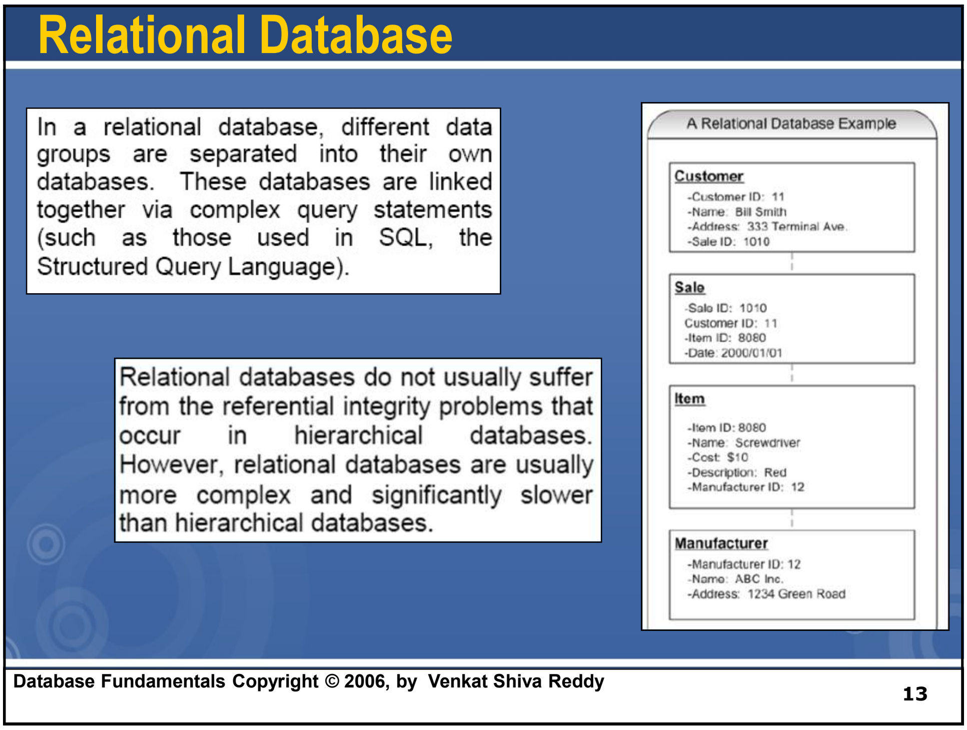
Item
-Name: Screwdriver
-Cost: $10
-Description: Red

Manufacturer
-Name: ABC Inc.
-Address: ...

-Name: DEF Inc.
-Address: ...

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

12



Relational Database

In a relational database, different data groups are separated into their own databases. These databases are linked together via complex query statements (such as those used in SQL, the Structured Query Language).

Relational databases do not usually suffer from the referential integrity problems that occur in hierarchical databases. However, relational databases are usually more complex and significantly slower than hierarchical databases.

A Relational Database Example

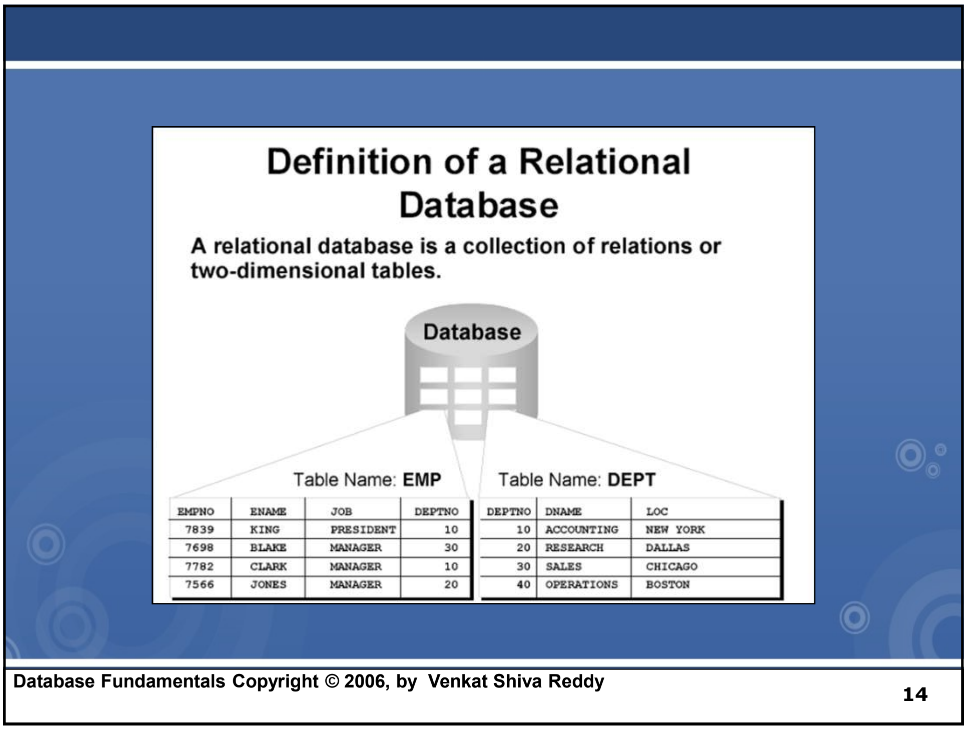
Customer
-Customer ID: 11
-Name: Bill Smith
-Address: 333 Terminal Ave.
-Sale ID: 1010

Sale
-Sale ID: 1010
Customer ID: 11
-Item ID: 8080
-Date: 2000/01/01

Item
-Item ID: 8080
-Name: Screwdriver
-Cost: $10
-Description: Red
-Manufacturer ID: 12

Manufacturer
-Manufacturer ID: 12
-Name: ABC Inc.
-Address: 1234 Green Road

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy 13



Definition of a Relational Database

A relational database is a collection of relations or two-dimensional tables.

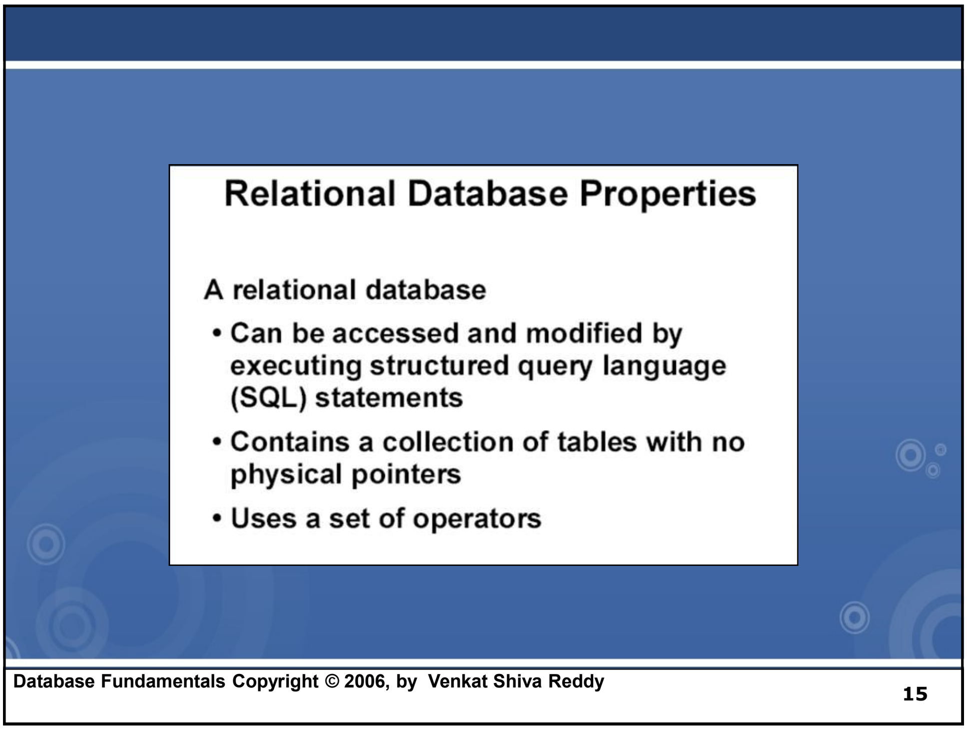
Table Name: **EMP**

EMPNO	ENAME	JOB	DEPTNO
7839	KING	PRESIDENT	10
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7566	JONES	MANAGER	20

Table Name: **DEPT**

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy 14

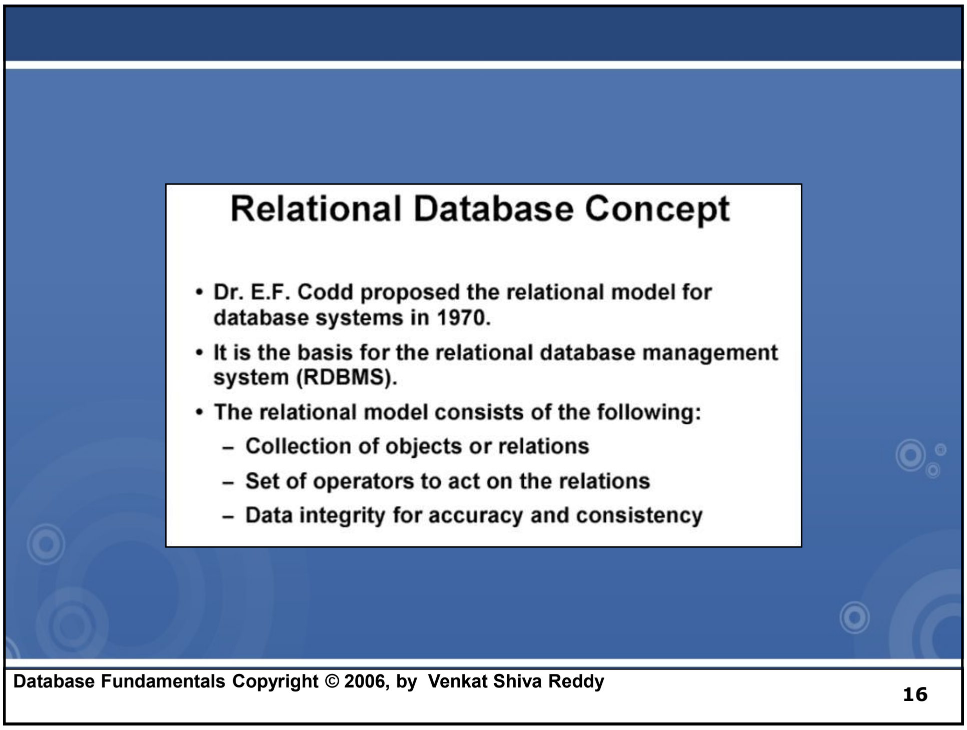


Relational Database Properties

A relational database

- Can be accessed and modified by executing structured query language (SQL) statements
- Contains a collection of tables with no physical pointers
- Uses a set of operators

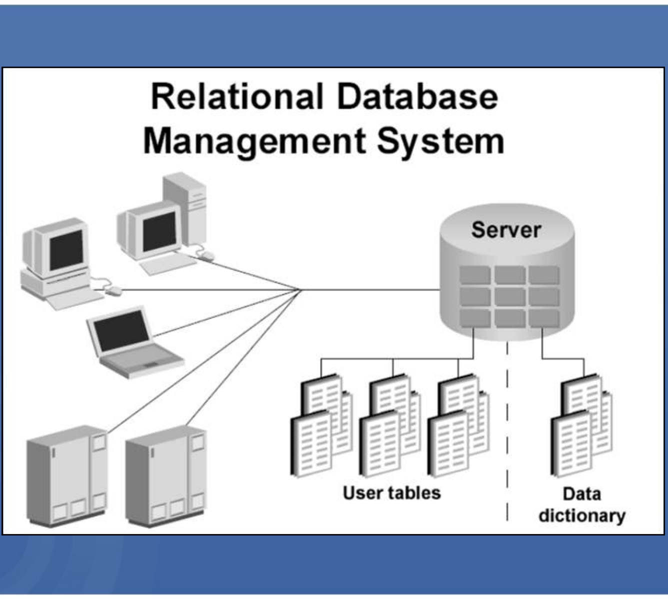
15



Relational Database Concept

- Dr. E.F. Codd proposed the relational model for database systems in 1970.
- It is the basis for the relational database management system (RDBMS).
- The relational model consists of the following:
  - Collection of objects or relations
  - Set of operators to act on the relations
  - Data integrity for accuracy and consistency

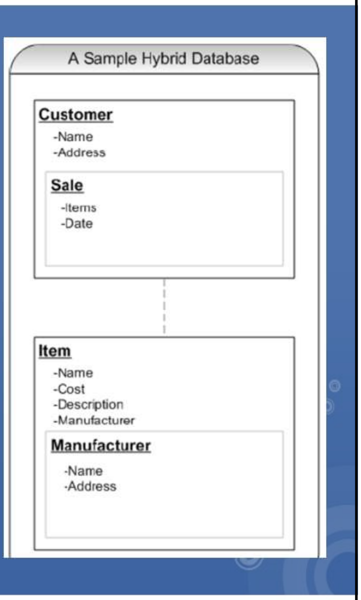
16



Relational Database Management System

Hybrid Database

A combination of
Hierarchical and Relational



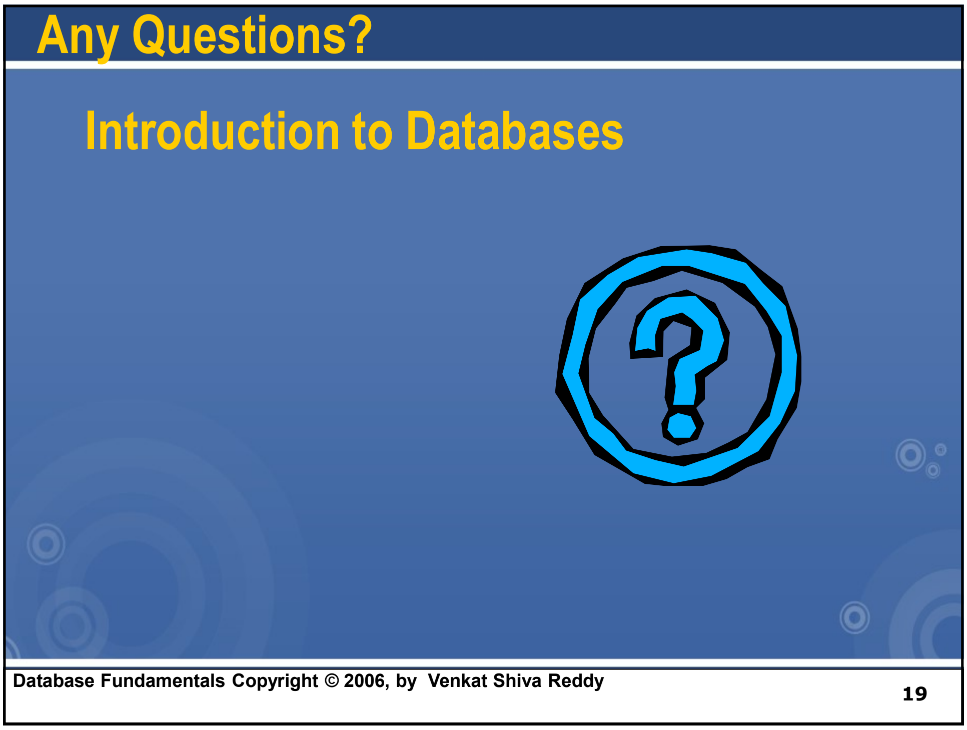
A Sample Hybrid Database

Customer
-Name
-Address

Sale
-Items
-Date

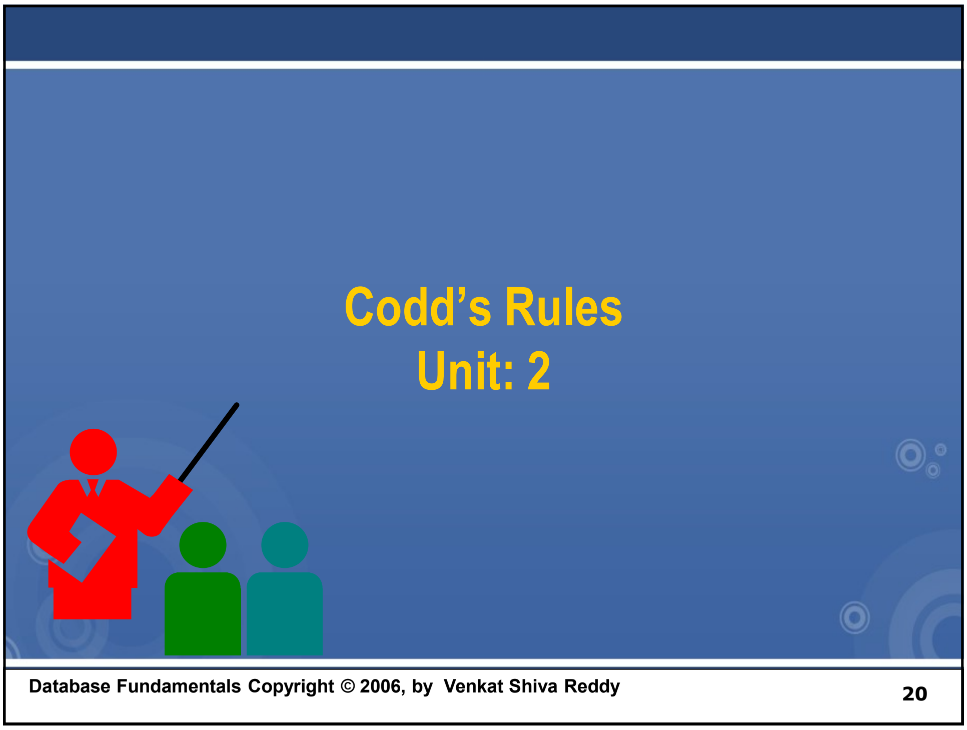
Item
-Name
-Cost
-Description
-Manufacturer

Manufacturer
-Name
-Address

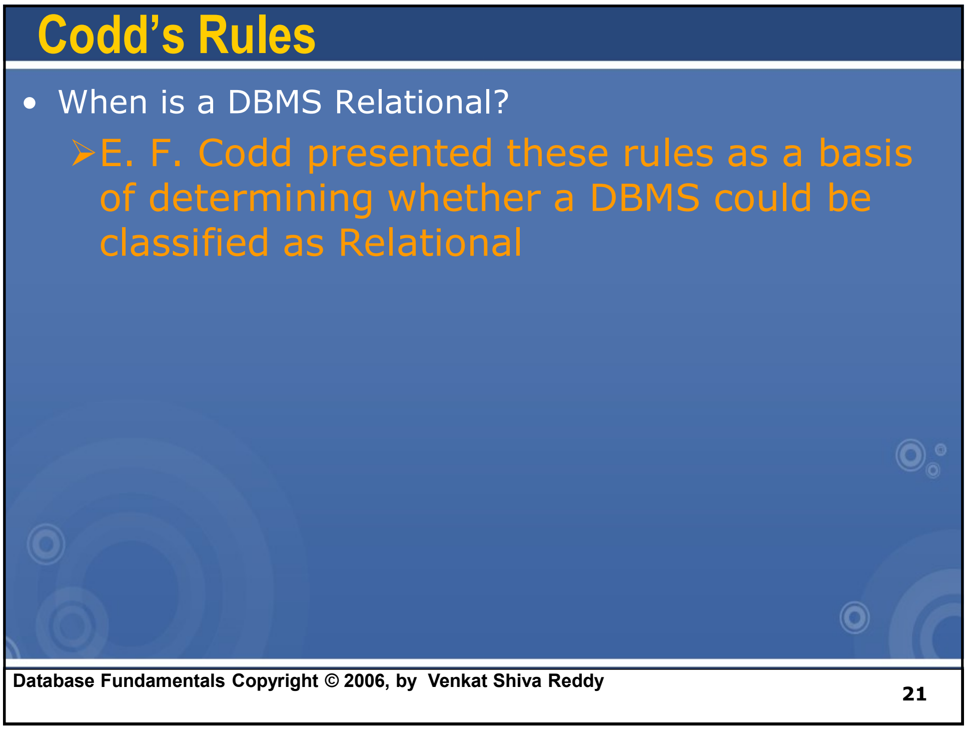


Any Questions?

Introduction to Databases

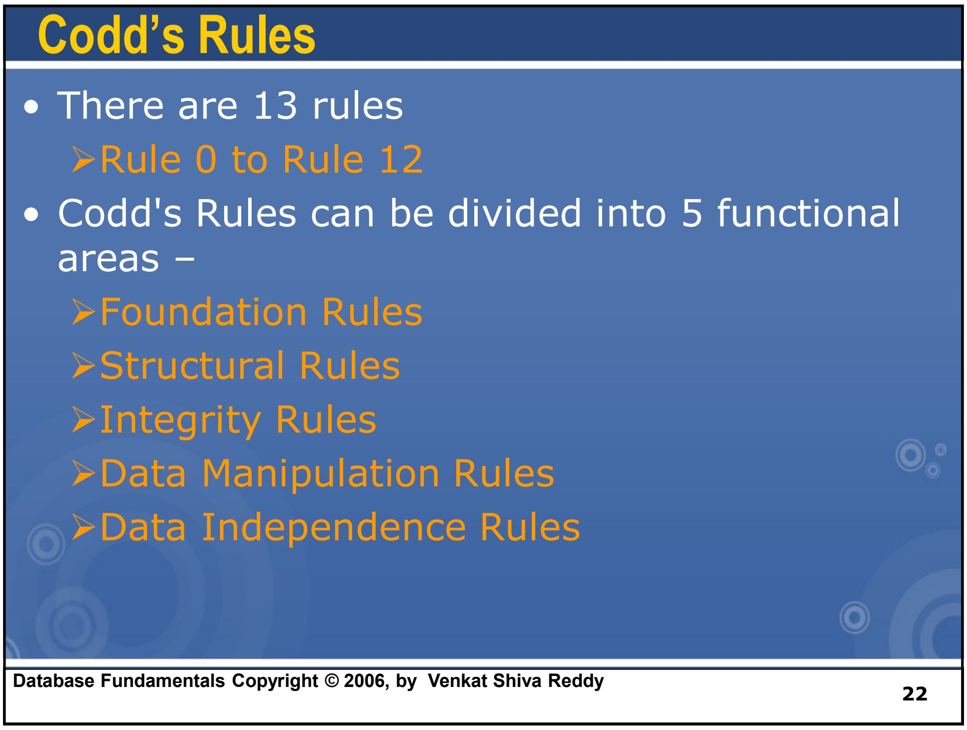


Codd's Rules
Unit: 2



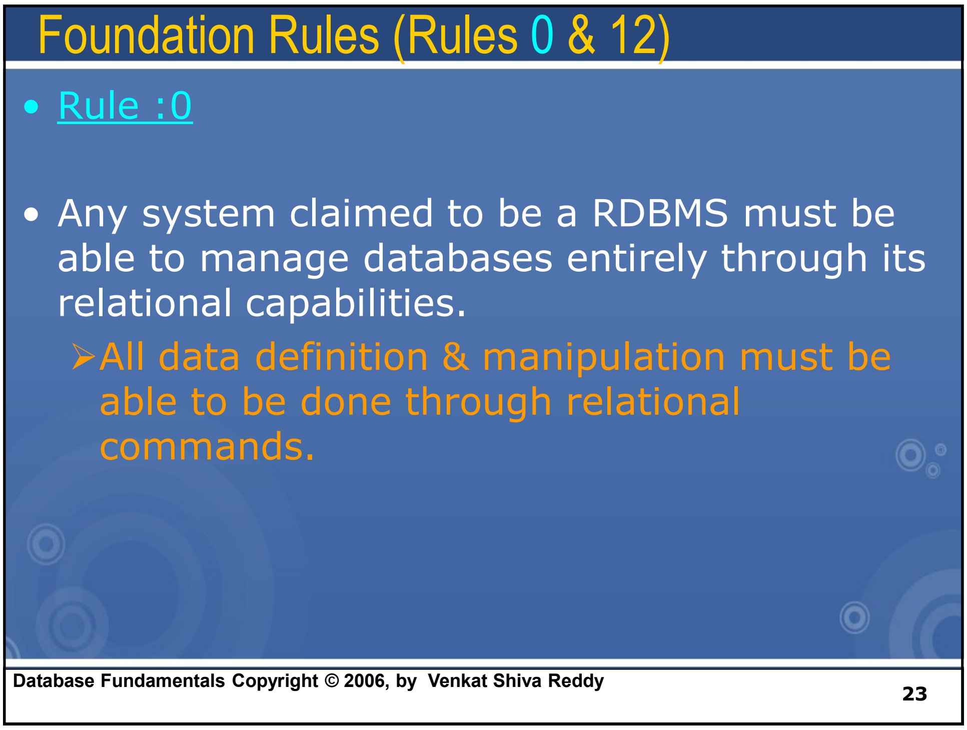
Codd’s Rules

- When is a DBMS Relational?
  - ➢E. F. Codd presented these rules as a basis of determining whether a DBMS could be classified as Relational



Codd’s Rules

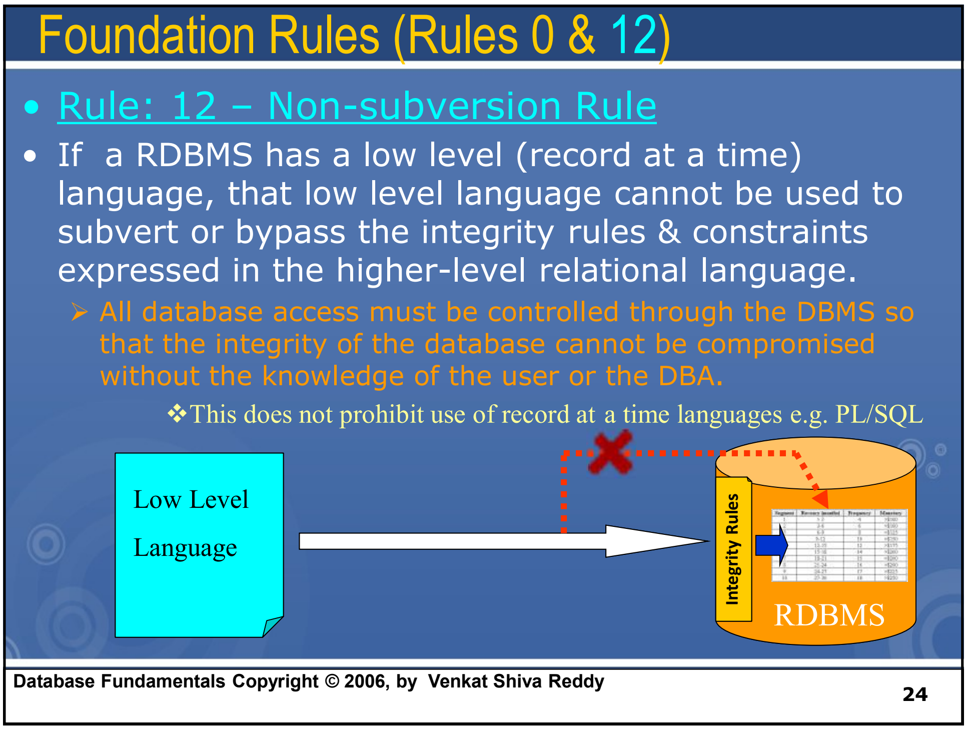
- There are 13 rules
  - ➢Rule 0 to Rule 12
- Codd's Rules can be divided into 5 functional areas –
  - ➢Foundation Rules
  - ➢Structural Rules
  - ➢Integrity Rules
  - ➢Data Manipulation Rules
  - ➢Data Independence Rules



Foundation Rules (Rules 0 & 12)

- Rule :0

- Any system claimed to be a RDBMS must be able to manage databases entirely through its relational capabilities.
  - ➢All data definition & manipulation must be able to be done through relational commands.



