DESIGNING DATABASES

System Analysis and Design

1

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

LEARNING OBJECTIVES

- ✓Describe the database design process, its outcomes, and the relational database model.
- ✓Describe normalization and the rules for second and third normal form.
- ✓Transform an entity-relationship (E-R) diagram into an equivalent set of well-structured (normalized) relations.
- ✓Merge normalized relations from separate user views into a consolidated set of well-structured relations.

2

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

LEARNING OBJECTIVES (CONT.)

- ✓Describe physical database design concepts:
  - ✓Choose storage formats for fields in database tables.
  - ✓Translate well-structured relations into efficient database tables.
  - ✓Explain when to use different types of file organizations to store computer files.
  - ✓Describe the purpose of indexes and the important considerations in selecting attributes to be indexed.

3

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

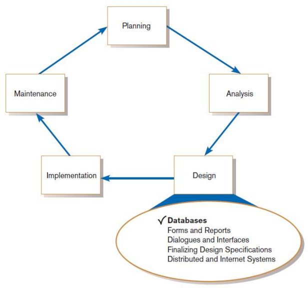
Merging Relations

Physical File and Database Design

Electronic Commerce Application

INTRODUCTION

Systems development life cycle with design phase highlighted



4

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

DATABASE DESIGN

File and database design occurs in two steps.

1. Develop a logical database model, which describes data using notation that corresponds to a data organization used by a database management system.
   - Relational database model
2. Prescribe the technical specifications for computer files and databases in which to store the data.
   - Physical database design provides specifications

Logical and physical database design in parallel with other system design steps

5

Introduction

Database Design

Normalization

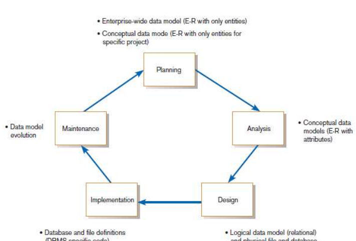
Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

THE PROCESS OF DATABASE DESIGN



Relationship between data modeling and the systems development life cycle

6

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

THE PROCESS OF DATABASE DESIGN (CONT.)

Four key steps in logical database modeling and design:

1. Develop a logical data model for each known user interface for the application using normalization principles.
2. Combine normalized data requirements from all user interfaces into one consolidated logical database model (view integration).
3. Translate the conceptual E-R data model for the application into normalized data requirements.
4. Compare the consolidated logical database design with the translated E-R model and produce one final logical database model for the application.

7

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

PHYSICAL DATABASE DESIGN

Key physical database design decisions include:

- Choosing a storage format for each attribute from the logical database model.
- Grouping attributes from the logical database model into physical records.
- Arranging related records in secondary memory (hard disks and magnetic tapes) so that records can be stored, retrieved and updated rapidly.
- Selecting media and structures for storing data to make access more efficient.

8

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

DELIVERABLES AND OUTCOMES

Logical database design

- Must account for every data element on a system input or output
  - Normalized relations are the primary deliverable.

Physical database design

- Converts relations into database tables
  - Programmers and database analysts code the definitions of the database.
  - Written in Structured Query Language (SQL)

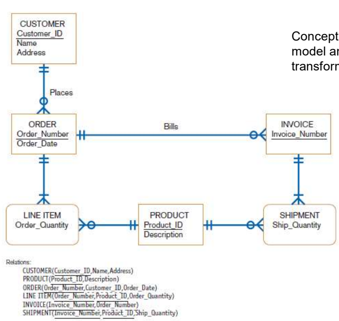
9

RELATIONAL DATABASE MODEL

Relational database model: data represented as a set of related tables or relations

Relation: a named, two-dimensional table of data; each relation consists of a set of named columns and an arbitrary number of unnamed rows

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application



Conceptual data model and transformed relations

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

RELATIONAL DATABASE MODEL (CONT.)

Relations have several properties that distinguish them from nonrelational tables:

- Entries in cells are simple.
- Entries in columns are from the same set of values.
- Each row is unique.
- The sequence of columns can be interchanged without changing the meaning or use of the relation.
- The rows may be interchanged or stored in any sequence.

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

WELL-STRUCTURED RELATION AND PRIMARY KEYS

Well-Structured Relation (or **table)**
- A relation that contains a minimum amount of redundancy
- Allows users to insert, modify, and delete the rows without errors or inconsistencies

Primary Key
- An attribute whose value is unique across all occurrences of a relation

All relations have a primary key.
- This is how rows are ensured to be unique.
- A primary key may involve a single attribute or be composed of multiple attributes.