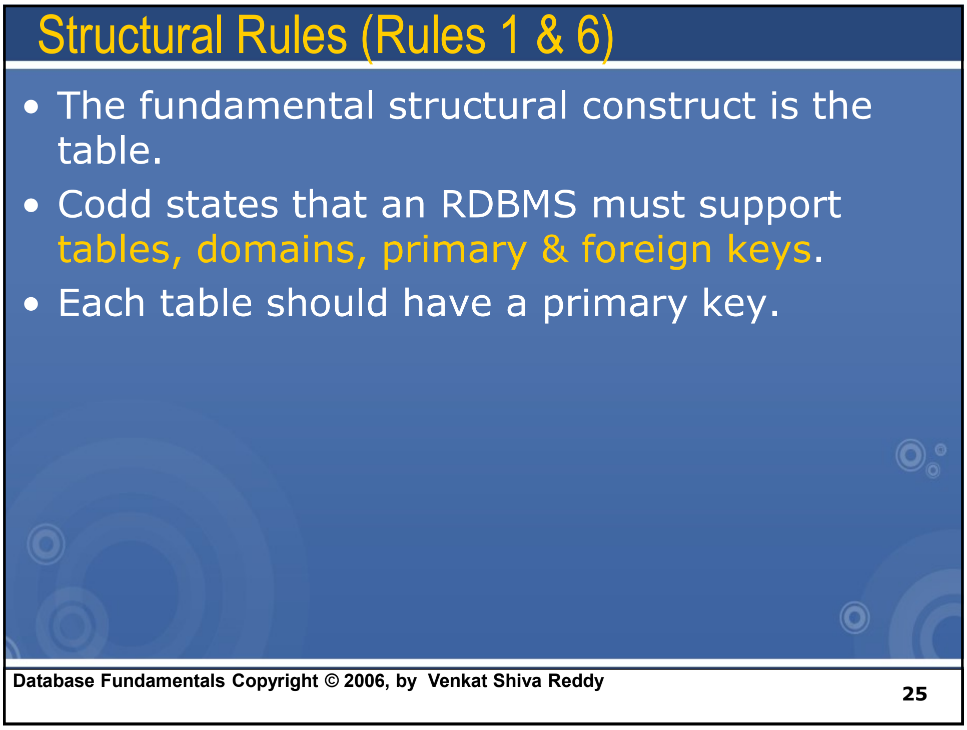
Foundation Rules (Rules 0 & 12)

- Rule: 12 – Non-subversion Rule
- If a RDBMS has a low level (record at a time) language, that low level language cannot be used to subvert or bypass the integrity rules & constraints expressed in the higher-level relational language.
  - ➢ All database access must be controlled through the DBMS so that the integrity of the database cannot be compromised without the knowledge of the user or the DBA.
    - ❖This does not prohibit use of record at a time languages e.g. PL/SQL

Low Level Language

Integrity Rules

RDBMS

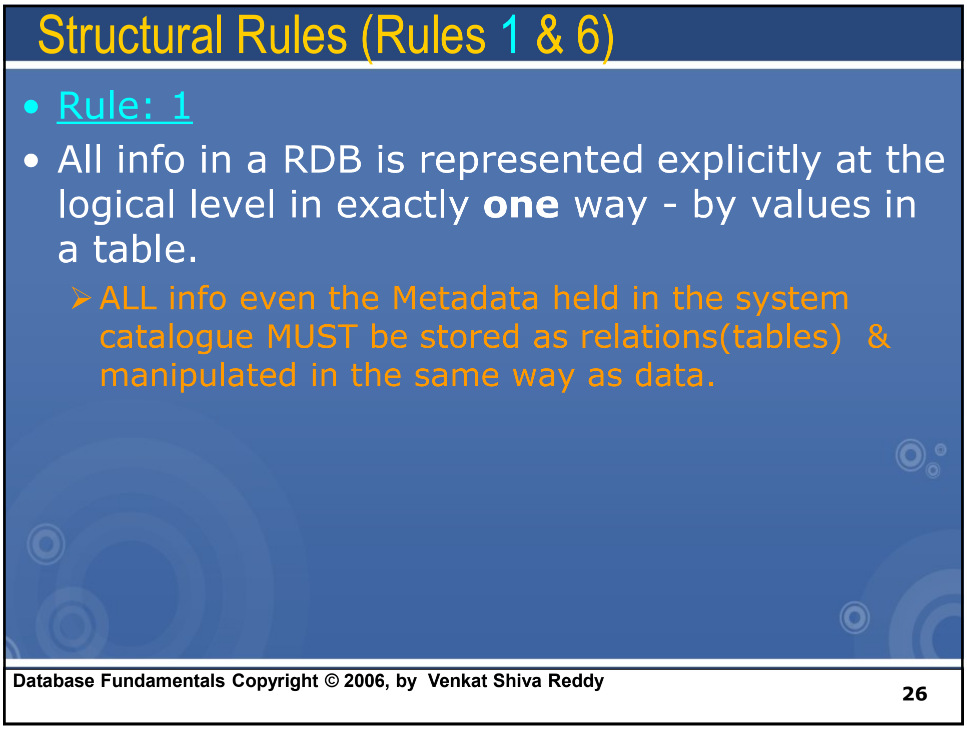


Structural Rules (Rules 1 & 6)

- The fundamental structural construct is the table.
- Codd states that an RDBMS must support tables, domains, primary & foreign keys.
- Each table should have a primary key.

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

25



Structural Rules (Rules 1 & 6)

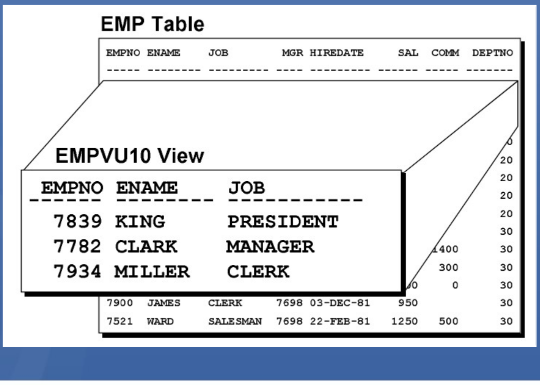
- Rule: 1
- All info in a RDB is represented explicitly at the logical level in exactly **one** way - by values in a table.
  - ALL info even the Metadata held in the system catalogue MUST be stored as relations(tables) & manipulated in the same way as data.

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

26

Structural Rules (Rules 1 & 6)

- Rule: 6 – View Updating
- All views that are theoretically updatable are updatable by the system.
  - ➢ Not really implemented yet by any available system.



EMP Table

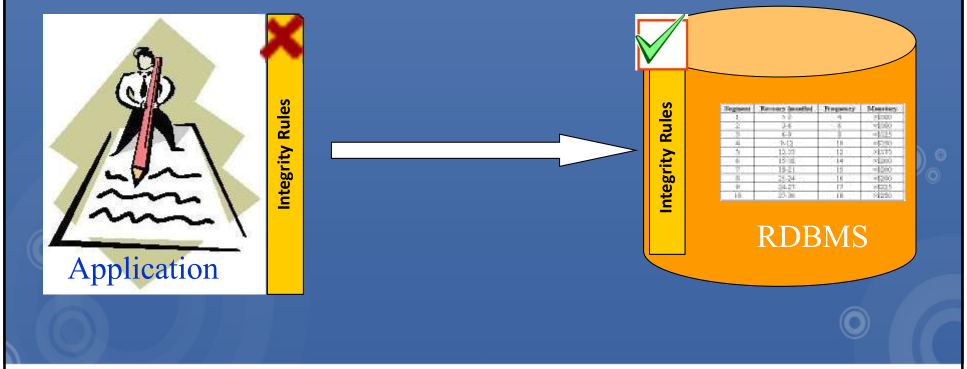
```
EMPNO ENAME    JOB       MGR  HIREDATE  SAL    COMM  DEPTNO
----- -------- --------- ---- --------- ------ ----- -------
                                                          20
                                                          20
                                                          20
                                                          20
                                                          30
                                                    1400  30
                                                     300  30
                                                       0  30
7900  JAMES    CLERK     7698 03-DEC-81    950            30
7521  WARD     SALESMAN  7698 22-FEB-81   1250       500  30
```

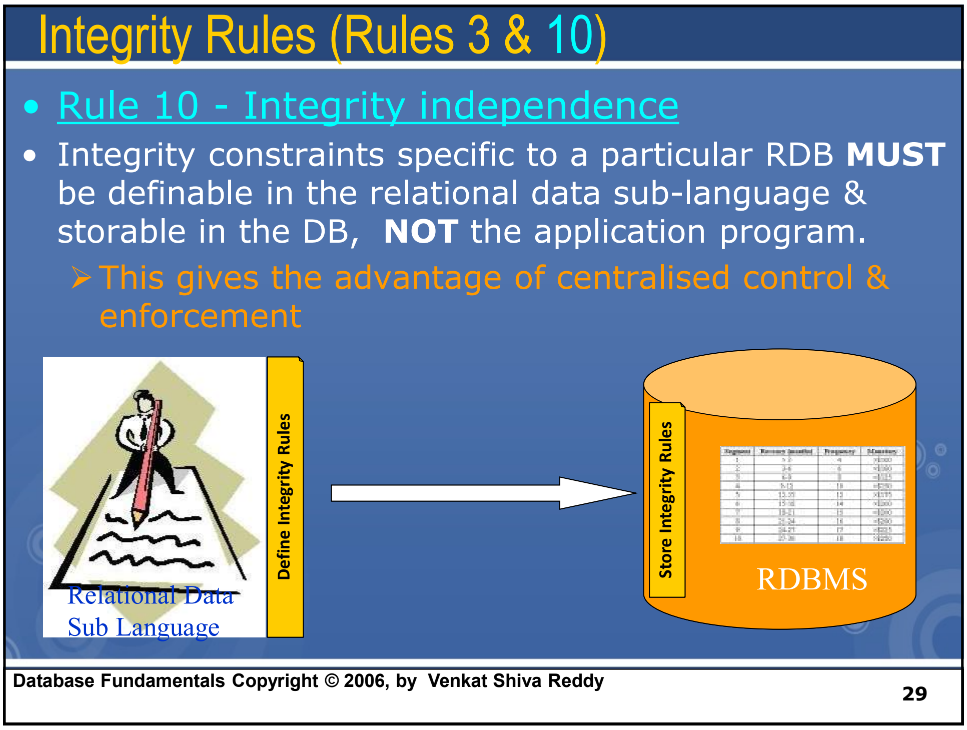
EMPVU10 View

```
EMPNO  ENAME     JOB
------ --------- ------------
  7839 KING      PRESIDENT
  7782 CLARK     MANAGER
  7934 MILLER    CLERK
```

Integrity Rules (Rules 3 & 10)

- Integrity should be maintained by the DBMS not the application.



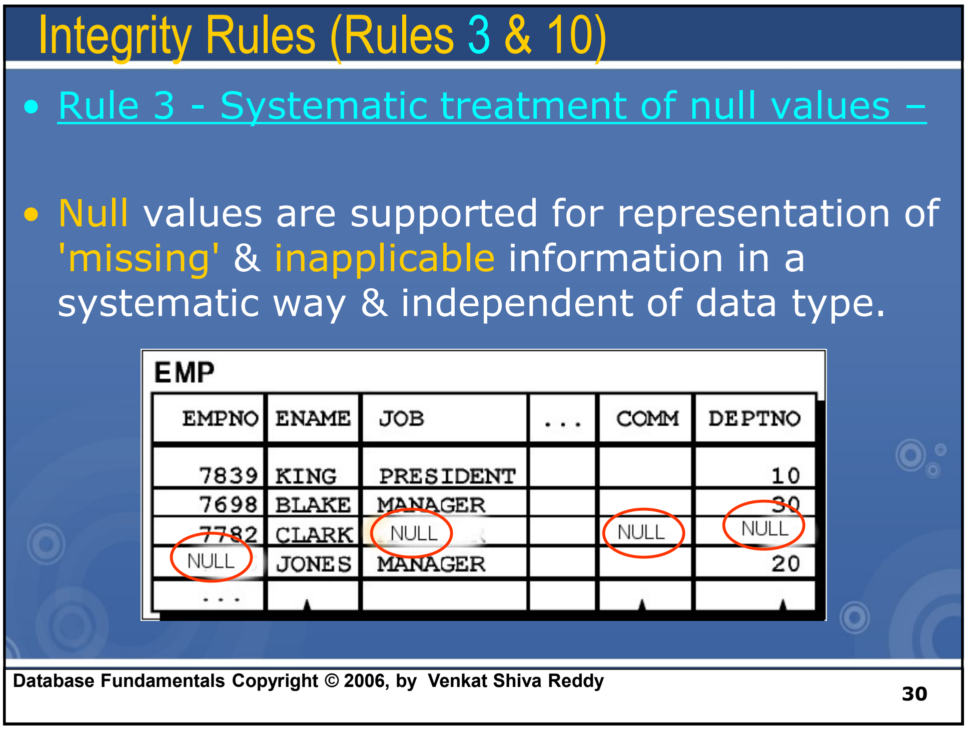


Integrity Rules (Rules 3 & 10)

- Rule 10 - Integrity independence
- Integrity constraints specific to a particular RDB **MUST** be definable in the relational data sub-language & storable in the DB, **NOT** the application program.
  - ➢ This gives the advantage of centralised control & enforcement

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

29



Integrity Rules (Rules 3 & 10)

- Rule 3 - Systematic treatment of null values –
- Null values are supported for representation of 'missing' & inapplicable information in a systematic way & independent of data type.

EMP

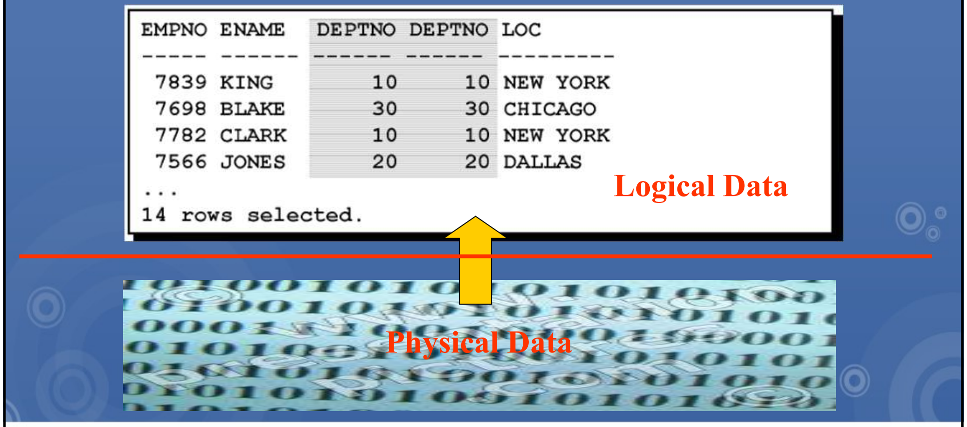
EMPNO	ENAME	JOB	. . .	COMM	DEPTNO
7839	KING	PRESIDENT			10
7698	BLAKE	MANAGER			30
7782	CLARK	NULL		NULL	NULL
NULL	JONES	MANAGER			20
. . .					

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

30

Data Manipulation Rules (Rule 2, 4, 5 & 7)

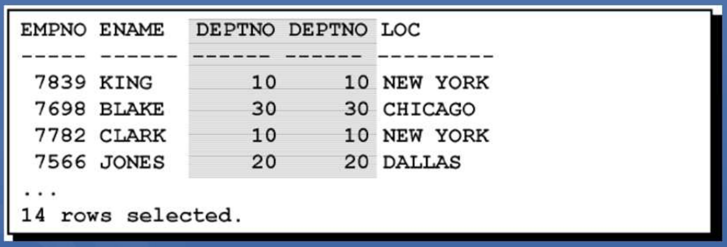
- User should be able to manipulate the 'Logical View' of the data with no need for knowledge of how it is Physically stored or accessed.



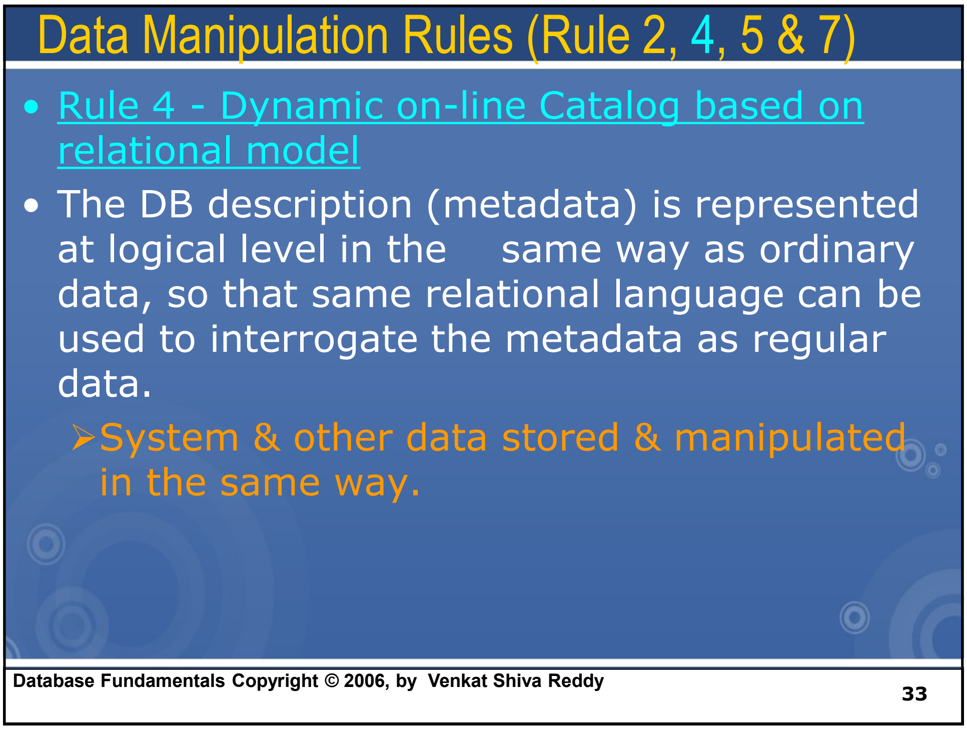
```
EMPNO ENAME   DEPTNO DEPTNO LOC
----- ------ ------ ------ ---------
 7839 KING       10     10 NEW YORK
 7698 BLAKE      30     30 CHICAGO
 7782 CLARK      10     10 NEW YORK
 7566 JONES      20     20 DALLAS
...
14 rows selected.
```

Data Manipulation Rules (Rule 2, 4, 5 & 7)

- Rule 2 - Guaranteed Access -
- Each & every datum in an RDB is guaranteed to be logically accessible by a combination of table name, primary key value & column name.



```
EMPNO ENAME   DEPTNO DEPTNO LOC
----- ------ ------ ------ ---------
 7839 KING       10     10 NEW YORK
 7698 BLAKE      30     30 CHICAGO
 7782 CLARK      10     10 NEW YORK
 7566 JONES      20     20 DALLAS
...
14 rows selected.
```

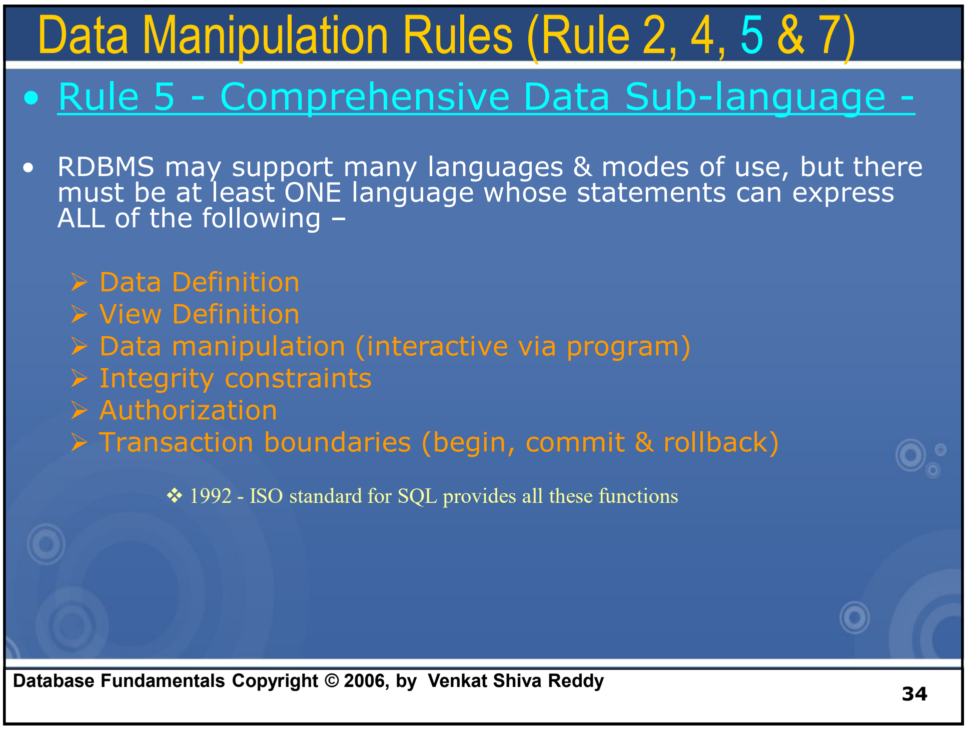


Data Manipulation Rules (Rule 2, 4, 5 & 7)

- Rule 4 - Dynamic on-line Catalog based on relational model
- The DB description (metadata) is represented at logical level in the same way as ordinary data, so that same relational language can be used to interrogate the metadata as regular data.
  - ➢System & other data stored & manipulated in the same way.

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

33



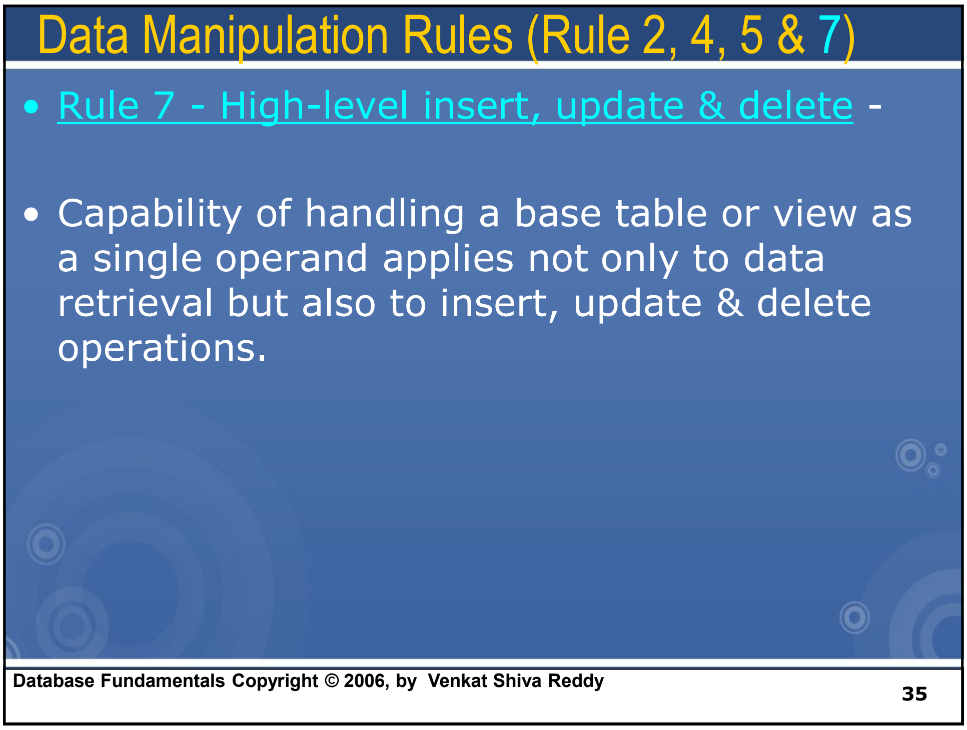
Data Manipulation Rules (Rule 2, 4, 5 & 7)

- Rule 5 - Comprehensive Data Sub-language -
- RDBMS may support many languages & modes of use, but there must be at least ONE language whose statements can express ALL of the following –
  - ➢ Data Definition
  - ➢ View Definition
  - ➢ Data manipulation (interactive via program)
  - ➢ Integrity constraints
  - ➢ Authorization
  - ➢ Transaction boundaries (begin, commit & rollback)

❖ 1992 - ISO standard for SQL provides all these functions

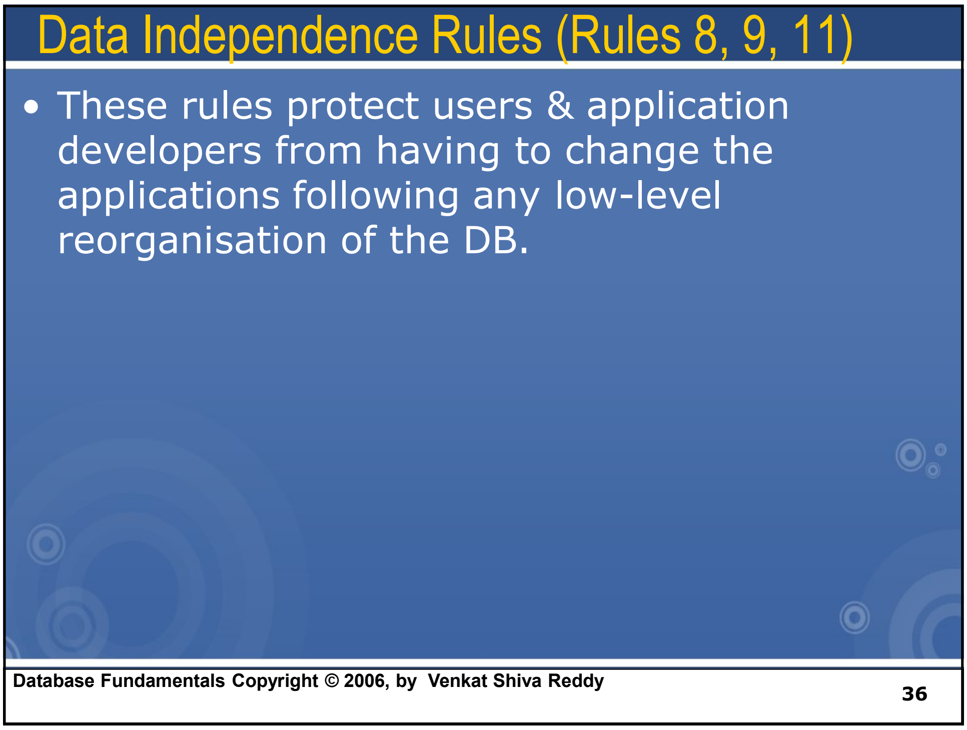
Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

34



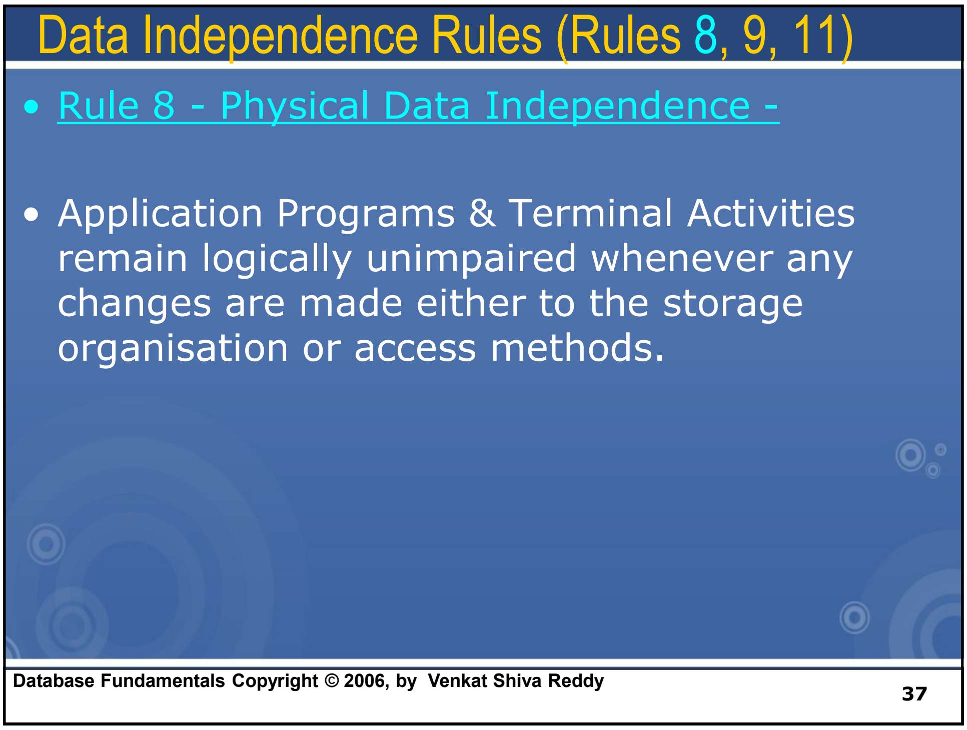
Data Manipulation Rules (Rule 2, 4, 5 & 7)

- Rule 7 - High-level insert, update & delete -
- Capability of handling a base table or view as a single operand applies not only to data retrieval but also to insert, update & delete operations.