13

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

NORMALIZATION AND RULES OF NORMALIZATION

Normalization: the process of converting complex data structures into simple, stable data structures

The result of normalization is that every nonprimary key attribute depends upon the whole primary key.

14

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

NORMALIZATION AND RULES OF NORMALIZATION (CONT.)

First Normal Form (1NF)
- Unique rows, no multivalued attributes
- All relations are in 1NF

Second Normal Form (2NF)
- Each nonprimary key attribute is identified by the whole primary key (called full functional dependency)

Third Normal Form (3NF)
- Nonprimary key attributes do not depend on each other (i.e. no transitive dependencies)

15

15

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

FUNCTIONAL DEPENDENCIES AND PRIMARY KEYS

Functional Dependency: a particular relationship between two attributes

- For a given relation, attribute B is functionally dependent on attribute A if, for every valid value of A, that value of A uniquely determines the value of B.
- The functional dependence of B on A is represented by A→B.

16

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

FUNCTIONAL DEPENDENCIES AND PRIMARY KEYS (CONT.)

Functional dependency is not a mathematical dependency.

Instances (or sample data) in a relation do not prove the existence of a functional dependency.

Knowledge of problem domain is most reliable method for identifying functional dependency.

17

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

SECOND NORMAL FORM (2NF)

A relation is in second normal form (2NF) if any of the following conditions apply:

- The primary key consists of only one attribute.
- No nonprimary key attributes exist in the relation.
- Every nonprimary key attribute is functionally dependent on the full set of primary key attributes.

To convert a relation into 2NF, decompose the relation into new relations using the attributes, called *determinants*, that determine other attributes.

The determinants are the primary keys of the new relations.

18

THIRD NORMAL FORM (3NF)

A relation is in third normal form (3NF) if it is in second normal form (2NF) and there are no functional (transitive) dependencies between two (or more) nonprimary key attributes.

19

EMPLOYEE2

Emp_ID	Name	Dept	Salary	Course	Date_Completed
100	Margaret Simpson	Marketing	42,000	SPSS	6/19/2017
100	Margaret Simpson	Marketing	42,000	Surveys	10/7/2017
140	Alan Beeton	Accounting	39,000	Tax Acc	12/8/2017
110	Chris Lucero	Info Systems	41,500	SPSS	1/22/2017
110	Chris Lucero	Info Systems	41,500	C++	4/22/2017
190	Lorenzo Davis	Finance	38,000	Investments	5/7/2017
150	Susan Martin	Marketing	38,500	SPSS	6/19/2017
150	Susan Martin	Marketing	38,500	TQM	8/12/2017

FIGURE 9-6
Relation with redundancy

Emp_ID → Name,Dept,Salary (partial dependency)
Emp_ID,Course → Date_Completed (complete depencency)

EMP COURSE

Emp_ID	Course	Date_ Completed
100	SPSS	6/19/2017
100	Surveys	10/7/2017
140	Tax Acc	12/8/2017
110	SPSS	1/22/2017
110	C++	4/22/2017
190	Investments	5/7/2017
150	SPSS	6/19/2017
150	TQM	8/12/2017

FIGURE 9-7
EMP COURSE relation

20

FIGURE 9-9
Removing transitive dependencies
(a) Relation with transitive dependency

SALES

Customer_ID	Customer_Name	Salesperson	Region
8023	Anderson	Smith	South
9167	Bancroft	Hicks	West
7924	Hobbs	Smith	South
6837	Tucker	Hernandez	East
8596	Eckersley	Hicks	West
7018	Arnold	Faulb	North

Customer_ID → Customer_Name,Salesperson,Region

Salesperson → Region

(b) Relation in 3NF

SALES1

Customer_ID	Customer_Name	Salesperson
8023	Anderson	Smith
9167	Bancroft	Hicks
7924	Hobbs	Smith
6837	Tucker	Hernandez
8596	Eckersley	Hicks
7018	Arnold	Faulb

SPERSON

Salesperson	Region
Smith	South
Hicks	West
Hernandez	East
Faulb	North

21

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

THIRD NORMAL FORM (3NF) (CONT.)