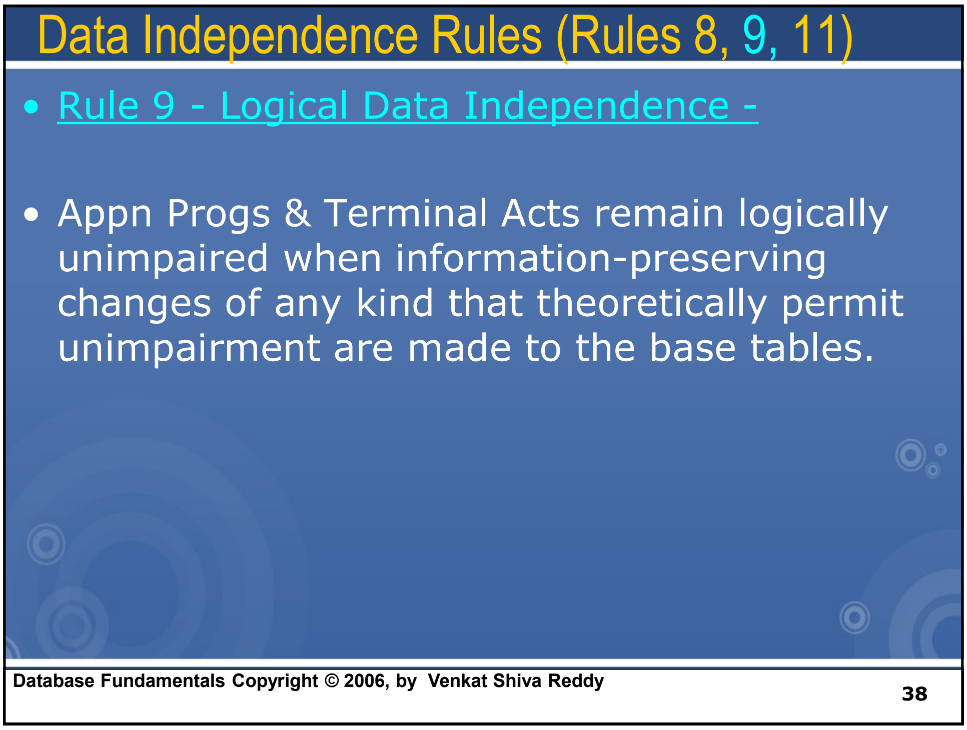
Data Independence Rules (Rules 8, 9, 11)

- These rules protect users & application developers from having to change the applications following any low-level reorganisation of the DB.



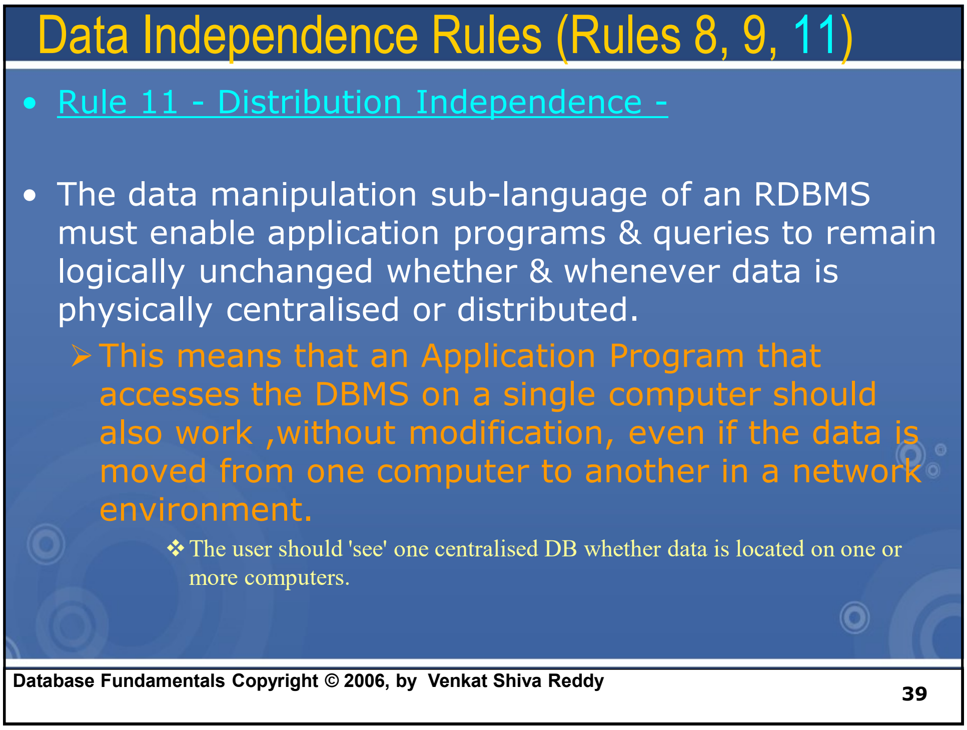
Data Independence Rules (Rules 8, 9, 11)

- Rule 8 - Physical Data Independence -
- Application Programs & Terminal Activities remain logically unimpaired whenever any changes are made either to the storage organisation or access methods.



Data Independence Rules (Rules 8, 9, 11)

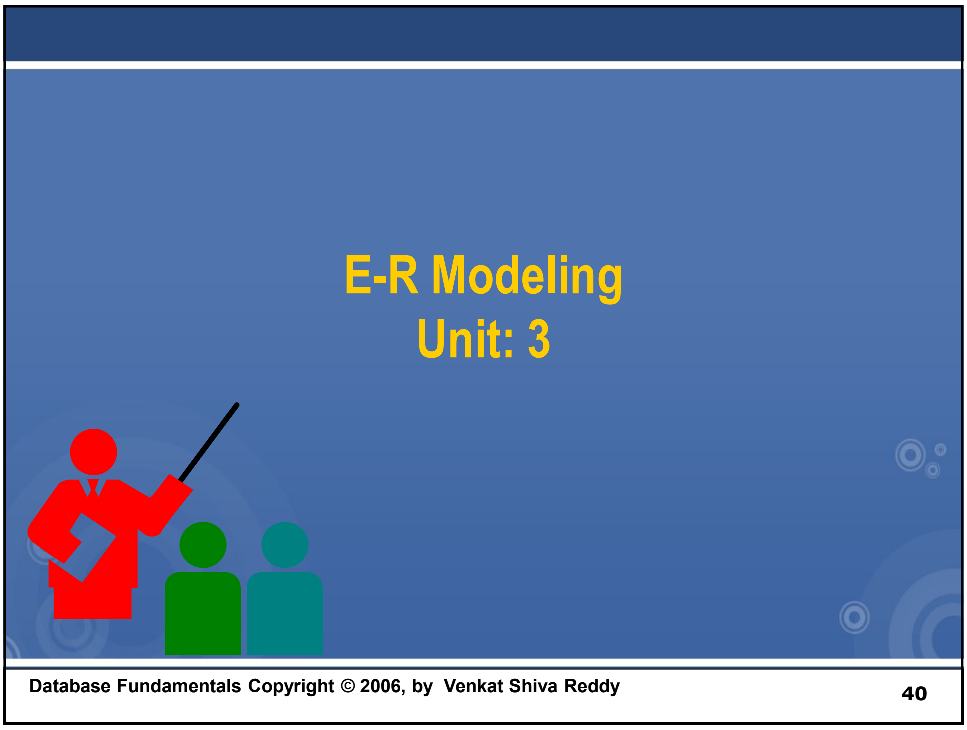
- Rule 9 - Logical Data Independence -
- Appn Progs & Terminal Acts remain logically unimpaired when information-preserving changes of any kind that theoretically permit unimpairment are made to the base tables.



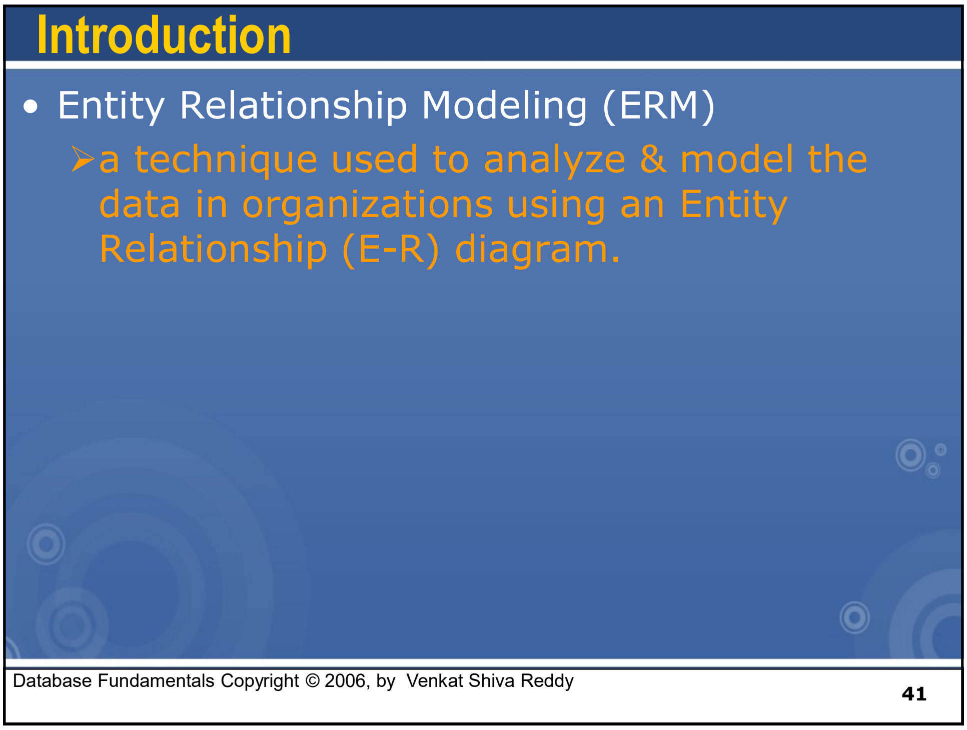
Data Independence Rules (Rules 8, 9, 11)

- Rule 11 - Distribution Independence -

- The data manipulation sub-language of an RDBMS must enable application programs & queries to remain logically unchanged whether & whenever data is physically centralised or distributed.
  - ➢ This means that an Application Program that accesses the DBMS on a single computer should also work ,without modification, even if the data is moved from one computer to another in a network environment.
    - ❖ The user should 'see' one centralised DB whether data is located on one or more computers.



E-R Modeling
Unit: 3

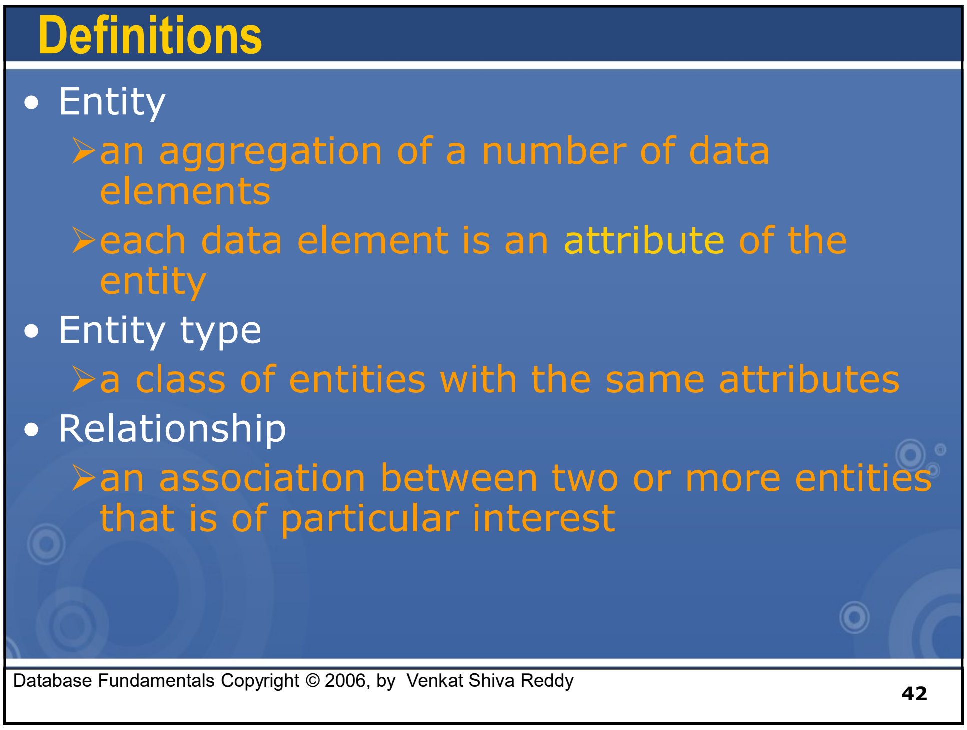


Introduction

- Entity Relationship Modeling (ERM)
  - ➢a technique used to analyze & model the data in organizations using an Entity Relationship (E-R) diagram.

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

41

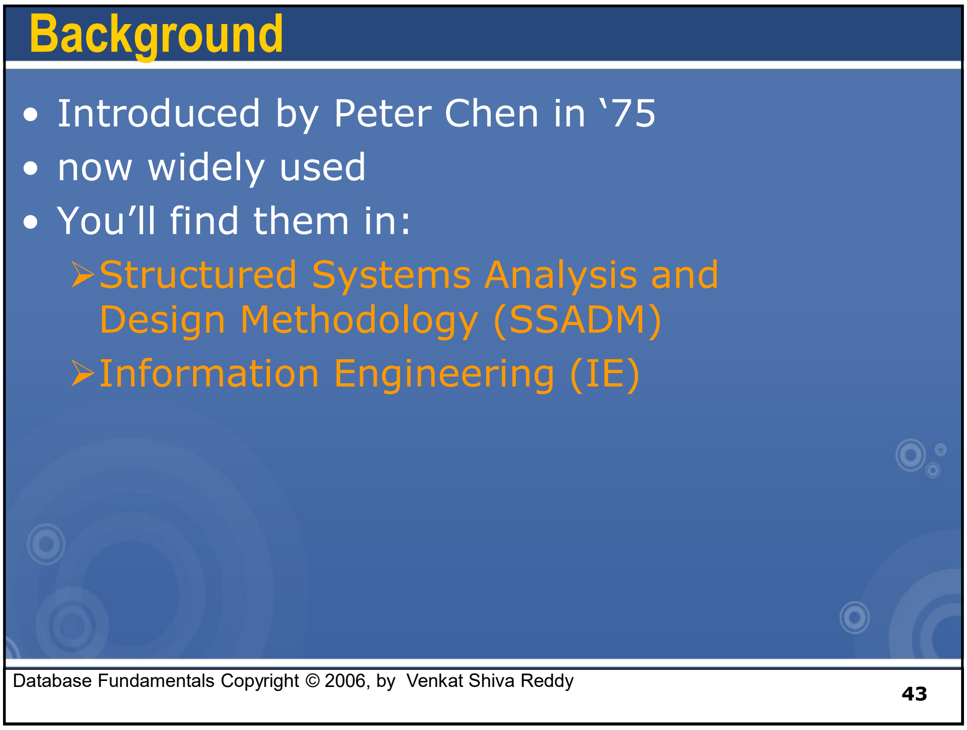


Definitions

- Entity
  - ➢an aggregation of a number of data elements
  - ➢each data element is an attribute of the entity
- Entity type
  - ➢a class of entities with the same attributes
- Relationship
  - ➢an association between two or more entities that is of particular interest

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

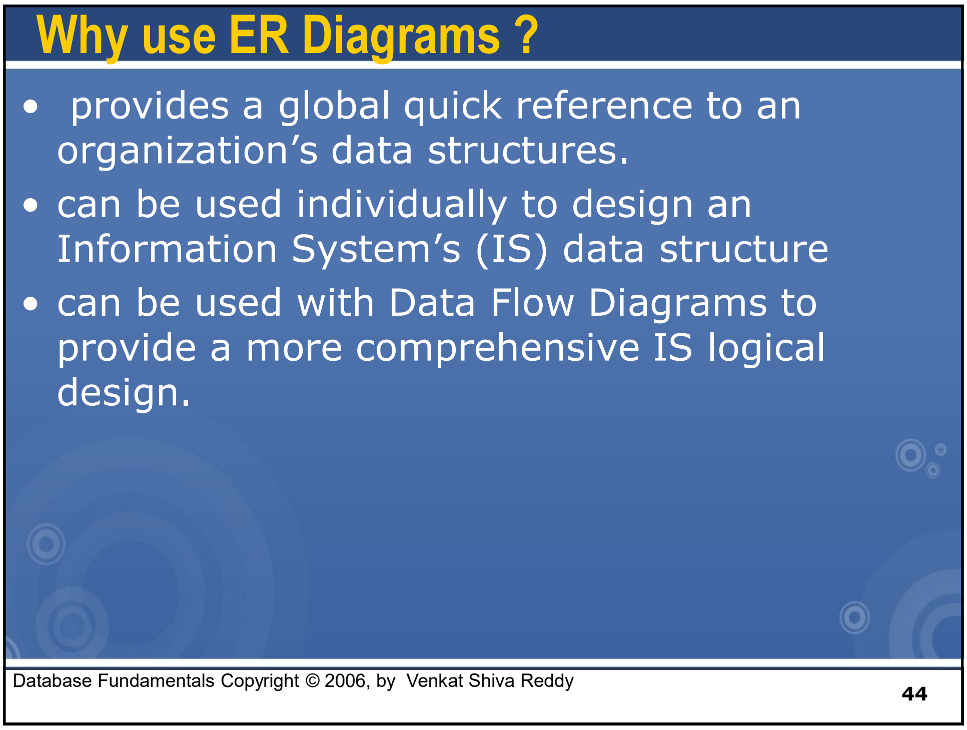
42



Background

- Introduced by Peter Chen in '75
- now widely used
- You'll find them in:
  - ➢Structured Systems Analysis and Design Methodology (SSADM)
  - ➢Information Engineering (IE)

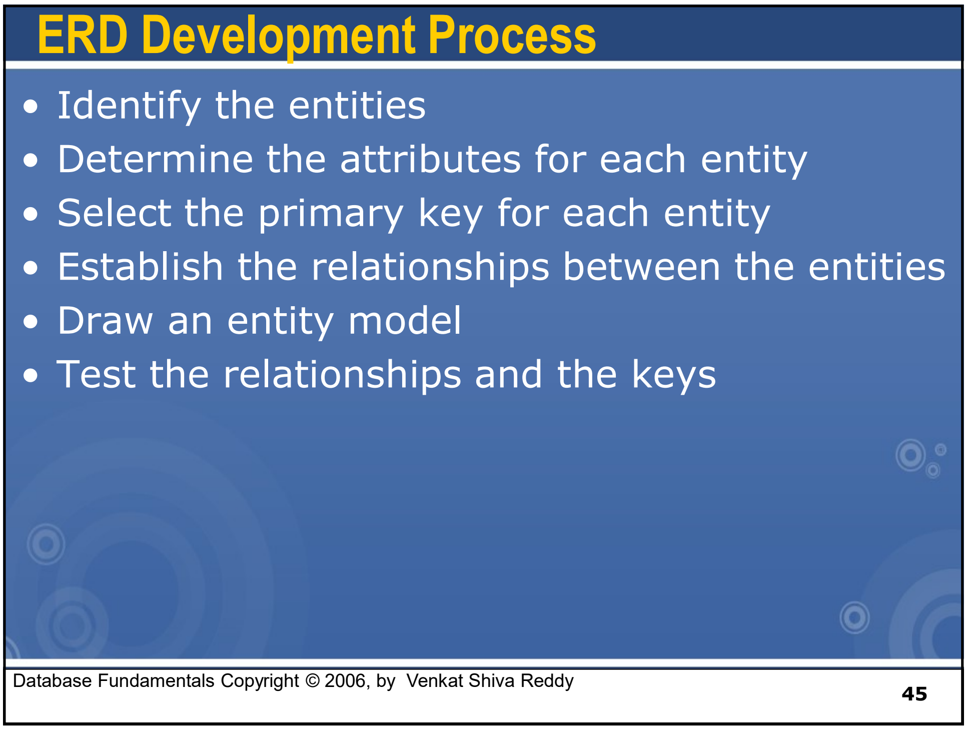
Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy 43



Why use ER Diagrams ?

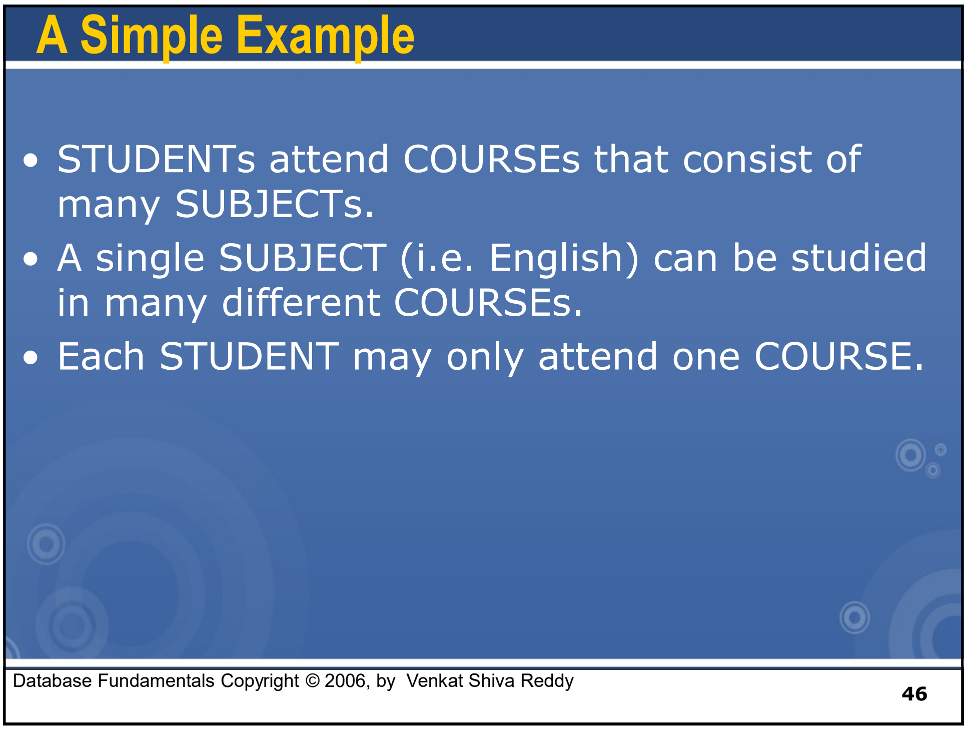
- provides a global quick reference to an organization's data structures.
- can be used individually to design an Information System's (IS) data structure
- can be used with Data Flow Diagrams to provide a more comprehensive IS logical design.