Foreign Key: an attribute that appears as a nonprimary key attribute (or part of a primary key) in one relation and as a primary key attribute in another relation

Referential Integrity: an integrity constraint specifying that the value (or existence) of an attribute in one relation depends on the value (or existence) of the same attribute in another relation

22

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

1NF TO 3NF

First Normal Form (1NF)

- Eliminate Repeating Groups
  - **List of values**
  - **Enumerated Fields**
- Each field should be a single datatype

Second Normal Form (2NF)

- Remove **functional dependencies**
  - Groups of columns that depend on each other, rather than the key

Third Normal Form (3NF)

- Remove **transient dependencies**
  - Where a field depends more on another column, rather than the primary key

23

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

Name	Phone1	Phone2	Address	City	State
Mortimer, Dan	856-555-0000	856-555-0001	1 Sassafras	Anytown	NJ
Washington, Reggie	856-555-0010	856-555-0020	2 Oak	Somewhere	NJ
Young, Luke	215-555-1000	215-555-2000	3 Pine	Elsewhere	PA

Single Data Types

Repeating

Last	First	[illegible]	Phone2	Address	City	State
Mortimer	Dan	856-555-0000	856-555-0001	1 Sassafras	Anytown	NJ
Washington	Reggie	856-555-0010	856-555-0020	2 Oak	Somewhere	NJ
Young	Luke	215-555-1000	215-555-2000	3 Pine	Elsewhere	PA

Repeating

Last	First	Phone	Address	City	State
Mortimer	Dan	856-555-0000	1 Sassafras	Anytown	NJ
Mortimer	Dan	856-555-0001	1 Sassafras	Anytown	NJ
Washington	Reggie	856-555-0020	2 Oak	Somewhere	NJ
Washington	Reggie	856-555-0010	2 Oak	Somewhere	NJ
Young	Luke	215-555-1000	3 Pine	Elsewhere	PA
Young	Luke	215-555-2000	3 Pine	Elsewhere	PA

1NF

24

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

Repeating

ID	Last	First	Address	City	State
1	Mortimer	Dan	1 Sassafras	Anytown	NJ
2	Washington	Reggie	2 Oak	Somewhere	NJ
3	Young	Luke	3 Pine	Elsewhere	PA

2NF

ID	FK_ID	Phone
10	1	856-555-0000
11	1	856-555-0001
12	2	856-555-0020
13	2	856-555-0010
14	3	215-555-1000
15	3	215-555-2000

25

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

ID	Last	First	FK_ID
1	Mortimer	Dan	100
2	Washington	Reggie	101
3	Young	Luke	102

3NF

ID	Address	City	State
100	1 Sassafras	Anytown	NJ
101	2 Oak	Somewhere	NJ
102	3 Pine	Elsewhere	PA

26

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

ID	Last	First	FK_ID
1	Mortimer	Dan	100
2	Washington	Reggie	101
3	Young	Luke	102

ID	Address	City_ID	FK_ID
100	1 Sassafras	30	2000
101	2 Oak	31	2000
102	3 Pine	33	2002

ID	City
30	Anytown
31	Somewhere
33	Elsewhere

ID	State
2000	NJ
2002	PA

27

DENORMALIZATION

Sometimes it is necessary to denormalize a table for performance reasons.

Denormalization is where you recombine tables that were split apart to conform to the rules of the various normal forms.

Denormalization should never be done lightly, because it opens up your database to the anomalies and errors normalization was designed to eliminate.

28

ID	Last	First	FK_ID
1	Mortimer	Dan	100
2	Washington	Reggie	101
3	Young	Luke	102

1..1 — 0..M

ID	FK_ID	Phone
10	1	856-555-0000
11	1	856-555-0001
12	2	856-555-0020
13	2	856-555-0010
14	3	215-555-1000
15	3	215-555-2000

1..M — 1..1

ID	Address	City	State
100	1 Sassafras	Anytown	NJ
101	2 Oak	Somewhere	NJ
102	3 Pine	Elsewhere	PA

29

TRANSFORMING E-R DIAGRAMS INTO RELATIONS

It is useful to transform the conceptual data model into a set of normalized relations.

Steps

- *Represent entities.*
- *Represent relationships.*
- *Normalize the relations.*
- *Merge the relations.*

30

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

REPRESENTING ENTITIES

Each regular entity is transformed into a relation.

The identifier of the entity type becomes the primary key of the corresponding relation.

31

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

REPRESENTING ENTITIES

The primary key must satisfy the following two conditions.

- The value of the key must uniquely identify every row in the relation.
- The key should be nonredundant.

The entity type label is translated into a relation name.

32

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

BINARY 1:N AND 1:1RELATIONSHIPS

The procedure for representing relationships depends on both the degree of the relationship—unary, binary, ternary—and the cardinalities of the relationship.

Binary 1:N Relationship is represented by adding the primary key attribute (or attributes) of the entity on the one side of the relationship as a foreign key in the relation that is on the many side of the relationship.