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy 44



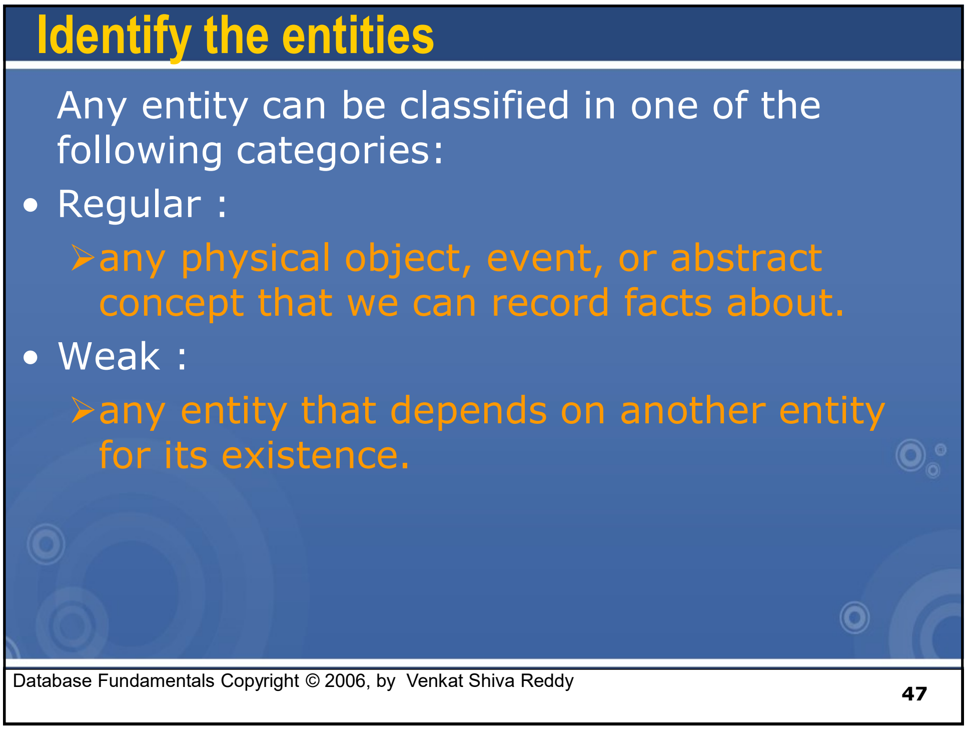
ERD Development Process

- Identify the entities
- Determine the attributes for each entity
- Select the primary key for each entity
- Establish the relationships between the entities
- Draw an entity model
- Test the relationships and the keys



A Simple Example

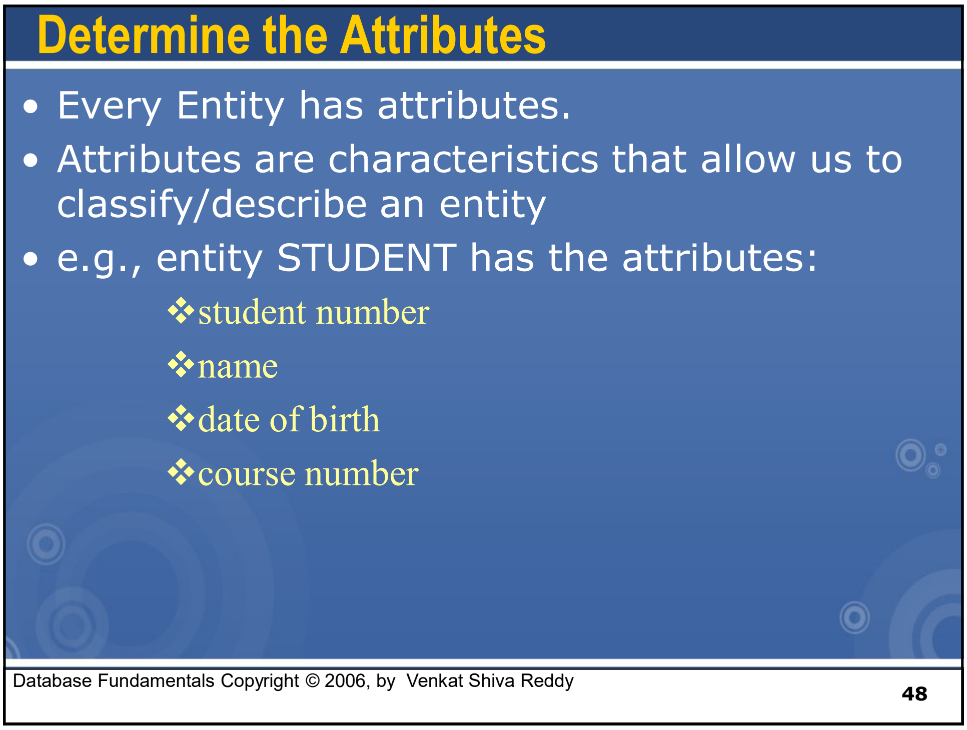
- STUDENTs attend COURSEs that consist of many SUBJECTs.
- A single SUBJECT (i.e. English) can be studied in many different COURSEs.
- Each STUDENT may only attend one COURSE.



Identify the entities

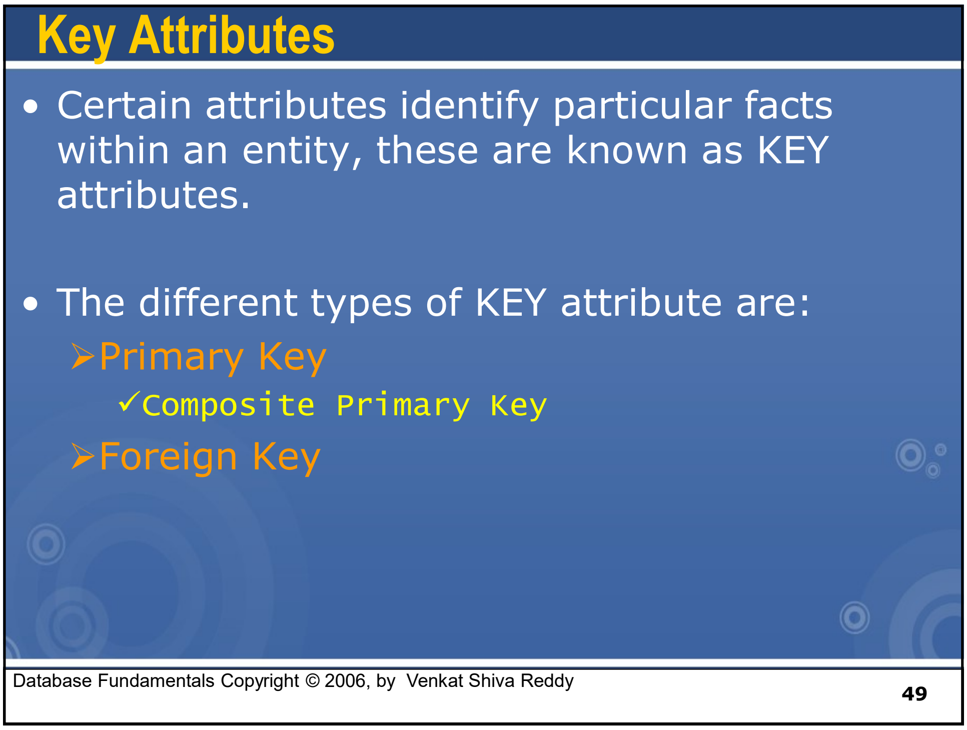
Any entity can be classified in one of the following categories:

- Regular :
  - ➢any physical object, event, or abstract concept that we can record facts about.
- Weak :
  - ➢any entity that depends on another entity for its existence.



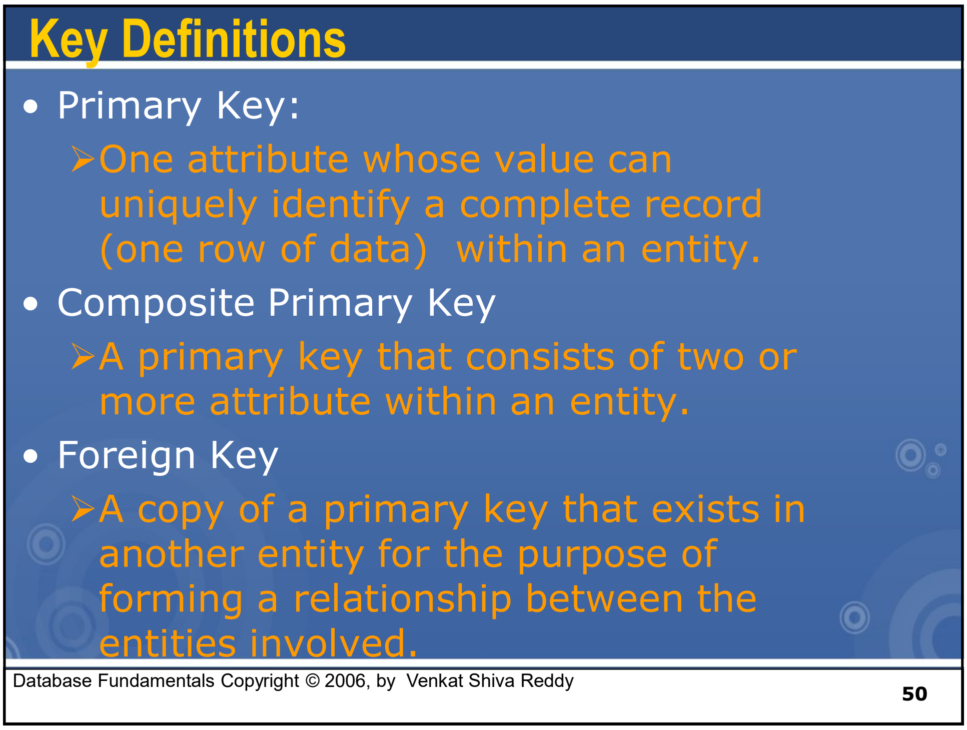
Determine the Attributes

- Every Entity has attributes.
- Attributes are characteristics that allow us to classify/describe an entity
- e.g., entity STUDENT has the attributes:
  - ❖student number
  - ❖name
  - ❖date of birth
  - ❖course number



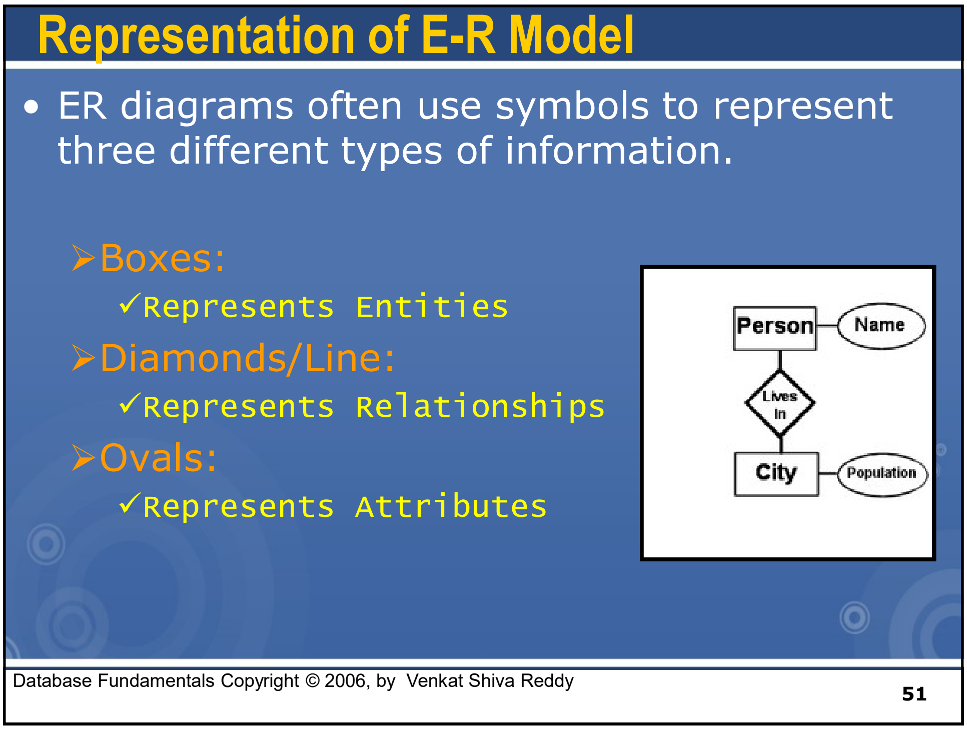
Key Attributes

- Certain attributes identify particular facts within an entity, these are known as KEY attributes.
- The different types of KEY attribute are:
  - ➢Primary Key
    - ✓Composite Primary Key
  - ➢Foreign Key



Key Definitions

- Primary Key:
  - ➢One attribute whose value can uniquely identify a complete record (one row of data) within an entity.
- Composite Primary Key
  - ➢A primary key that consists of two or more attribute within an entity.
- Foreign Key
  - ➢A copy of a primary key that exists in another entity for the purpose of forming a relationship between the entities involved.



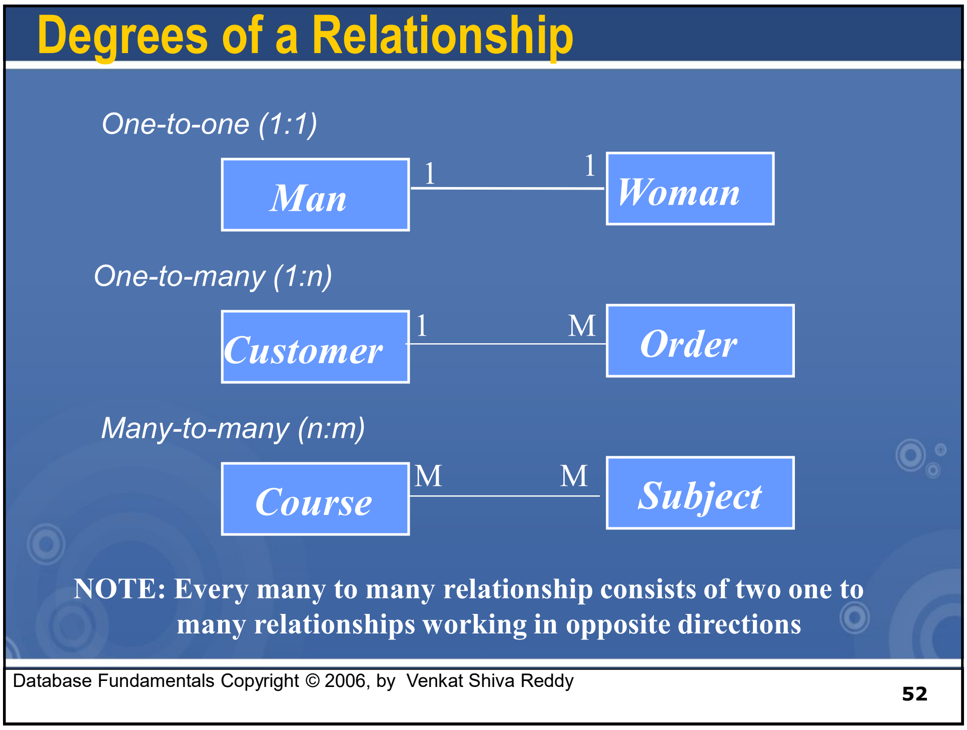
Representation of E-R Model

- ER diagrams often use symbols to represent three different types of information.

  - ➢Boxes:
    - ✓Represents Entities
  - ➢Diamonds/Line:
    - ✓Represents Relationships
  - ➢Ovals:
    - ✓Represents Attributes

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

51



Degrees of a Relationship

One-to-one (1:1)

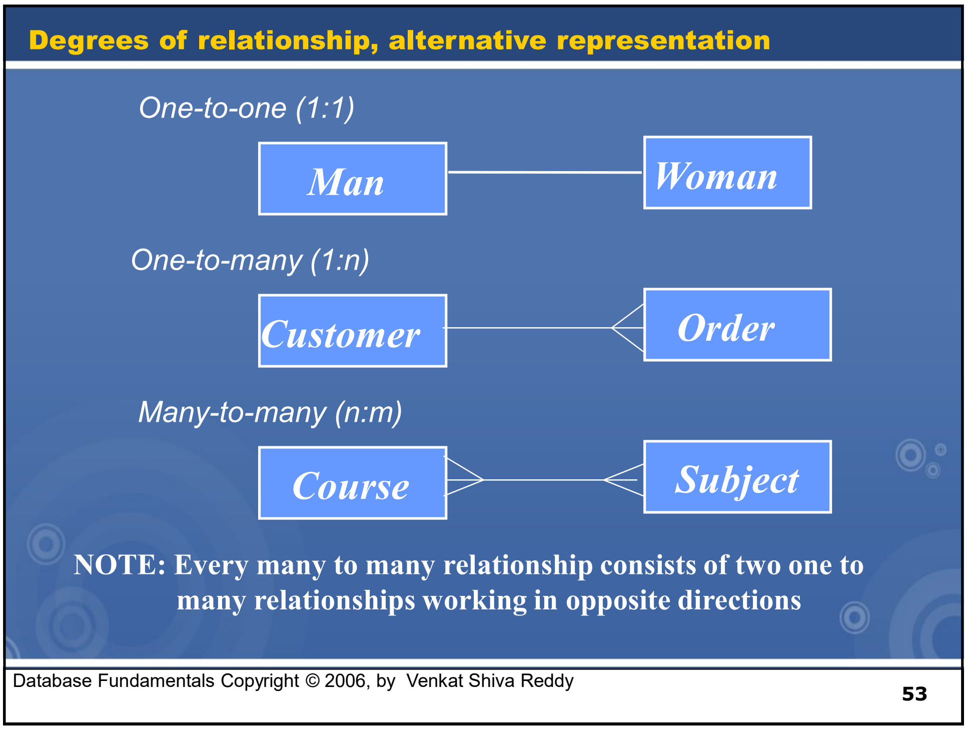
One-to-many (1:n)

Many-to-many (n:m)

NOTE: Every many to many relationship consists of two one to many relationships working in opposite directions

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

52



Degrees of relationship, alternative representation

One-to-one (1:1)

Man — **Woman**

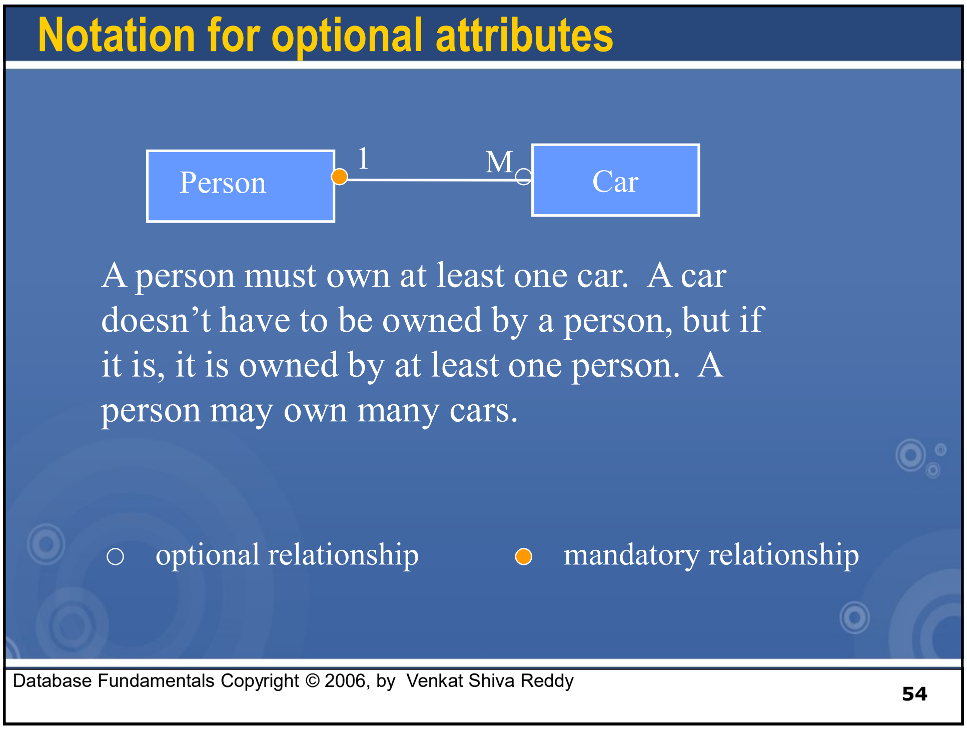
One-to-many (1:n)

Customer — **Order**

Many-to-many (n:m)

Course — **Subject**

NOTE: Every many to many relationship consists of two one to many relationships working in opposite directions

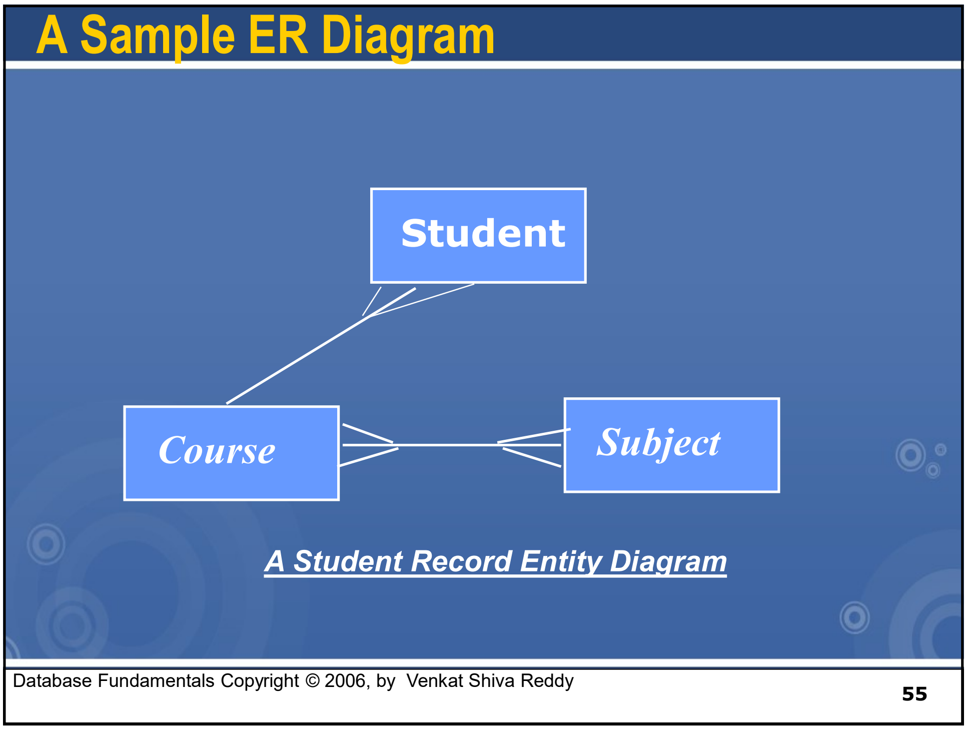


Notation for optional attributes

Person 1 — M Car

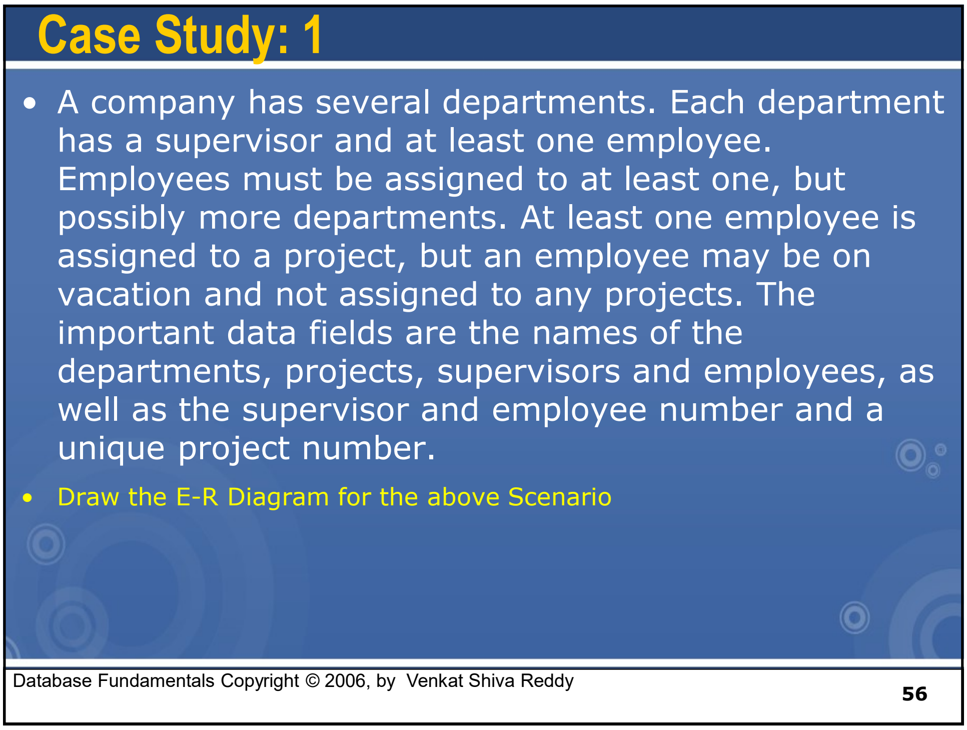
A person must own at least one car. A car doesn't have to be owned by a person, but if it is, it is owned by at least one person. A person may own many cars.

○ optional relationship ● mandatory relationship



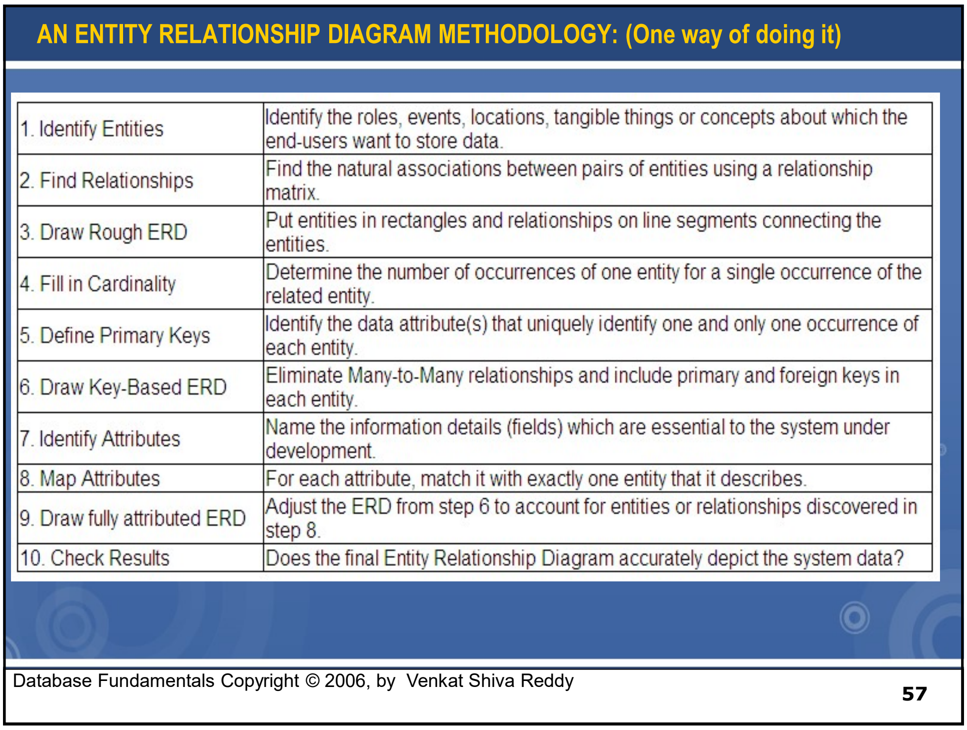
A Sample ER Diagram

A Student Record Entity Diagram



Case Study: 1

- A company has several departments. Each department has a supervisor and at least one employee. Employees must be assigned to at least one, but possibly more departments. At least one employee is assigned to a project, but an employee may be on vacation and not assigned to any projects. The important data fields are the names of the departments, projects, supervisors and employees, as well as the supervisor and employee number and a unique project number.
- Draw the E-R Diagram for the above Scenario

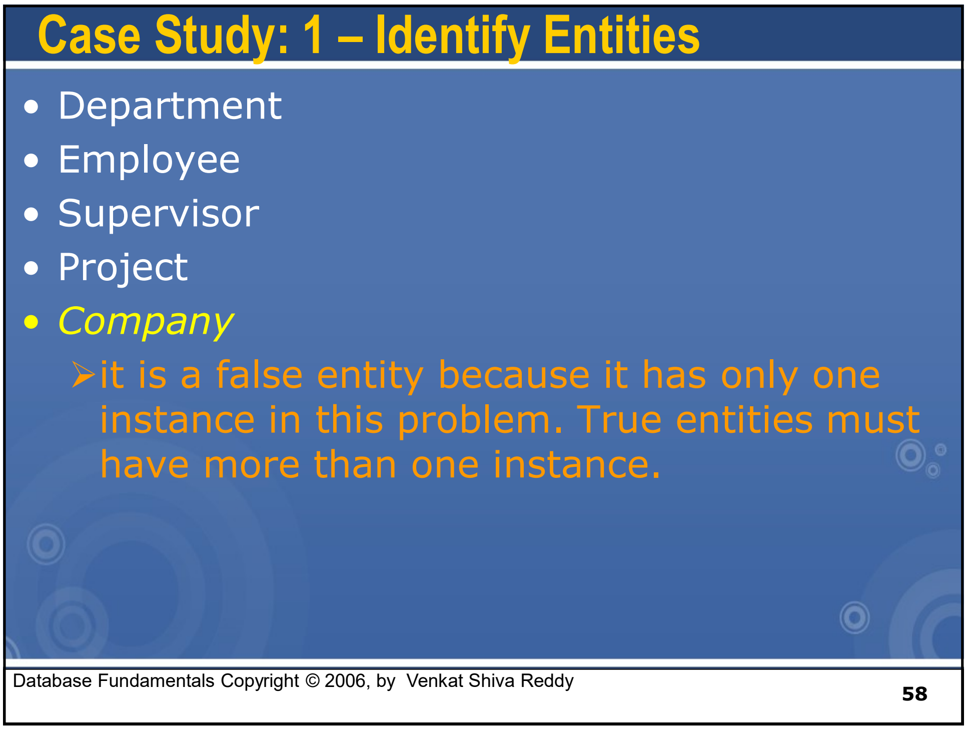


AN ENTITY RELATIONSHIP DIAGRAM METHODOLOGY: (One way of doing it)

1. Identify Entities	Identify the roles, events, locations, tangible things or concepts about which the end-users want to store data.
2. Find Relationships	Find the natural associations between pairs of entities using a relationship matrix.
3. Draw Rough ERD	Put entities in rectangles and relationships on line segments connecting the entities.
4. Fill in Cardinality	Determine the number of occurrences of one entity for a single occurrence of the related entity.
5. Define Primary Keys	Identify the data attribute(s) that uniquely identify one and only one occurrence of each entity.
6. Draw Key-Based ERD	Eliminate Many-to-Many relationships and include primary and foreign keys in each entity.
7. Identify Attributes	Name the information details (fields) which are essential to the system under development.
8. Map Attributes	For each attribute, match it with exactly one entity that it describes.
9. Draw fully attributed ERD	Adjust the ERD from step 6 to account for entities or relationships discovered in step 8.
10. Check Results	Does the final Entity Relationship Diagram accurately depict the system data?

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

57

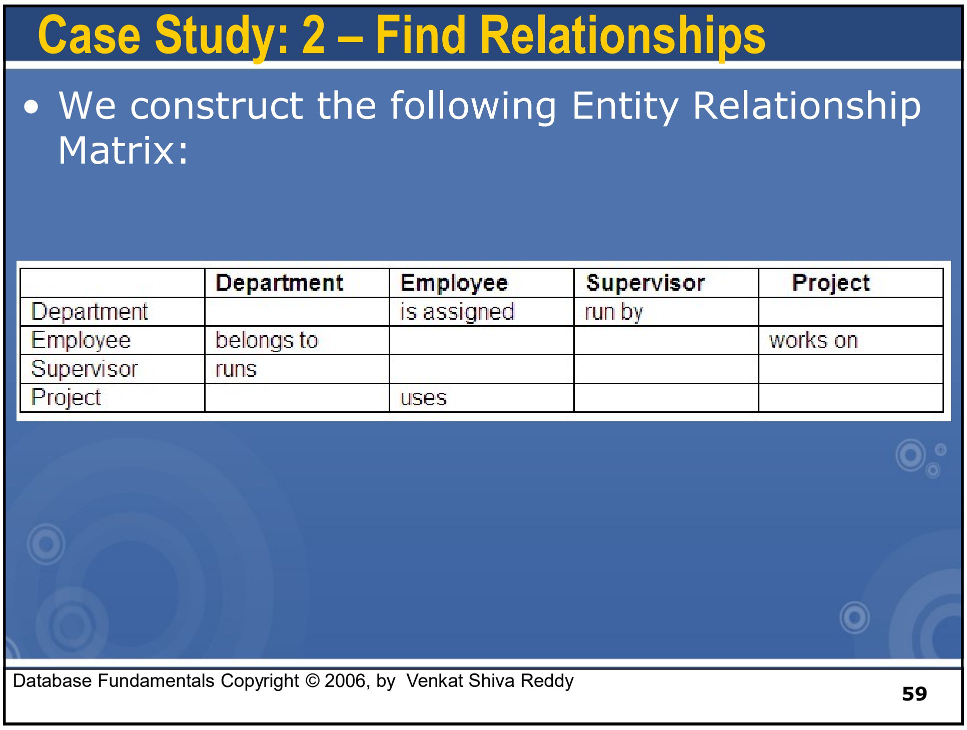


Case Study: 1 – Identify Entities

- Department
- Employee
- Supervisor
- Project
- *Company*
  - ➢it is a false entity because it has only one instance in this problem. True entities must have more than one instance.

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

58



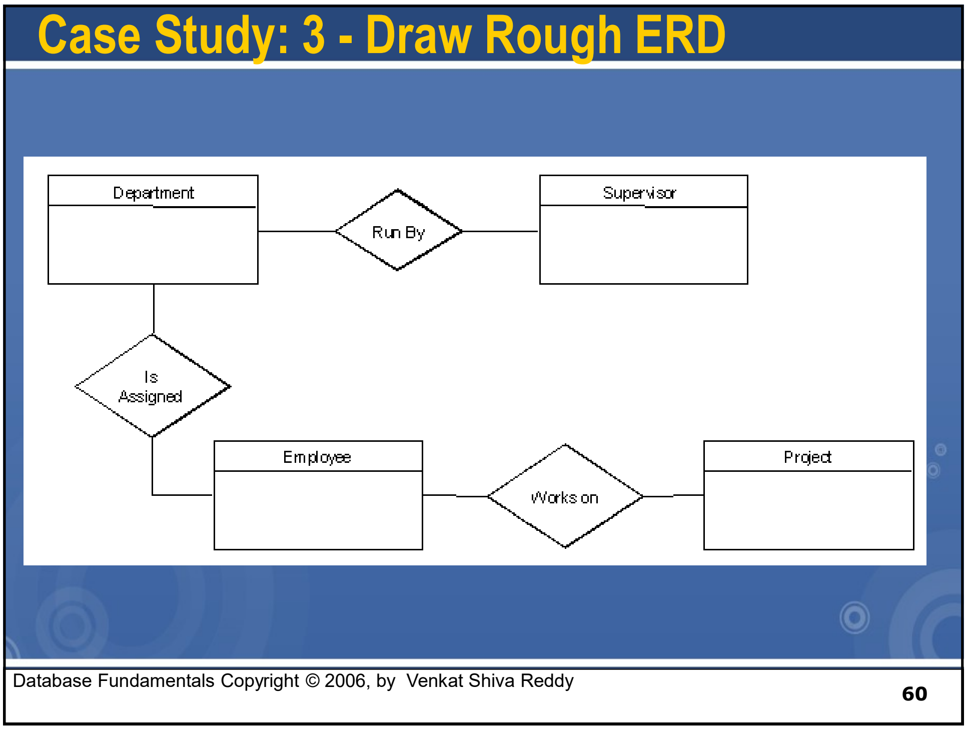
Case Study: 2 – Find Relationships

- We construct the following Entity Relationship Matrix:

	Department	Employee	Supervisor	Project
Department		is assigned	run by	
Employee	belongs to			works on
Supervisor	runs			
Project		uses		

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

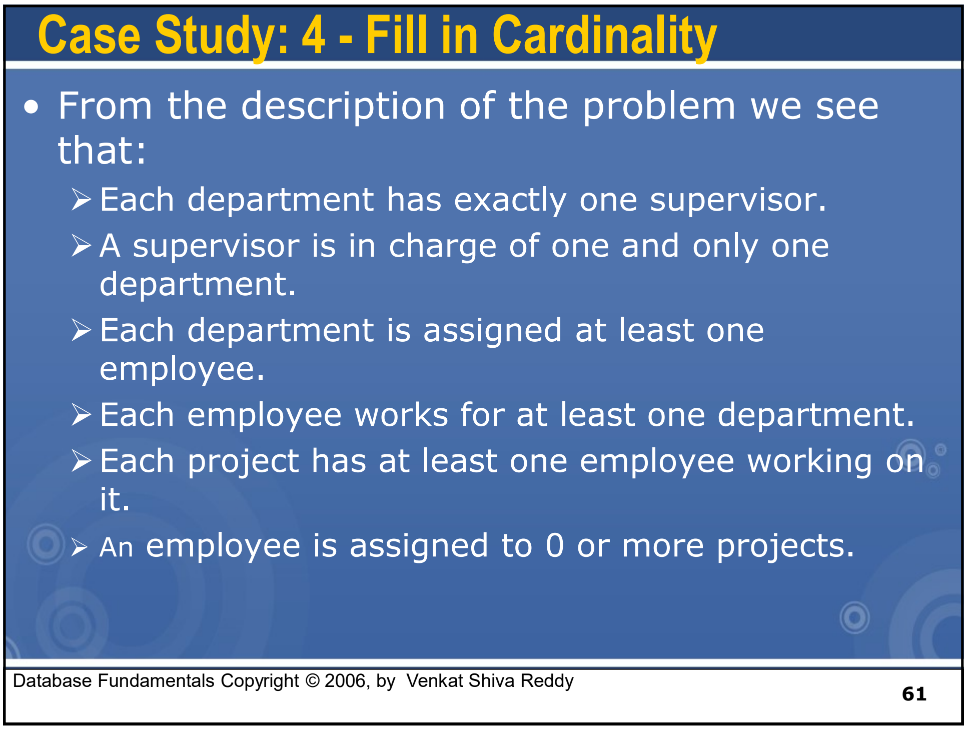
59



Case Study: 3 - Draw Rough ERD

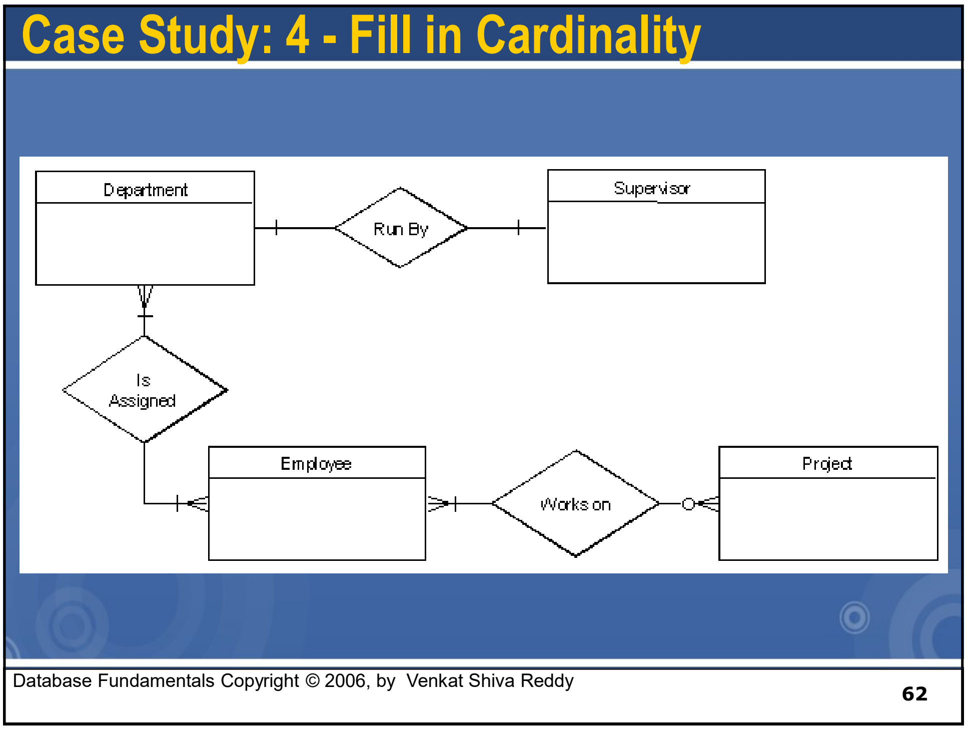
Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

60



Case Study: 4 - Fill in Cardinality

- From the description of the problem we see that:
  - Each department has exactly one supervisor.
  - A supervisor is in charge of one and only one department.
  - Each department is assigned at least one employee.
  - Each employee works for at least one department.
  - Each project has at least one employee working on it.
  - An employee is assigned to 0 or more projects.

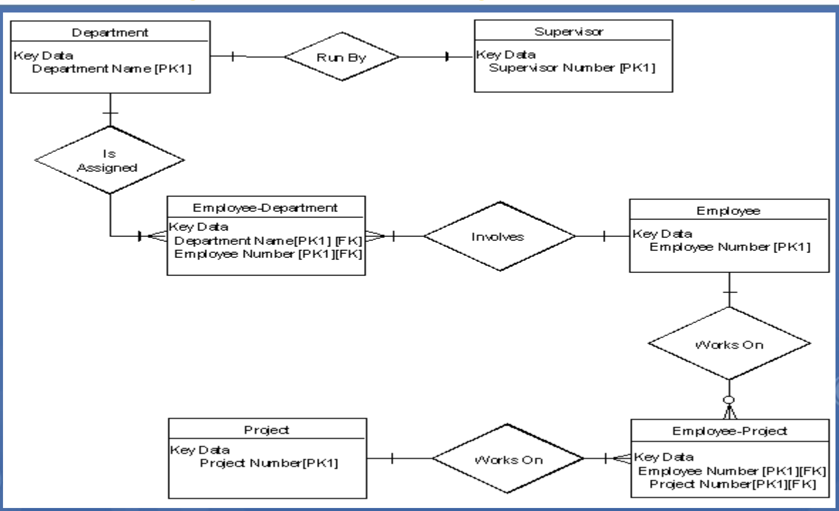


Case Study: 4 - Fill in Cardinality

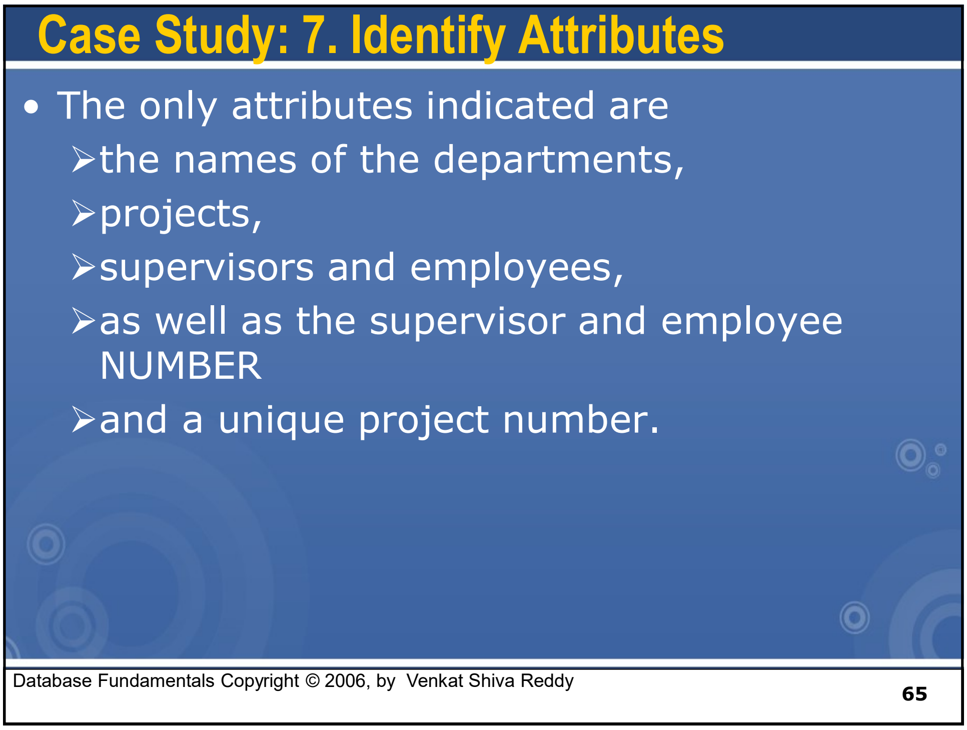
Case Study: 5 - Define Primary Keys

- The primary keys are
  - ➢Department Name,
  - ➢Supervisor Number,
  - ➢Employee Number,
  - ➢Project Number.

Case Study: 6 - Draw Key-Based ERD



There are two many-to-many relationships in the rough ERD above, between Department and Employee and between Employee and Project. Thus we need the associative entities Department-Employee and Employee-Project. The primary key for Department-Employee is the concatenated key Department Name and Employee Number. The primary key for Employee-Project is the concatenated key Employee Number and Project Number.

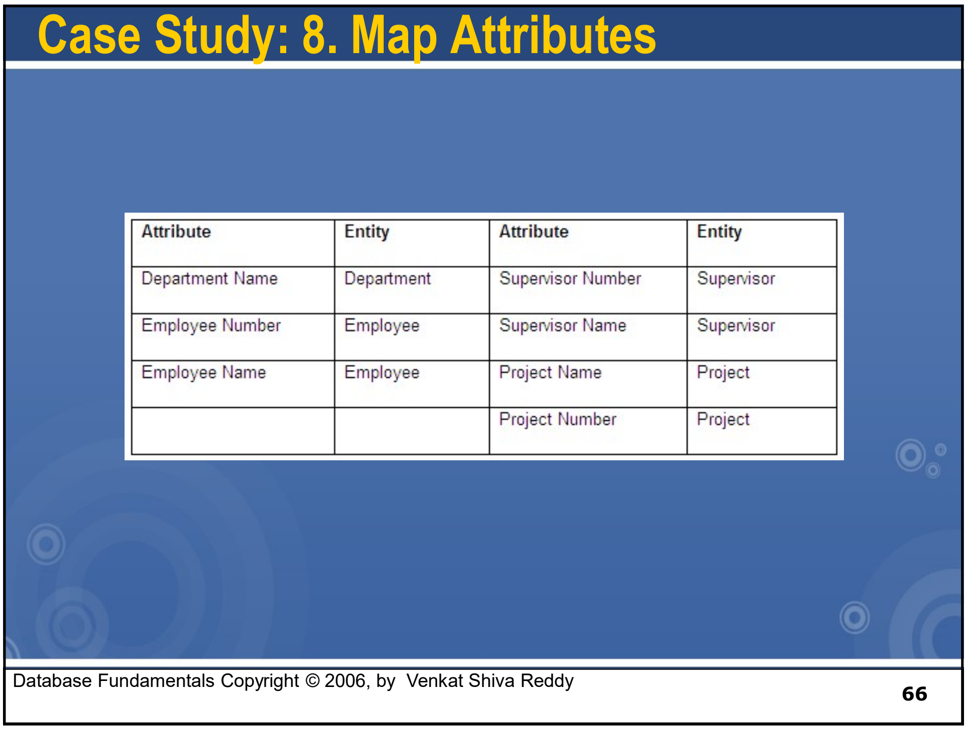


Case Study: 7. Identify Attributes

- The only attributes indicated are
  - the names of the departments,
  - projects,
  - supervisors and employees,
  - as well as the supervisor and employee NUMBER
  - and a unique project number.

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

65



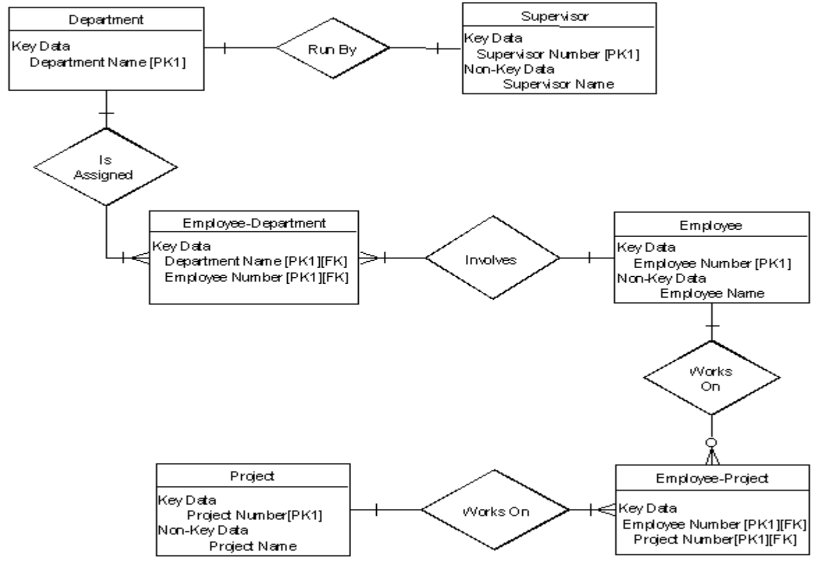
Case Study: 8. Map Attributes

Attribute	Entity	Attribute	Entity
Department Name	Department	Supervisor Number	Supervisor
Employee Number	Employee	Supervisor Name	Supervisor
Employee Name	Employee	Project Name	Project
		Project Number	Project

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

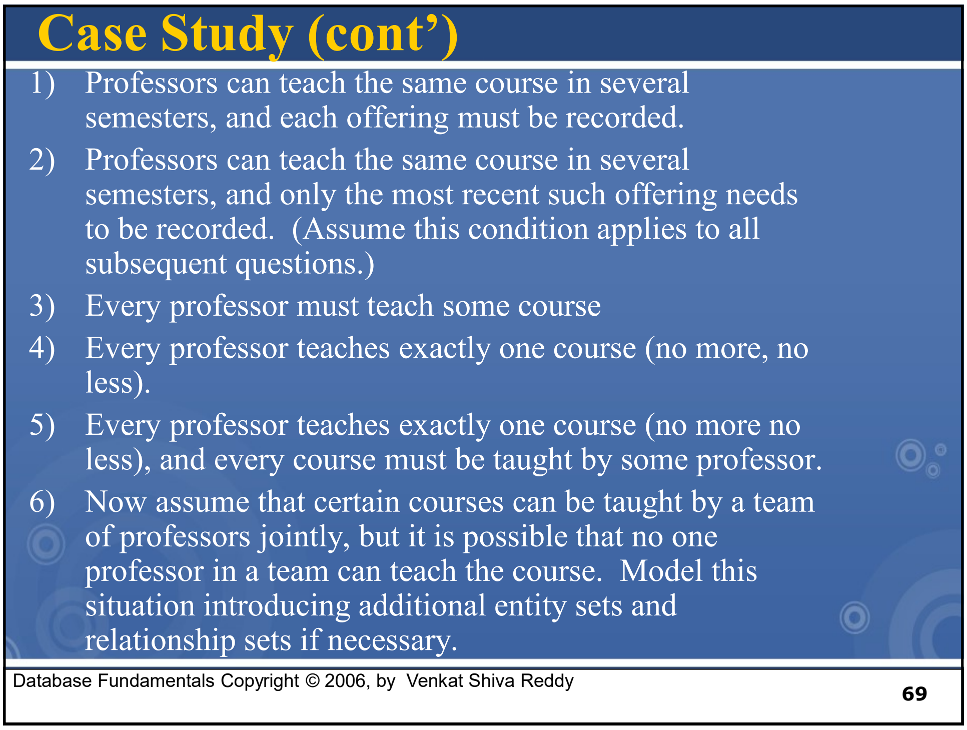
66

Case Study: 9. Draw Fully Attributed ERD



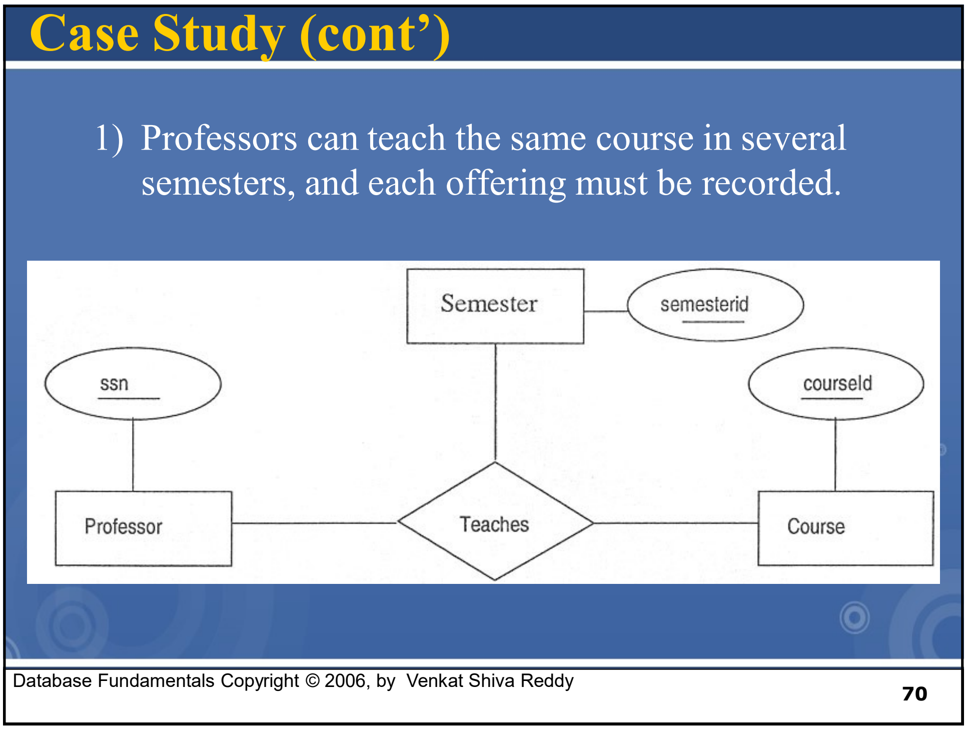
Case Study: 2

- A University database contains information about **professors** (identified by social security number, or *SSN*) and **courses** (identified by *courseid*). Professors teach courses.
- Each of the following situations concerns the **Teaches** relationship set. For each situation, draw an ER diagram that describes it (assuming that no further constraints hold).