33

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

BINARY 1:N AND 1:1RELATIONSHIPS (CONT.)

Binary or Unary 1:1 Relationship is represented by any of the following choices:

- Add the primary key of A as a foreign key of B.
- Add the primary key of B as a foreign key of A.
- Both of the above

34

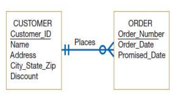


FIGURE 9-11
Representing a 1:N relationship
(a) E-R diagram

CUSTOMER

Customer_ID	Name	Address	City_State_ZIP	Discount
1273	Contemporary Designs	123 Oak St.	Austin, TX 28384	5%
6390	Casual Corner	18 Hoosier Dr.	Bloomington, IN 45821	3%

ORDER

Order_Number	Order_Date	Promised_Date	Customer_ID
57194	3/15/1X	3/28/1X	6390
63725	3/17/1X	4/01/1X	1273
80149	3/14/1X	3/24/1X	6390

(b) Relations

35

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

BINARY AND HIGHER-DEGREE M:N RELATIONSHIPS

Create another relation and include primary keys of all relations as primary key of new relation

36

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

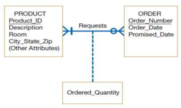
Merging Relations

Physical File and Database Design

Electronic Commerce Application

FIGURE 9-12
Representing an *M:N* relationship
(a) E-R diagram

(a)



(b) Relations

(b) ORDER

Order_Number	Order_Date	Promised_Date
61384	2/17/2014	3/01/2017
62009	2/13/2014	2/27/2017
62807	2/15/2014	3/01/2017

ORDER LINE

Order_Number	Product_ID	Quantity_ Ordered
61384	M128	2
61384	A261	1

PRODUCT

Product_ID	Description	Room	(Other Attributes)
M128	Bookcase	Study	—
A261	Wall unit	Family	—
R149	Cabinet	Study	—

37

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

UNARY RELATIONSHIPS

Unary 1:N Relationship

- Is modeled as a relation
- Primary key of that relation is the same as for the entity type
- Foreign key is added to the relation that references the primary key values

Recursive foreign key: a foreign key in a relation that references the primary key values of that same relation

38

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

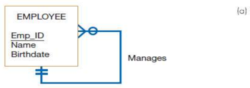
Electronic Commerce Application

UNARY RELATIONSHIPS

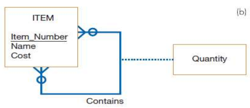
Unary M:N Relationship

- Model as one relation, then
- Create a separate relation to represent the M:N relationship.
- The primary key of the new relation is a composite key of two attributes that both take their values from the same primary key.
- Any attribute associated with the relationship is included as a nonkey attribute in this new relation.

39



EMPLOYEE(Emp_ID,Name,Birthdate,Manager_ID)



ITEM(Item_Number,Name,Cost)
ITEM-BILL(Item_Number,Component_Number,Quantity)

Two unary relationships

(a) EMPLOYEE with Manages relationship (1:N)

(b) Bill-of-materials structure (M:N)

40

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

MERGING RELATIONS

Purpose is to remove redundant relations

The last step in logical database design

Redundant relations could come about due to multiple E-R diagrams and/or user interfaces

Prior to physical file and database design

Example: given two relations:
- EMPLOYEE1(Emp_ID,Name,Address,Phone)
- EMPLOYEE2(Emp_ID,Name,Address,Jobcode,Number_of_Years)

You can merge them together:
- EMPLOYEE(Emp_ID,Name,Address,Phone,Jobcode,Number_of_Years)

41

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

VIEW INTEGRATION PROBLEMS

Must understand the meaning of the data and be prepared to resolve any problems that arise in the process

Synonyms: two different names used for the same attribute
- When merging, get agreement from users on a single, standard name.

Example of two relations with synonym primary keys of different names:
- STUDENT1(Student_ID,Name)
- STUDENT2(Matriculation_Number,Name,Address)

42

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

VIEW INTEGRATION PROBLEMS (CONT.)

Homonyms: a single attribute name that is used for two or more different attributes.

- Resolved by creating a new name
- Example: home address vs. local address?
  - STUDENT1 (Student_ID,Name,Address)
  - STUDENT2(Student_ID,Name,Phone_Number,Address)

43

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

VIEW INTEGRATION PROBLEMS (CONT.)