Case Study (cont')

1) Professors can teach the same course in several semesters, and each offering must be recorded.
2) Professors can teach the same course in several semesters, and only the most recent such offering needs to be recorded. (Assume this condition applies to all subsequent questions.)
3) Every professor must teach some course
4) Every professor teaches exactly one course (no more, no less).
5) Every professor teaches exactly one course (no more no less), and every course must be taught by some professor.
6) Now assume that certain courses can be taught by a team of professors jointly, but it is possible that no one professor in a team can teach the course. Model this situation introducing additional entity sets and relationship sets if necessary.

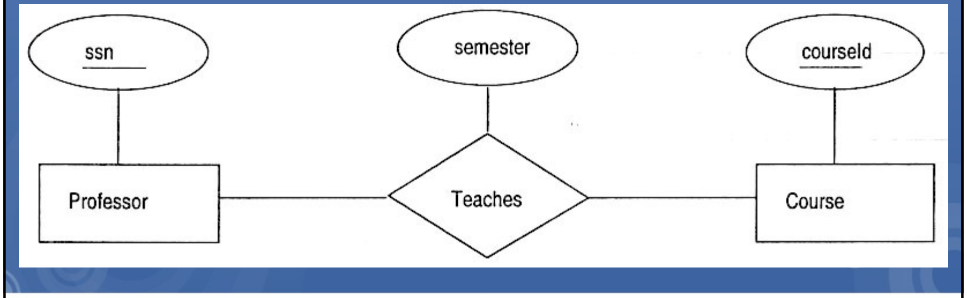


Case Study (cont')

1) Professors can teach the same course in several semesters, and each offering must be recorded.

Case Study (cont')

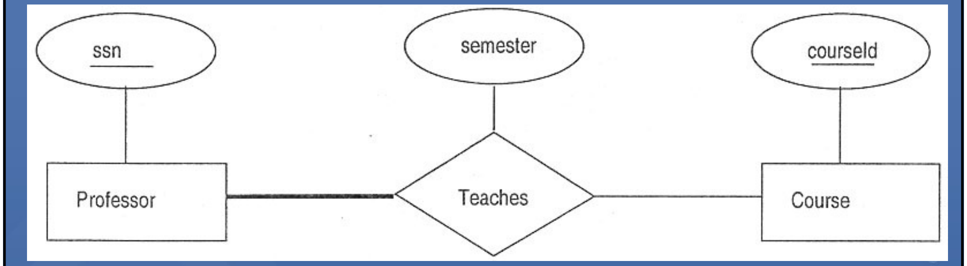
2) Professors can teach the same course in several semesters, and only the most recent such offering needs to be recorded. (Assume this condition applies to all subsequent questions.)



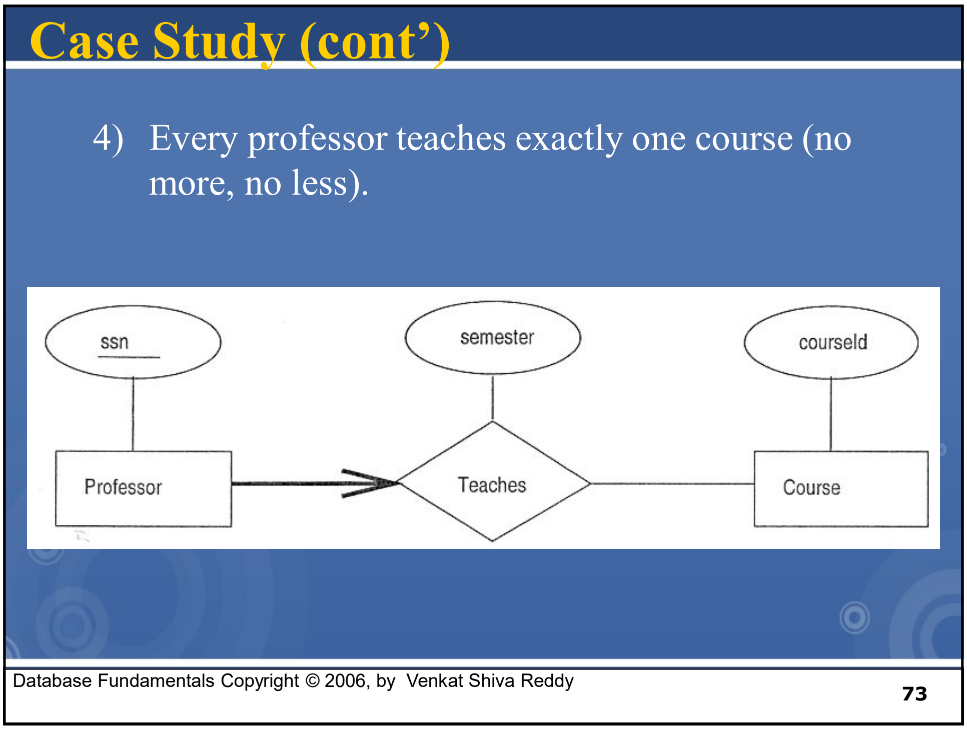
Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

Case Study (cont')

3) Every professor must teach some course

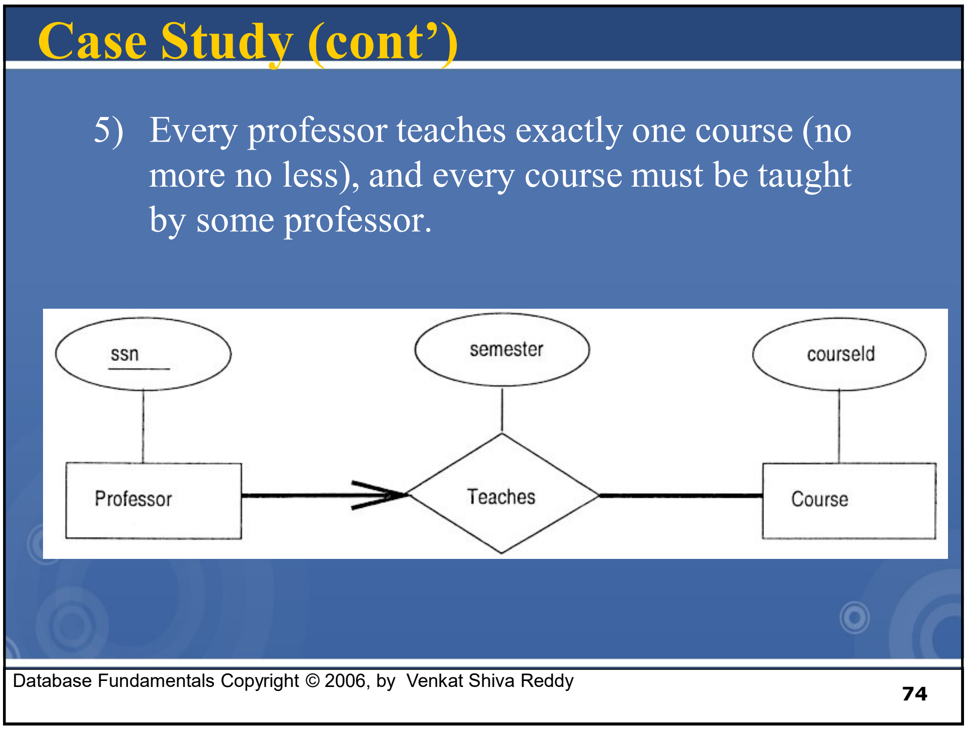


Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy



Case Study (cont')

4) Every professor teaches exactly one course (no more, no less).

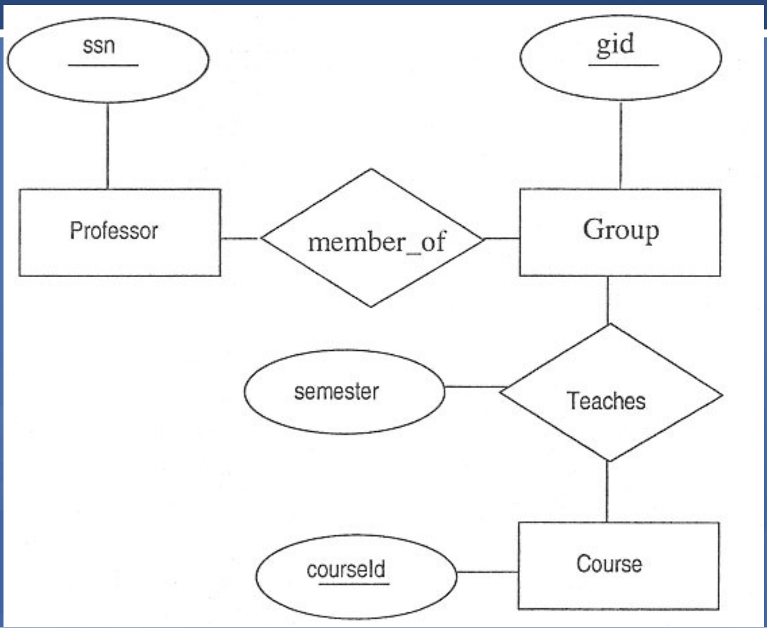


Case Study (cont')

5) Every professor teaches exactly one course (no more no less), and every course must be taught by some professor.

Case Study (cont')

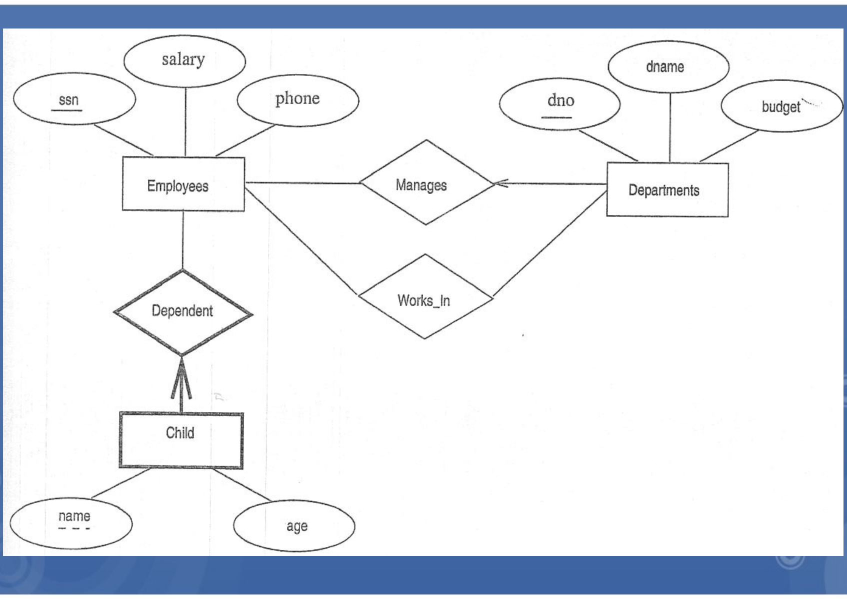
6) Now assume that certain courses can be taught by a team of professors jointly, but it is possible that no one professor in a team can teach the course. Model this situation introducing additional entity sets and relationship sets if necessary.

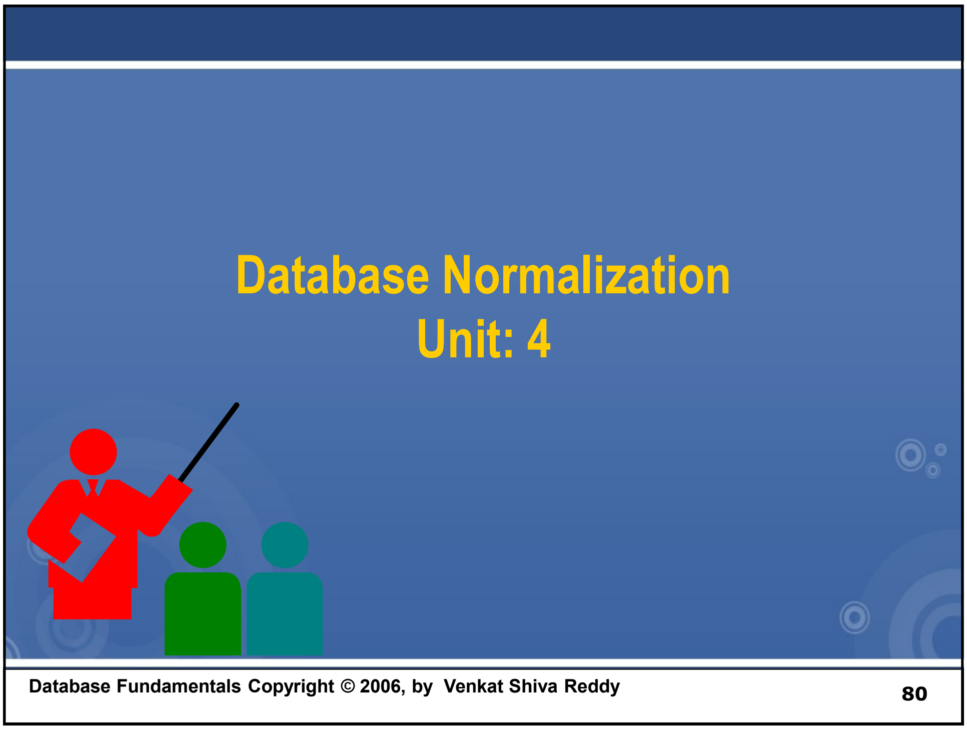


Exercise: 1

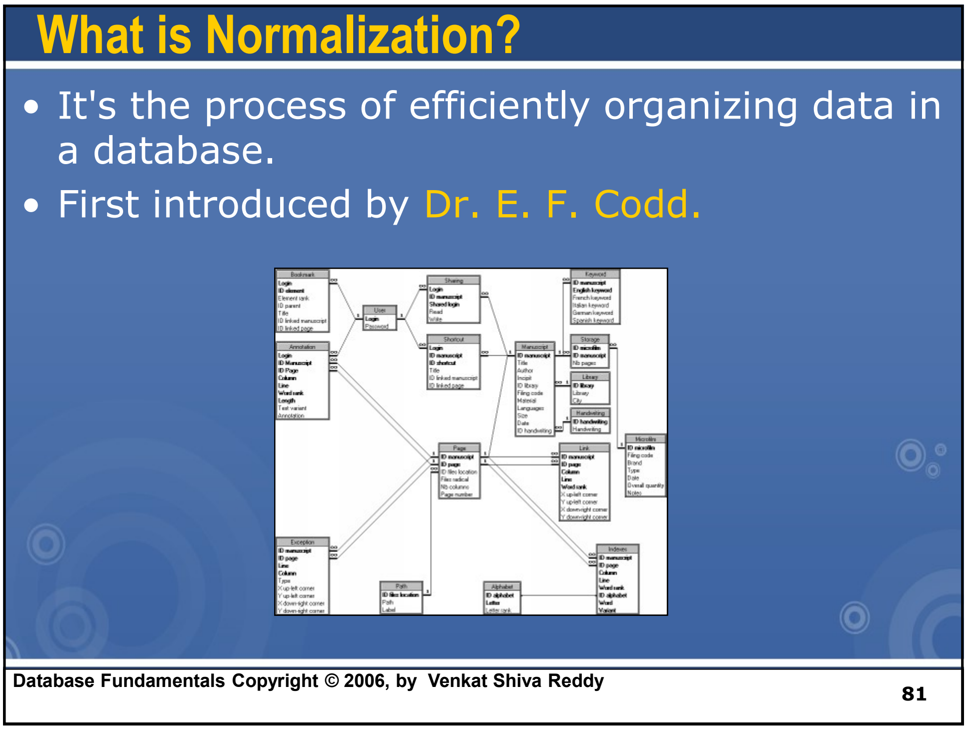
- A company database needs to store information about employees (identifyied by ssn, with salary and phone as attributes); departments (identified by dno, with dname and budget as attributes); and children of employees (with name and age as attributes).
- Employees work in departments; each department is managed by an employee; a child must be identified uniquely by name when the parent (who is an employee; assume that only one parent works for the company) is known.
- We are not interested in information about a child once the parent leaves the company.

Exercise: 1 - Solution



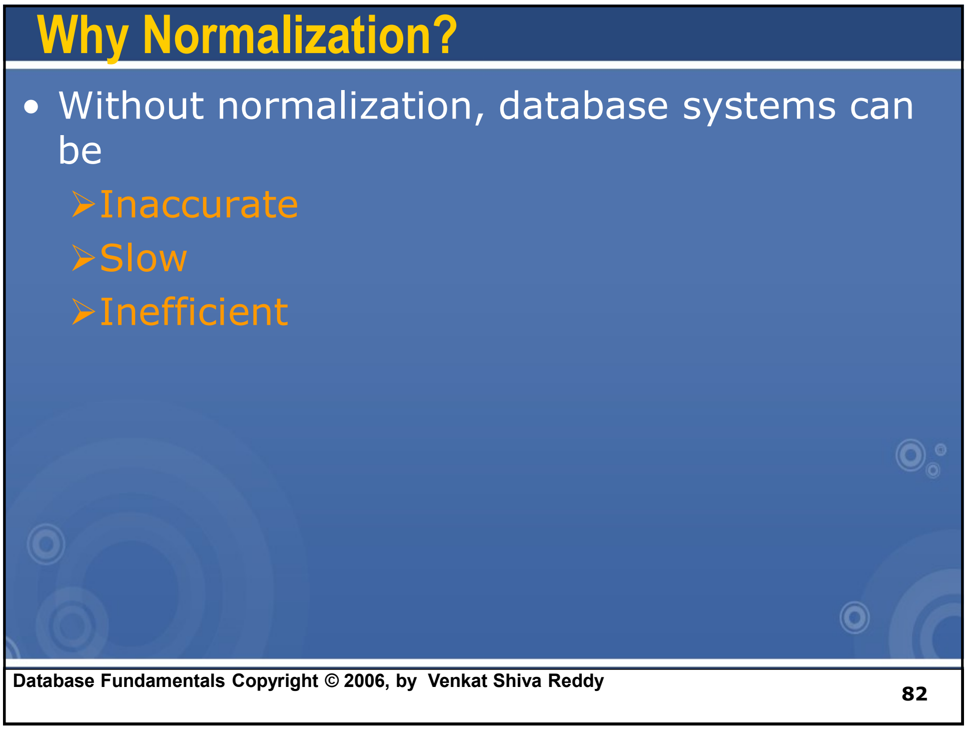


Database Normalization
Unit: 4



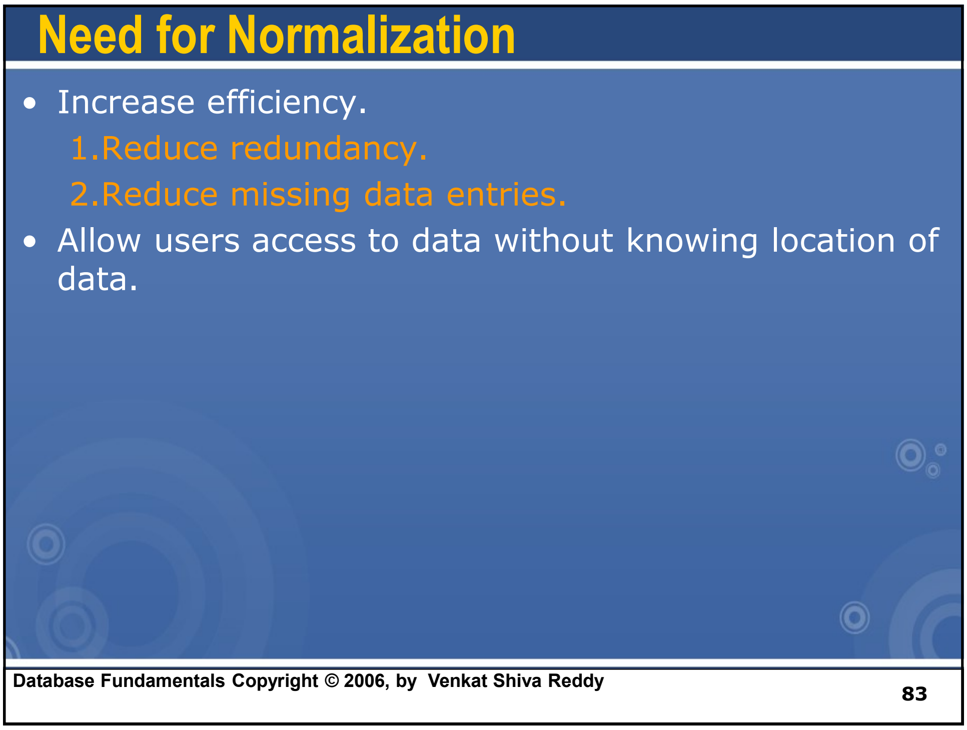
What is Normalization?

- It's the process of efficiently organizing data in a database.
- First introduced by Dr. E. F. Codd.



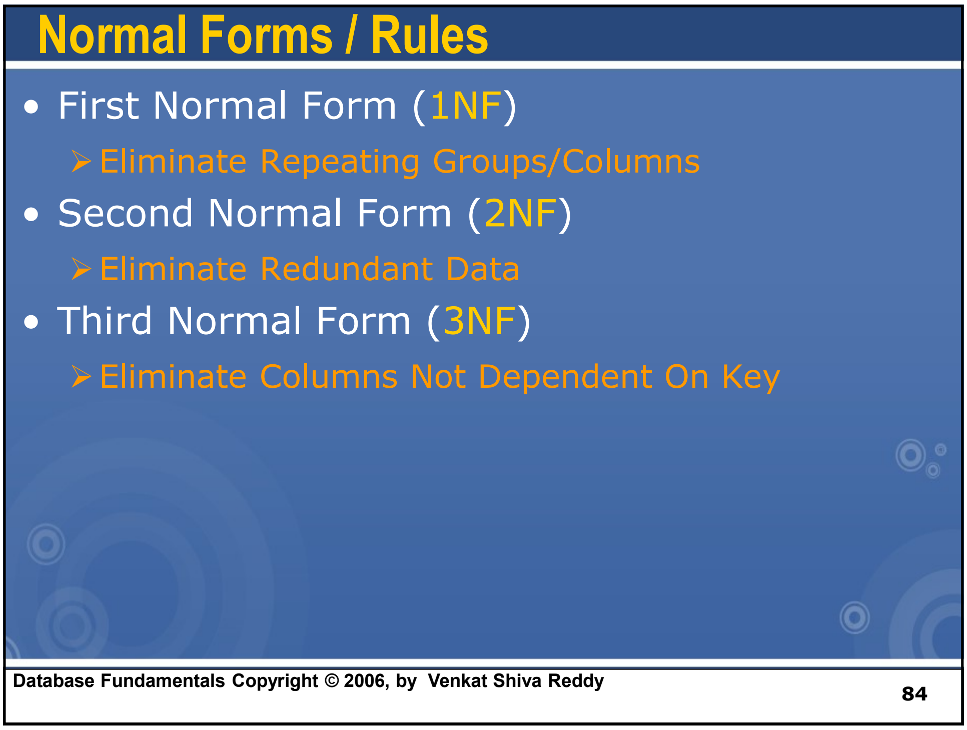
Why Normalization?

- Without normalization, database systems can be
  - ➢Inaccurate
  - ➢Slow
  - ➢Inefficient



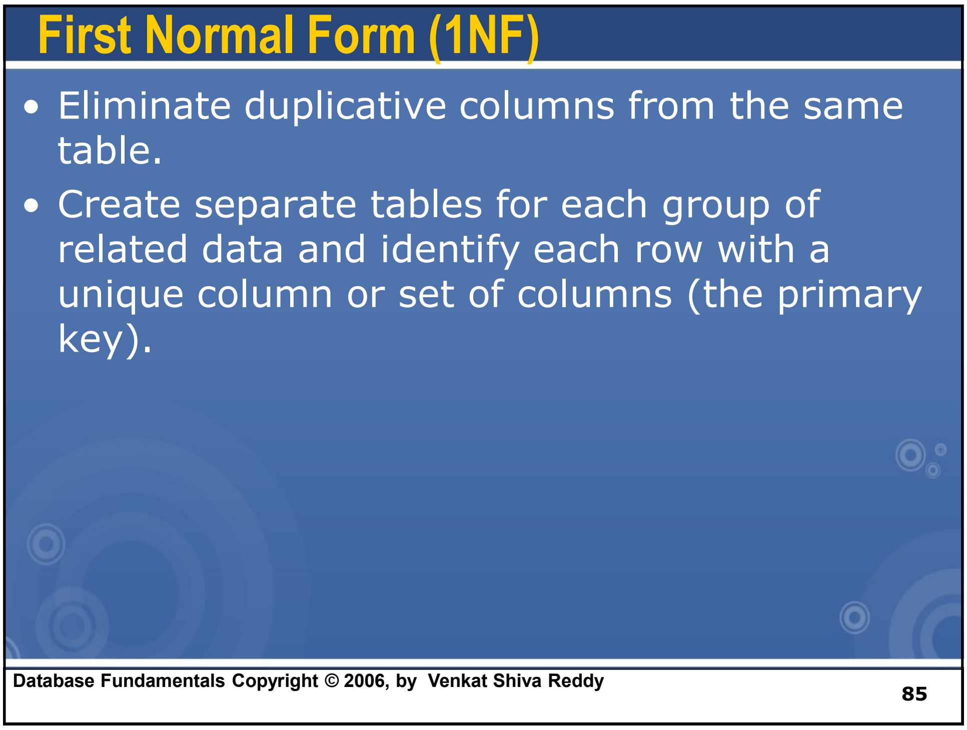
Need for Normalization

- Increase efficiency.
  1. Reduce redundancy.
  2. Reduce missing data entries.
- Allow users access to data without knowing location of data.



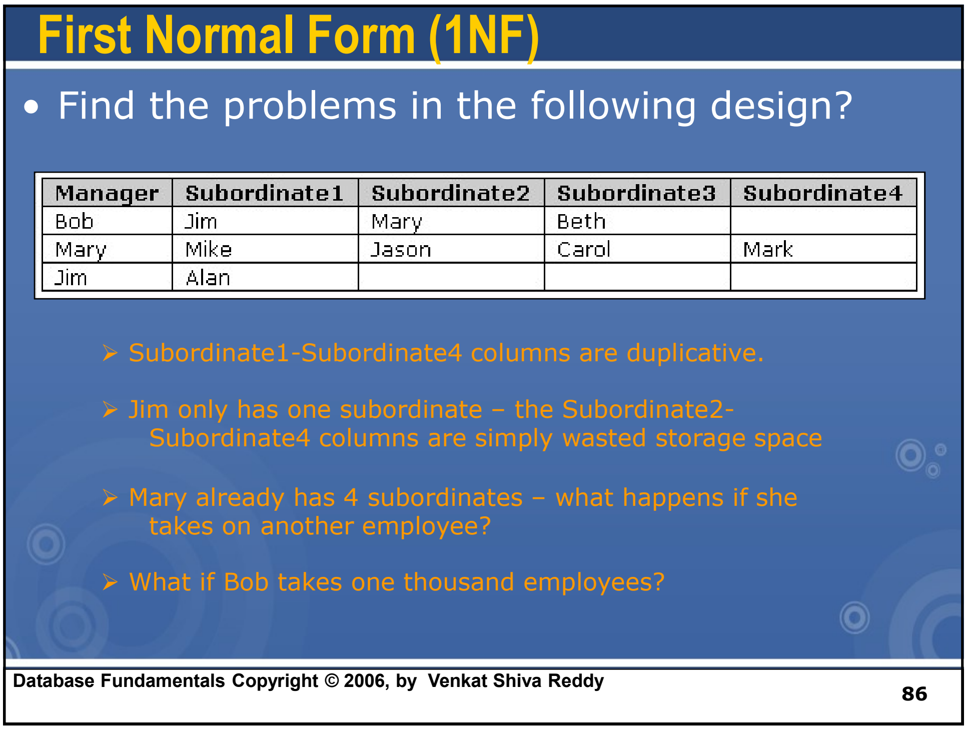
Normal Forms / Rules

- First Normal Form (1NF)
  - ➢ Eliminate Repeating Groups/Columns
- Second Normal Form (2NF)
  - ➢ Eliminate Redundant Data
- Third Normal Form (3NF)
  - ➢ Eliminate Columns Not Dependent On Key



First Normal Form (1NF)

- Eliminate duplicative columns from the same table.
- Create separate tables for each group of related data and identify each row with a unique column or set of columns (the primary key).

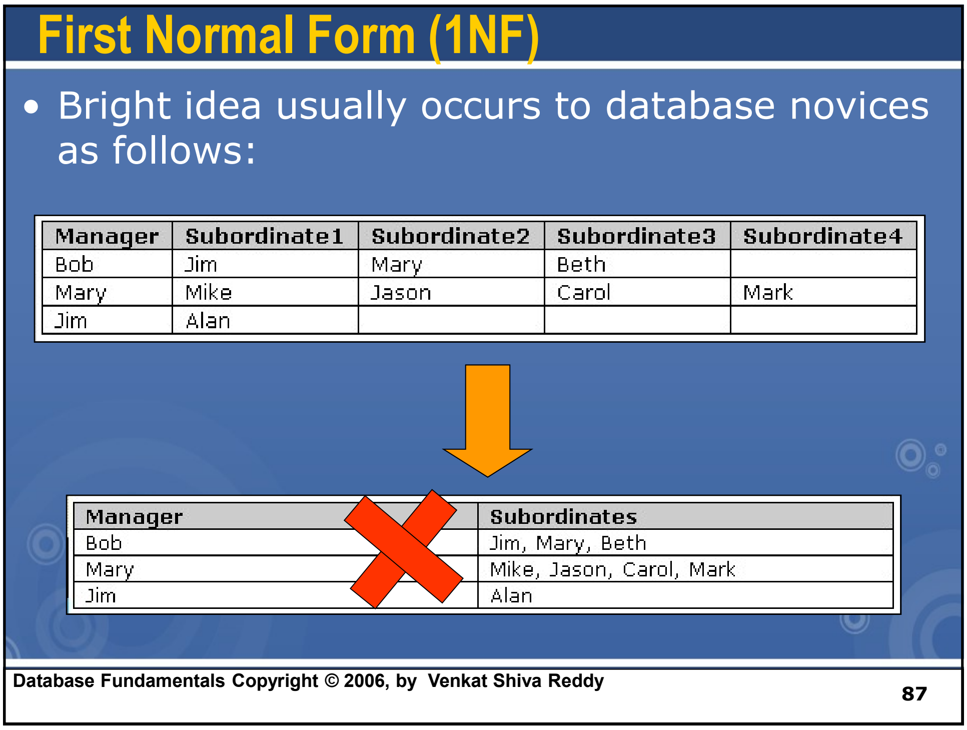


First Normal Form (1NF)

- Find the problems in the following design?

Manager	Subordinate1	Subordinate2	Subordinate3	Subordinate4
Bob	Jim	Mary	Beth	
Mary	Mike	Jason	Carol	Mark
Jim	Alan			

- ➢ Subordinate1-Subordinate4 columns are duplicative.
- ➢ Jim only has one subordinate – the Subordinate2-Subordinate4 columns are simply wasted storage space
- ➢ Mary already has 4 subordinates – what happens if she takes on another employee?
- ➢ What if Bob takes one thousand employees?

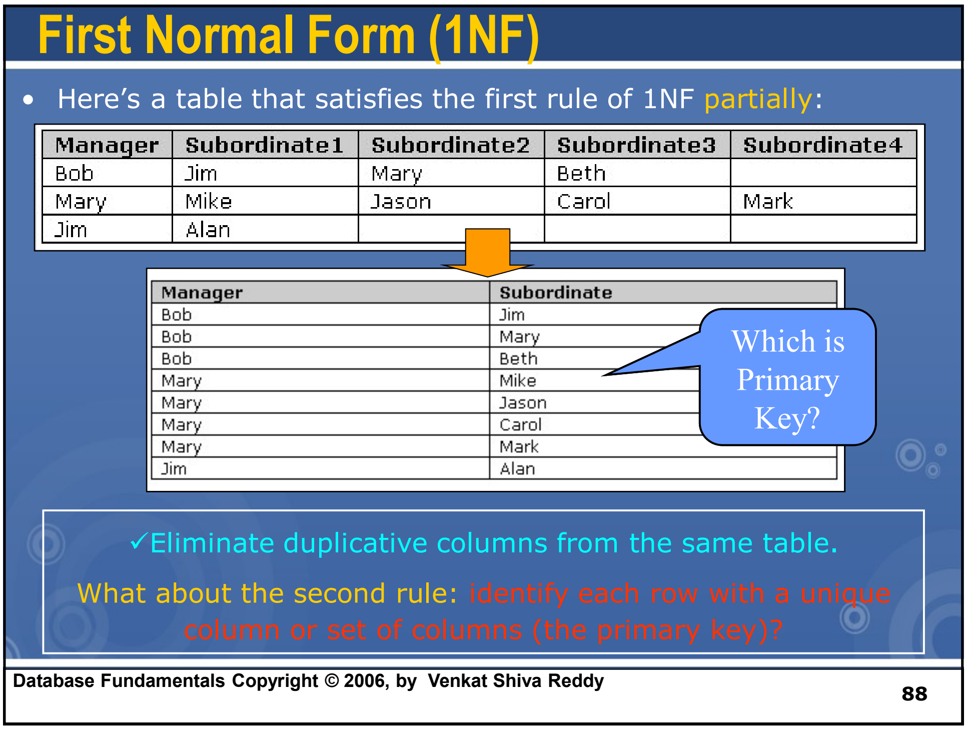


First Normal Form (1NF)

- Bright idea usually occurs to database novices as follows:

Manager	Subordinate1	Subordinate2	Subordinate3	Subordinate4
Bob	Jim	Mary	Beth	
Mary	Mike	Jason	Carol	Mark
Jim	Alan			

Manager	Subordinates
Bob	Jim, Mary, Beth
Mary	Mike, Jason, Carol, Mark
Jim	Alan



First Normal Form (1NF)

- Here's a table that satisfies the first rule of 1NF partially:

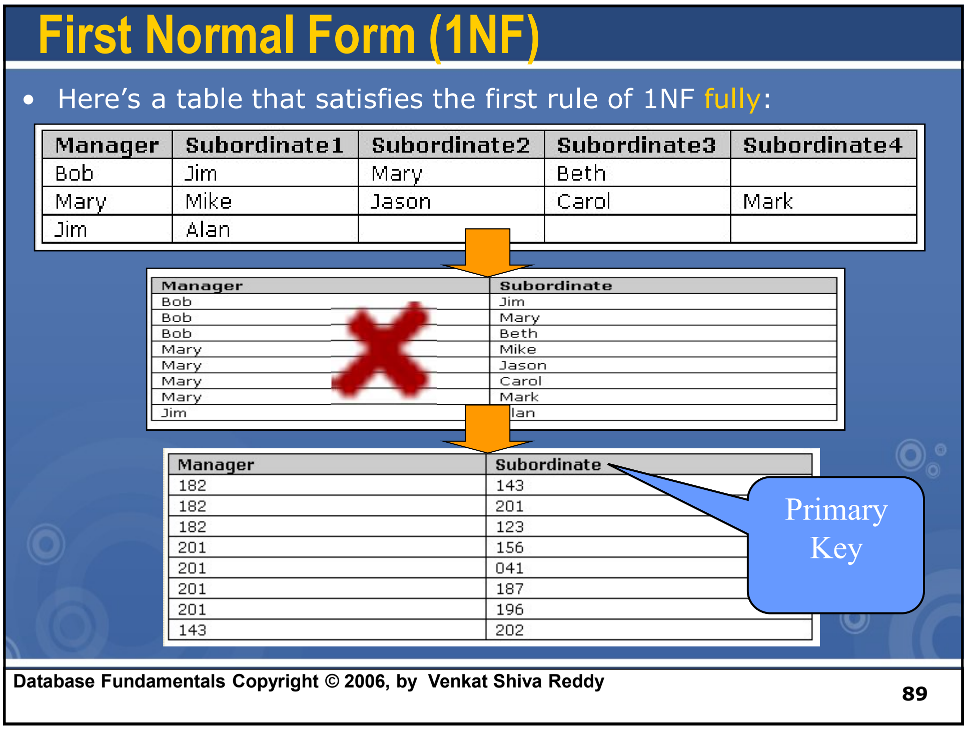
Manager	Subordinate1	Subordinate2	Subordinate3	Subordinate4
Bob	Jim	Mary	Beth	
Mary	Mike	Jason	Carol	Mark
Jim	Alan			

Manager	Subordinate
Bob	Jim
Bob	Mary
Bob	Beth
Mary	Mike
Mary	Jason
Mary	Carol
Mary	Mark
Jim	Alan

Which is Primary Key?

✓Eliminate duplicative columns from the same table.

What about the second rule: identify each row with a unique column or set of columns (the primary key)?



First Normal Form (1NF)

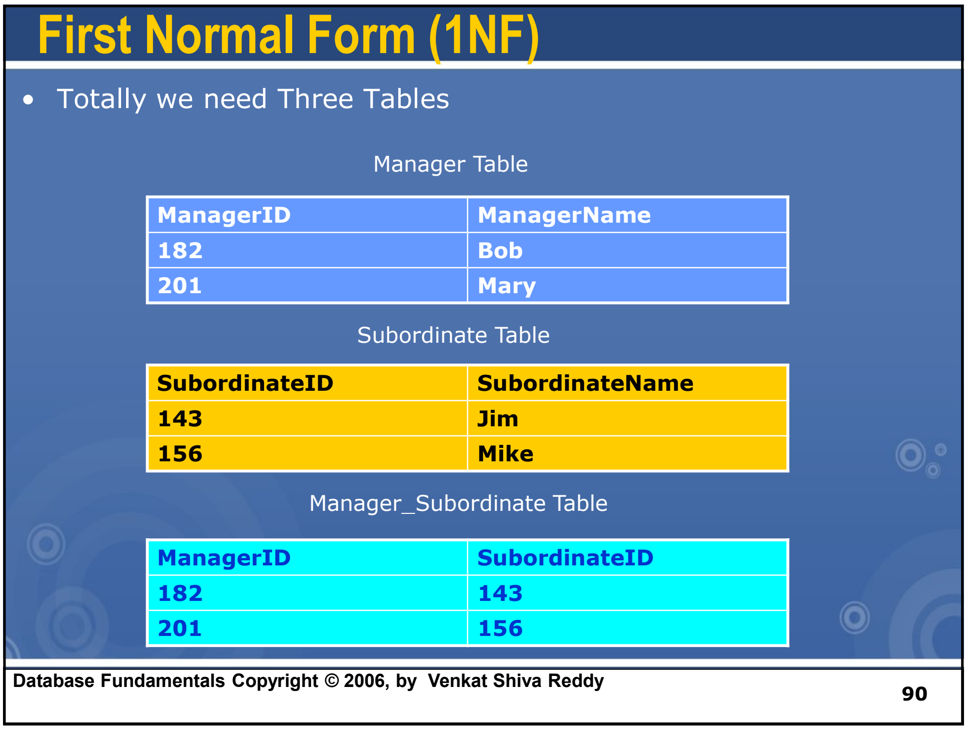
- Here’s a table that satisfies the first rule of 1NF fully:

Manager	Subordinate1	Subordinate2	Subordinate3	Subordinate4
Bob	Jim	Mary	Beth	
Mary	Mike	Jason	Carol	Mark
Jim	Alan			

Manager	Subordinate
Bob	Jim
Bob	Mary
Bob	Beth
Mary	Mike
Mary	Jason
Mary	Carol
Mary	Mark
Jim	lan

Manager	Subordinate
182	143
182	201
182	123
201	156
201	041
201	187
201	196
143	202

Primary Key



First Normal Form (1NF)

- Totally we need Three Tables

Manager Table

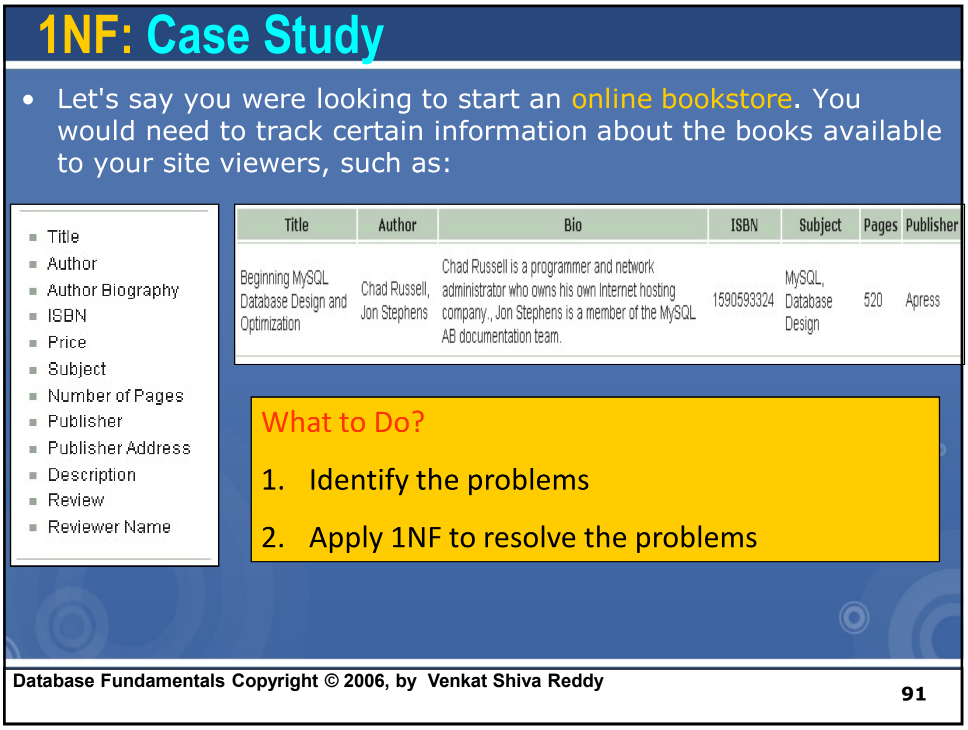
ManagerID	ManagerName
182	Bob
201	Mary

Subordinate Table

SubordinateID	SubordinateName
143	Jim
156	Mike

Manager_Subordinate Table

ManagerID	SubordinateID
182	143
201	156



1NF: Case Study

- Let's say you were looking to start an online bookstore. You would need to track certain information about the books available to your site viewers, such as:

- Title
- Author
- Author Biography
- ISBN
- Price
- Subject
- Number of Pages
- Publisher
- Publisher Address
- Description
- Review
- Reviewer Name

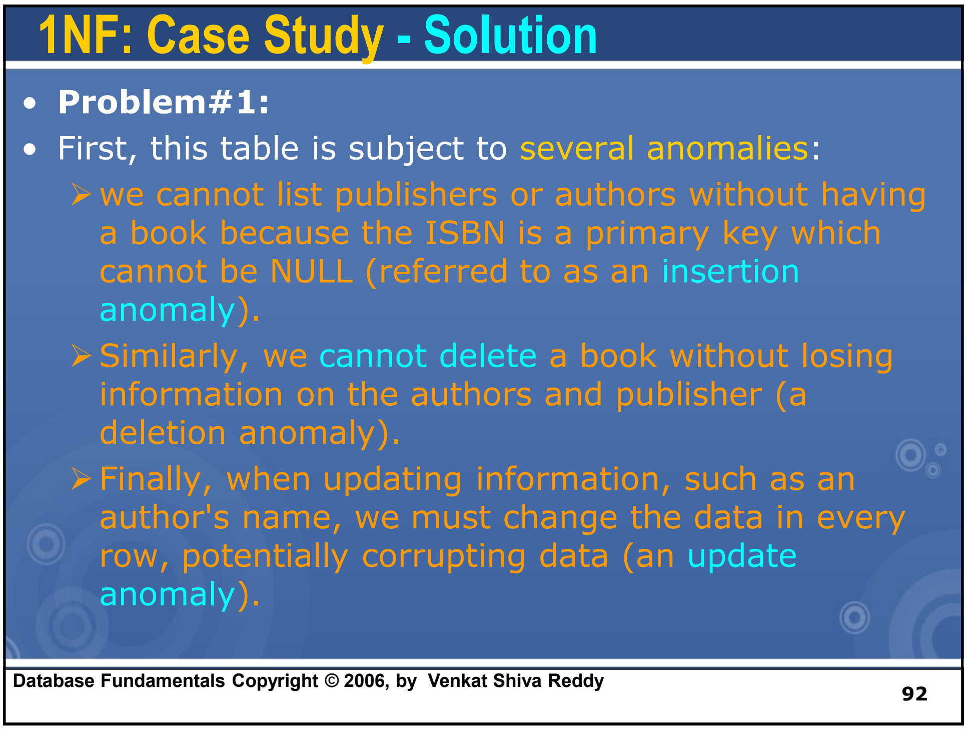
Title	Author	Bio	ISBN	Subject	Pages	Publisher
Beginning MySQL Database Design and Optimization	Chad Russell, Jon Stephens	Chad Russell is a programmer and network administrator who owns his own Internet hosting company., Jon Stephens is a member of the MySQL AB documentation team.	1590593324	MySQL, Database Design	520	Apress

What to Do?

1. Identify the problems
2. Apply 1NF to resolve the problems

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

91

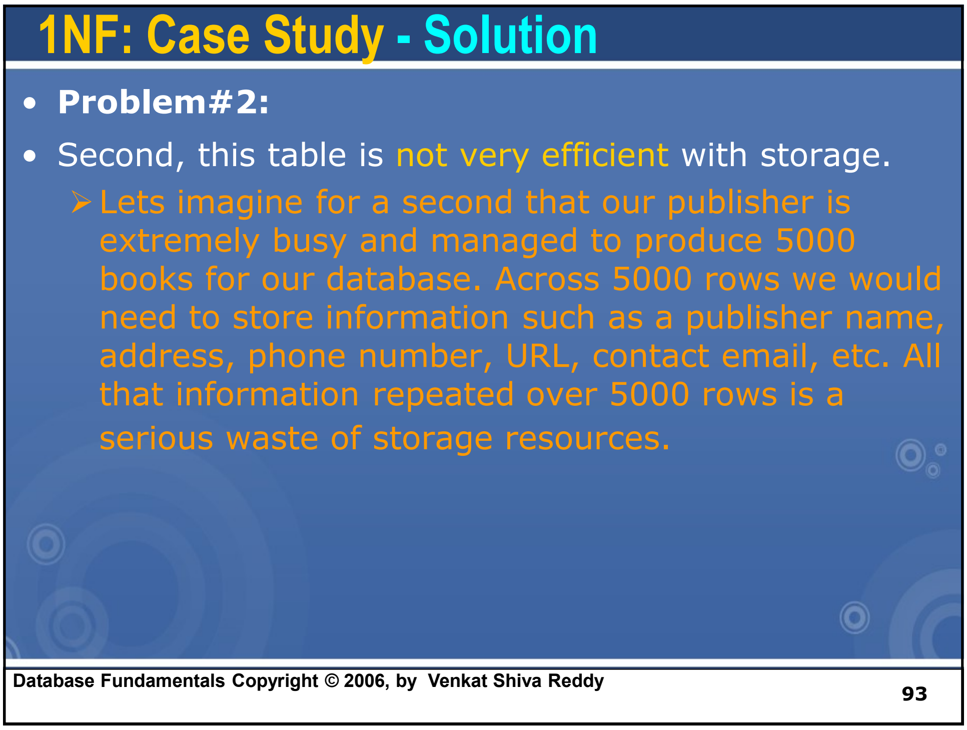


1NF: Case Study - Solution

- **Problem#1:**
- First, this table is subject to several anomalies:
  - we cannot list publishers or authors without having a book because the ISBN is a primary key which cannot be NULL (referred to as an insertion anomaly).
  - Similarly, we cannot delete a book without losing information on the authors and publisher (a deletion anomaly).
  - Finally, when updating information, such as an author's name, we must change the data in every row, potentially corrupting data (an update anomaly).

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

92

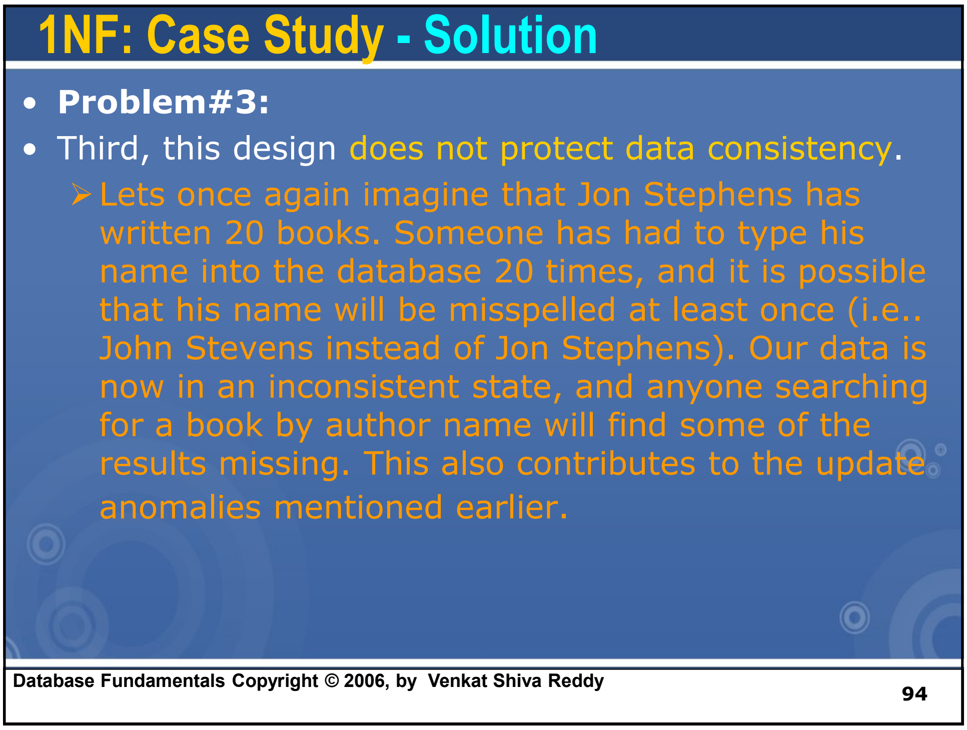


1NF: Case Study - Solution

- **Problem#2:**
- Second, this table is not very efficient with storage.
  - ➢ Lets imagine for a second that our publisher is extremely busy and managed to produce 5000 books for our database. Across 5000 rows we would need to store information such as a publisher name, address, phone number, URL, contact email, etc. All that information repeated over 5000 rows is a serious waste of storage resources.

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

93

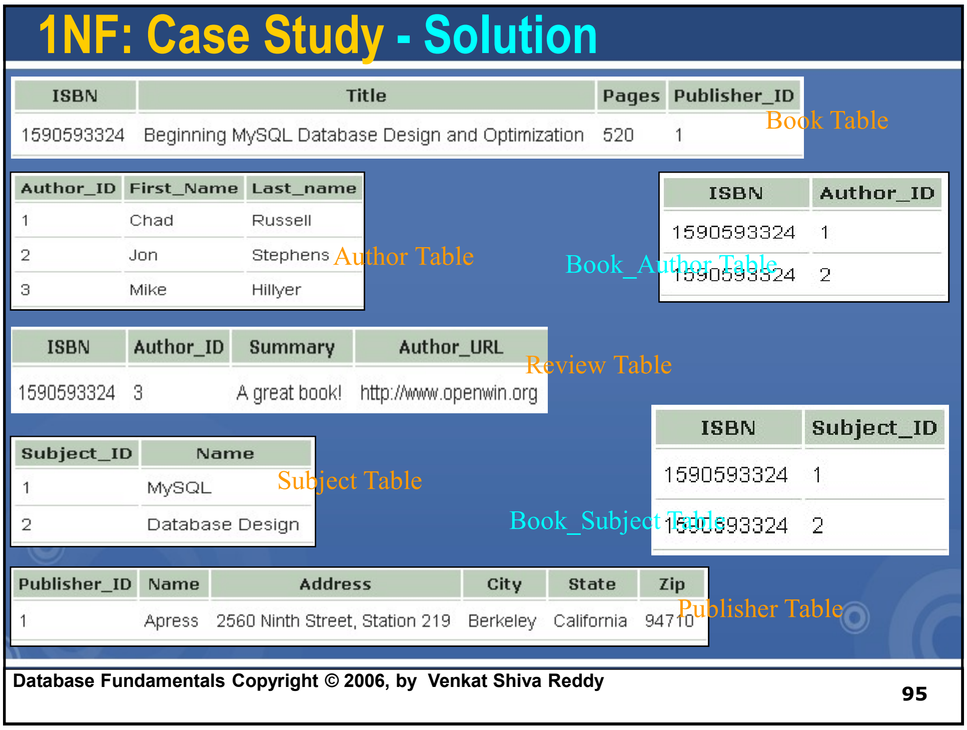


1NF: Case Study - Solution

- **Problem#3:**
- Third, this design does not protect data consistency.
  - ➢ Lets once again imagine that Jon Stephens has written 20 books. Someone has had to type his name into the database 20 times, and it is possible that his name will be misspelled at least once (i.e.. John Stevens instead of Jon Stephens). Our data is now in an inconsistent state, and anyone searching for a book by author name will find some of the results missing. This also contributes to the update anomalies mentioned earlier.

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

94



1NF: Case Study - Solution

Book Table

ISBN	Title	Pages	Publisher_ID
1590593324	Beginning MySQL Database Design and Optimization	520	1

Author Table

Author_ID	First_Name	Last_name
1	Chad	Russell
2	Jon	Stephens
3	Mike	Hillyer

Book_Author Table

ISBN	Author_ID
1590593324	1
1590593324	2

Review Table

ISBN	Author_ID	Summary	Author_URL
1590593324	3	A great book!	http://www.openwin.org

Subject Table

Subject_ID	Name
1	MySQL
2	Database Design

Book_Subject Table

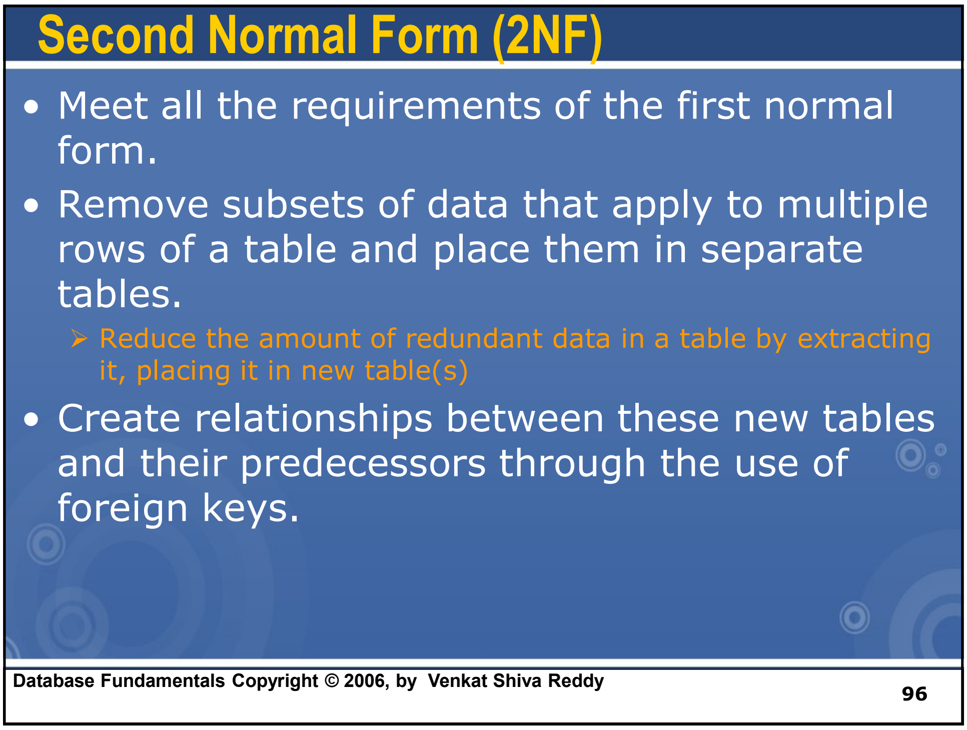
ISBN	Subject_ID
1590593324	1
1590593324	2

Publisher Table

Publisher_ID	Name	Address	City	State	Zip
1	Apress	2560 Ninth Street, Station 219	Berkeley	California	94710

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

95



Second Normal Form (2NF)

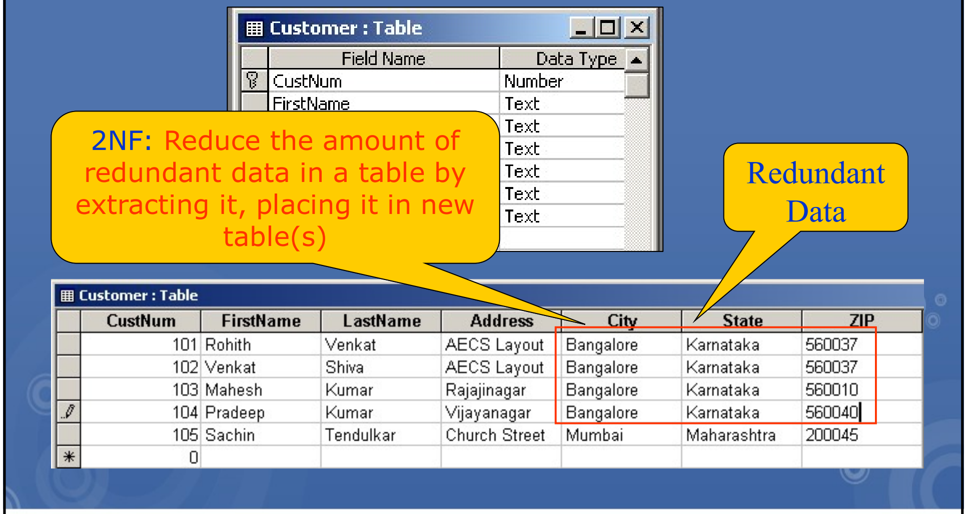
- Meet all the requirements of the first normal form.
- Remove subsets of data that apply to multiple rows of a table and place them in separate tables.
  - Reduce the amount of redundant data in a table by extracting it, placing it in new table(s)
- Create relationships between these new tables and their predecessors through the use of foreign keys.

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

96

Second Normal Form (2NF)

- Customer Database Table



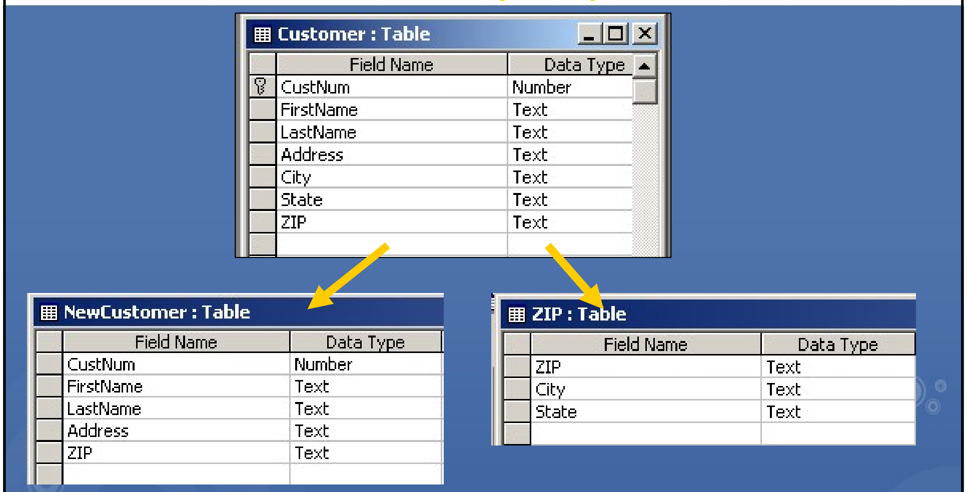
Field Name	Data Type
CustNum	Number
FirstName	Text
	Text
	Text
	Text
	Text
	Text

2NF: Reduce the amount of redundant data in a table by extracting it, placing it in new table(s)

Redundant Data

CustNum	FirstName	LastName	Address	City	State	ZIP
101	Rohith	Venkat	AECS Layout	Bangalore	Karnataka	560037
102	Venkat	Shiva	AECS Layout	Bangalore	Karnataka	560037
103	Mahesh	Kumar	Rajajinagar	Bangalore	Karnataka	560010
104	Pradeep	Kumar	Vijayanagar	Bangalore	Karnataka	560040
105	Sachin	Tendulkar	Church Street	Mumbai	Maharashtra	200045
0						

Second Normal Form (2NF)

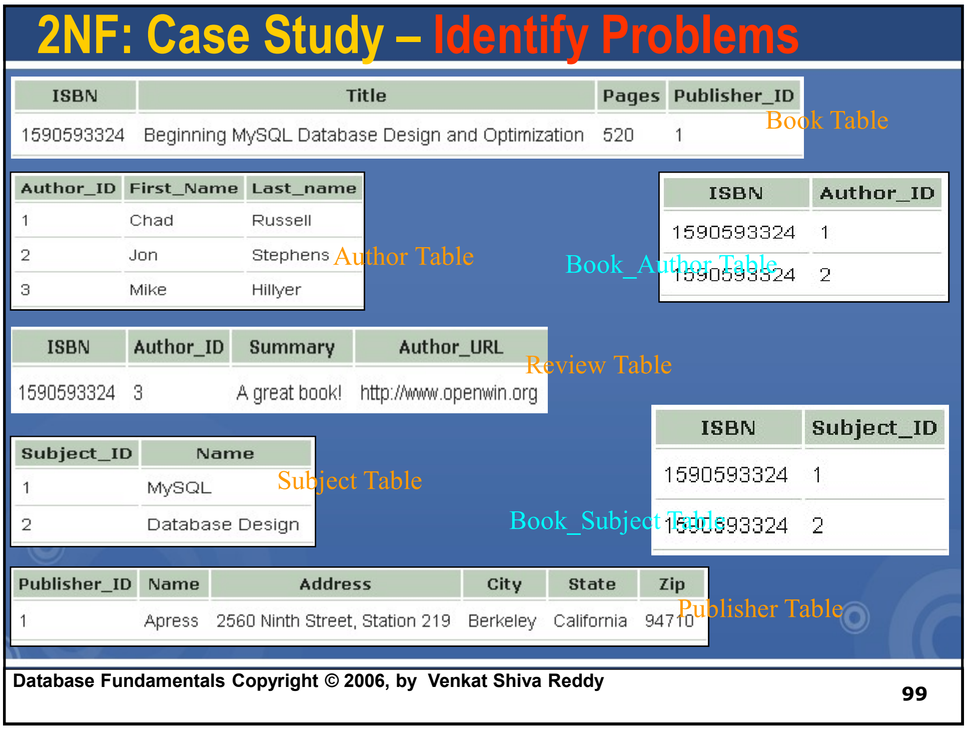


Field Name	Data Type
CustNum	Number
FirstName	Text
LastName	Text
Address	Text
City	Text
State	Text
ZIP	Text

Field Name	Data Type
CustNum	Number
FirstName	Text
LastName	Text
Address	Text
ZIP	Text

Field Name	Data Type
ZIP	Text
City	Text
State	Text

We've now minimized the amount of redundant information stored within the database and our structure is in second normal form!



2NF: Case Study – Identify Problems

Book Table

ISBN	Title	Pages	Publisher_ID
1590593324	Beginning MySQL Database Design and Optimization	520	1

Author Table

Author_ID	First_Name	Last_name
1	Chad	Russell
2	Jon	Stephens
3	Mike	Hillyer

Book_Author Table

ISBN	Author_ID
1590593324	1
1590593324	2

Review Table

ISBN	Author_ID	Summary	Author_URL
1590593324	3	A great book!	http://www.openwin.org

Subject Table

Subject_ID	Name
1	MySQL
2	Database Design

Book_Subject Table

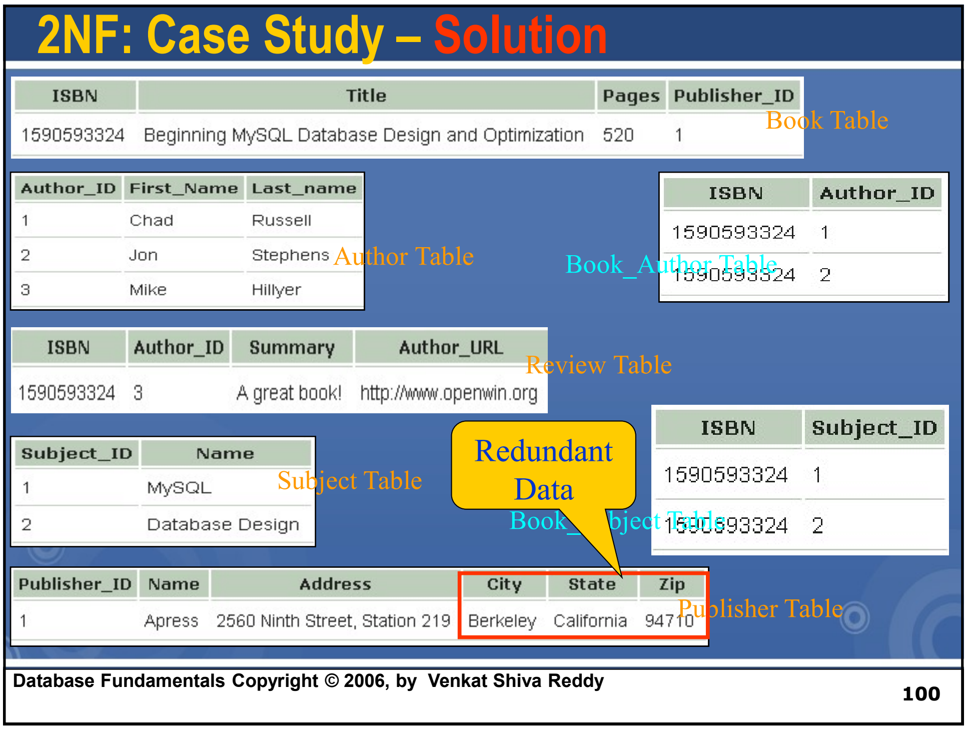
ISBN	Subject_ID
1590593324	1
1590593324	2

Publisher Table

Publisher_ID	Name	Address	City	State	Zip
1	Apress	2560 Ninth Street, Station 219	Berkeley	California	94710

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

99



2NF: Case Study – Solution

Book Table

ISBN	Title	Pages	Publisher_ID
1590593324	Beginning MySQL Database Design and Optimization	520	1

Author Table

Author_ID	First_Name	Last_name
1	Chad	Russell
2	Jon	Stephens
3	Mike	Hillyer

Book_Author Table

ISBN	Author_ID
1590593324	1
1590593324	2

Review Table

ISBN	Author_ID	Summary	Author_URL
1590593324	3	A great book!	http://www.openwin.org

Subject Table

Subject_ID	Name
1	MySQL
2	Database Design

Book_Subject Table

ISBN	Subject_ID
1590593324	1
1590593324	2

Redundant Data

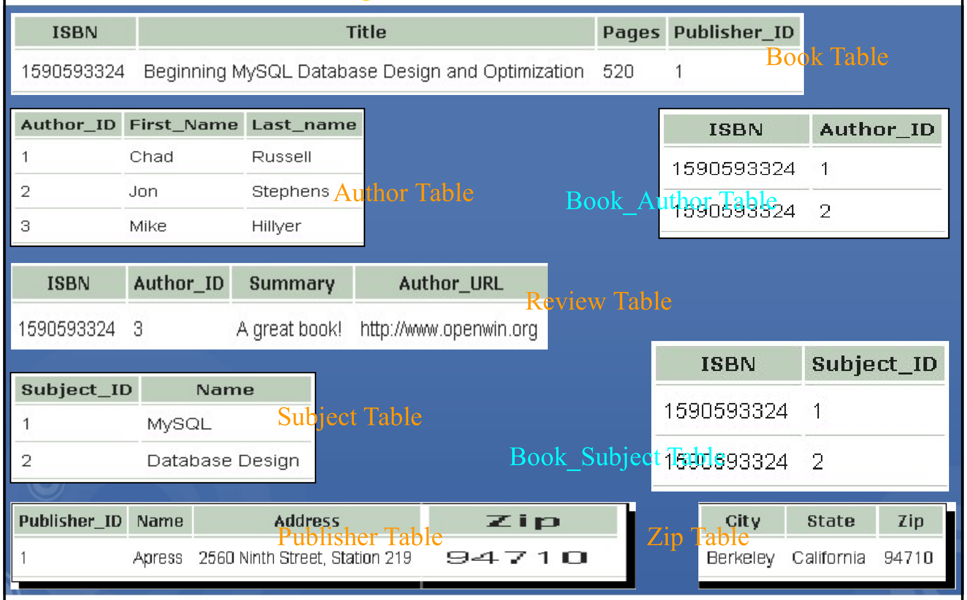
Publisher Table

Publisher_ID	Name	Address	City	State	Zip
1	Apress	2560 Ninth Street, Station 219	Berkeley	California	94710

Database Fundamentals Copyright © 2006, by Venkat Shiva Reddy

100

2NF: Case Study – Solution



Book Table

ISBN	Title	Pages	Publisher_ID
1590593324	Beginning MySQL Database Design and Optimization	520	1

Author Table

Author_ID	First_Name	Last_name
1	Chad	Russell
2	Jon	Stephens
3	Mike	Hillyer

Book_Author Table

ISBN	Author_ID
1590593324	1
1590593324	2

Review Table

ISBN	Author_ID	Summary	Author_URL
1590593324	3	A great book!	http://www.openwin.org

Subject Table

Subject_ID	Name
1	MySQL
2	Database Design

Book_Subject Table

ISBN	Subject_ID
1590593324	1
1590593324	2

Publisher Table

Publisher_ID	Name	Address	Zip
1	Apress	2560 Ninth Street, Station 219	94710

Zip Table

City	State	Zip
Berkeley	California	94710

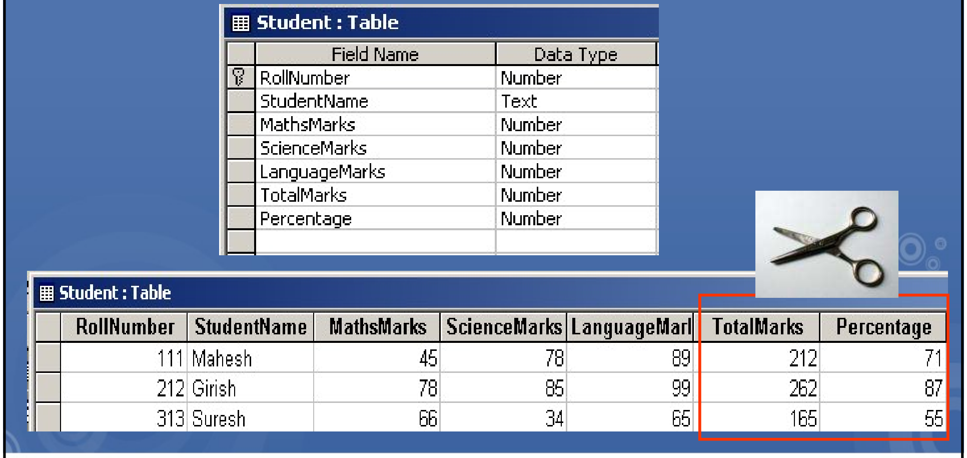
Third Normal Form (3NF)

- Meet all the requirements of the second normal form.
- Remove columns that are not dependent upon the primary key.

Third Normal Form (3NF)

- Remove columns that are not dependent upon the primary key

  If any columns would require updating
  if another column in the table was updated



Student : Table

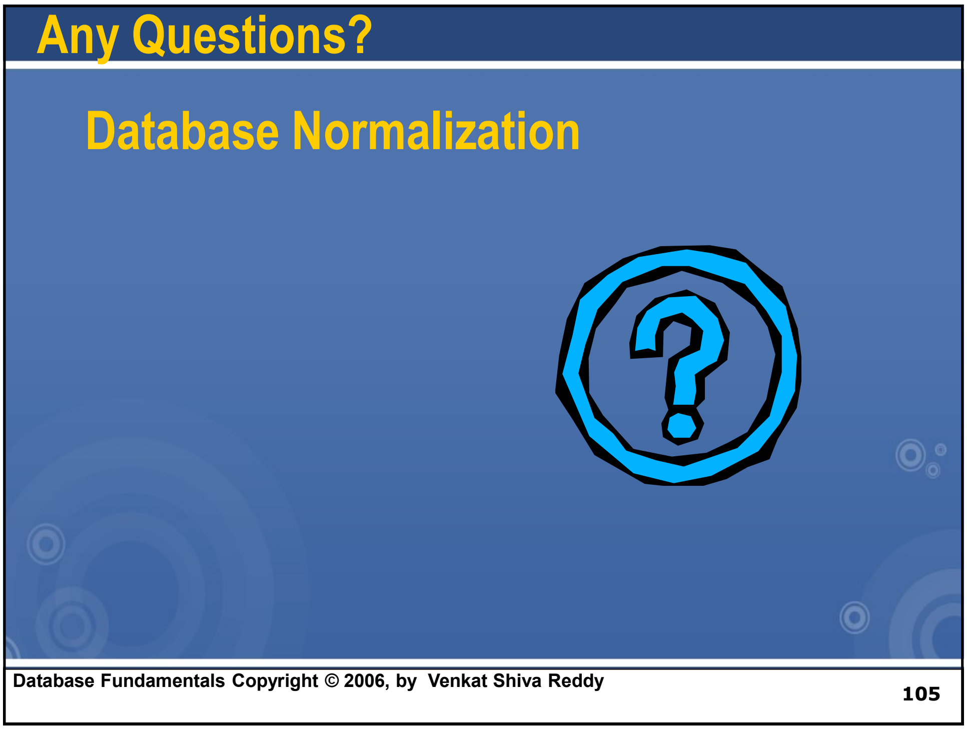
Field Name	Data Type
RollNumber	Number
StudentName	Text
MathsMarks	Number
ScienceMarks	Number
LanguageMarks	Number
TotalMarks	Number
Percentage	Number

Student : Table

RollNumber	StudentName	MathsMarks	ScienceMarks	LanguageMarl	TotalMarks	Percentage
111	Mahesh	45	78	89	212	71
212	Girish	78	85	99	262	87
313	Suresh	66	34	65	165	55

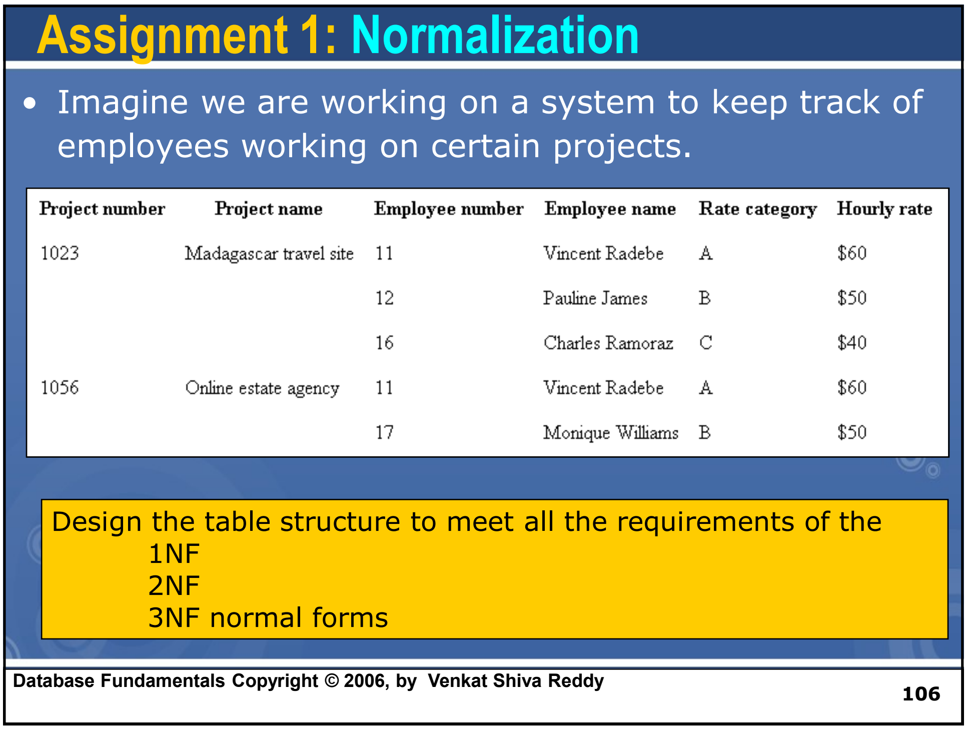
Other Normal Forms

- There are other normal forms –
  - Boyce-Codd normal form, and
  - 4th normal form,
- But these are very rarely used for business applications. In most cases, tables in 3rd normal form are already in these normal forms anyway.



Any Questions?

Database Normalization

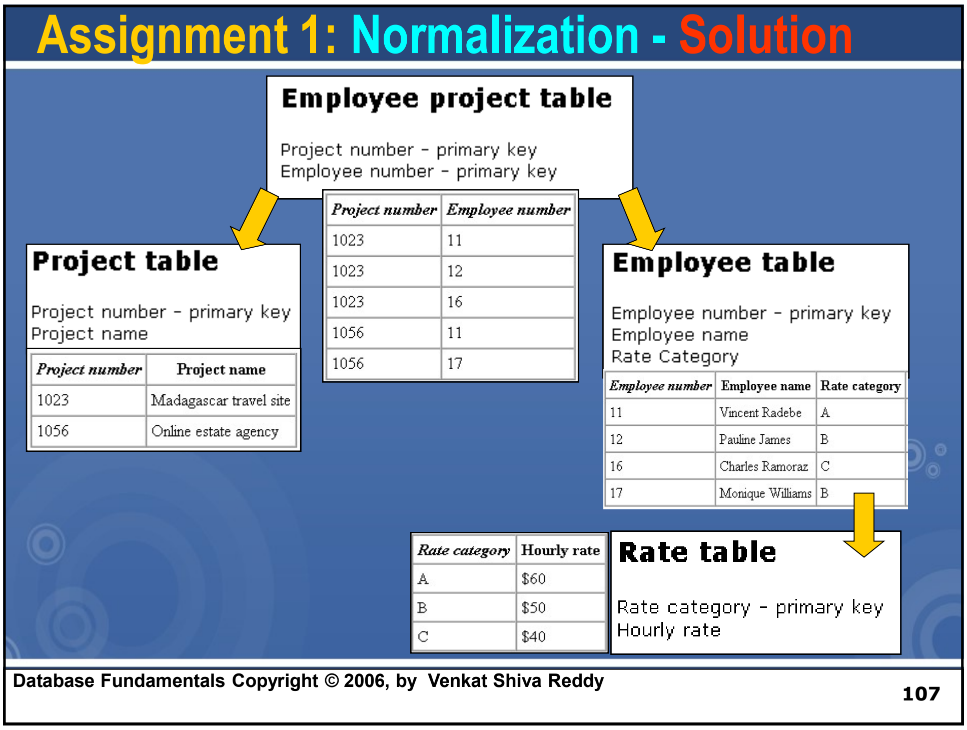


Assignment 1: Normalization

- Imagine we are working on a system to keep track of employees working on certain projects.

Project number	Project name	Employee number	Employee name	Rate category	Hourly rate
1023	Madagascar travel site	11	Vincent Radebe	A	$60
		12	Pauline James	B	$50
		16	Charles Ramoraz	C	$40
1056	Online estate agency	11	Vincent Radebe	A	$60
		17	Monique Williams	B	$50

Design the table structure to meet all the requirements of the
1NF
2NF
3NF normal forms



Assignment 1: Normalization - Solution

Employee project table

Project number - primary key
Employee number - primary key

Project number	*Employee number*
1023	11
1023	12
1023	16
1056	11
1056	17

Project table

Project number - primary key
Project name

Project number	Project name
1023	Madagascar travel site
1056	Online estate agency

Employee table

Employee number - primary key
Employee name
Rate Category

Employee number	Employee name	Rate category
11	Vincent Radebe	A
12	Pauline James	B
16	Charles Ramoraz	C
17	Monique Williams	B

Rate table

Rate category - primary key
Hourly rate

Rate category	Hourly rate
A	$60
B	$50
C	$40