Dependencies between nonkeys— dependencies may be created as a result of view integration

Example: suppose we have these two relations:
- STUDENT1(Student_ID,Major)
- STUDENT2(Student_ID,Adviser)

You'd merge into this:
- STUDENT(Student_ID,Major,Adviser)

But if we have a transitive dependency like this:
- Major → Advisor

You need to **normalize** to remove the transitive dependency
- STUDENT(Student_ID,Major)
- MAJOR ADVISER(Major,Adviser)

44

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

VIEW INTEGRATION PROBLEMS (CONT.)

Class/Subclass — relationships may be hidden in user views or relations

Example: two relations
- PATIENT1(Patient_ID,Name,Address,Date_Treated)
- PATIENT2(Patient_ID,Room_Number)

In-patient vs. Out patient? Implies supertype/subtype
- PATIENT(Patient_ID,Name,Address)
- INPATIENT(Patient_ID,Room_Number)
- OUTPATIENT(Patient_ID,Date_Treated)

45

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

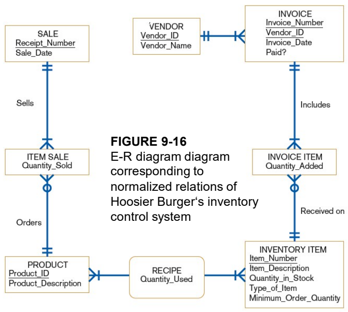


FIGURE 9-16
E-R diagram diagram corresponding to normalized relations of Hoosier Burger's inventory control system

46

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

PHYSICAL FILE AND DATABASE DESIGN

The following information is required:

- Normalized relations, including volume estimates
- Definitions of each attribute
- Descriptions of where and when data are used, entered, retrieved, deleted, and updated (including frequencies)
- Expectations or requirements for response time and data integrity
- Descriptions of the technologies used for implementing the files and database

47

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

DESIGNING FIELDS

Field: the smallest unit of named application data recognized by system software

- Attributes from relations will be represented as fields

Data Type: a coding scheme recognized by system software for representing organizational data

48

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

CHOOSING DATA TYPES

Selecting a data type balances four objectives:

- Minimize storage space.
- Represent all possible values of the field.
- Improve data integrity of the field.
- Support all data manipulations desired on the field.

49

Data Type	Description
VARCHAR2	Variable-length character data with a maximum length of 4000 characters; you must enter a maximum field length (e.g., VARCHAR2(30) for a field with a maximum length of 30 characters). A value less than 30 characters will consume only the required space.
CHAR	Fixed-length character data with a maximum length of 255 characters; default length is 1 character (e.g., CHAR(5) for a field with a fixed length of five characters, capable of holding a value from 0 to 5 characters long).
LONG	Capable of storing up to two gigabytes of one variable-length character data field (e.g., to hold a medical instruction or a customer comment).
NUMBER	Positive and negative numbers in the range 10^{-130} to 10^{126}; can specify the precision (total number of digits to the left and right of the decimal point) and the scale (the number of digits to the right of the decimal point) (e.g., NUMBER(5) specifies an integer field with a maximum of 5 digits and NUMBER(5, 2) specifies a field with no more than five digits and exactly two digits to the right of the decimal point).
DATE	Any date from January 1, 4712 BC to December 31, 4712 AD; date stores the century, year, month, day, hour, minute, and second.
BLOB	Binary large object, capable of storing up to four gigabytes of binary data (e.g., a photograph or sound clip).

50

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

CALCULATED FIELDS

Calculated (or **computed** or **derived**) **field**: a field that can be derived from other database fields

It is common for an attribute to be mathematically related to other data.

The calculate value is either stored or computed when it is requested.

51

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

CONTROLLING DATA INTEGRITY

Default Value: a value a field will assume unless an explicit value is entered for that field

Range Control: limits range of values that can be entered into field

- Both numeric and alphanumeric data

Referential Integrity: an integrity constraint specifying that the value (or existence) of an attribute in one relation depends on the value (or existence) of the same attribute in another relation

Null Value: a special field value, distinct from zero, blank, or any other value, that indicates that the value for the field is missing or otherwise unknown

52

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

DESIGNING PHYSICAL TABLES

Relational database is a set of related tables.

Physical Table: a named set of rows and columns that specifies the fields in each row of the table

Denormalization: the process of splitting or combining normalized relations into physical tables based on affinity of use of rows and fields

Denormalization optimizes certain data processing activities at the expense of others.

53

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

DESIGNING PHYSICAL TABLES (CONT.)

Various forms of denormalization, which involves combining data from several normalized tables, can be done.

- No hard-and-fast rules for deciding

Three common situations where denormalization may be used:

- *Two entities with a one-to-one relationship*
- *A many-to-many relationship (associative entity) with nonkey attributes*
- *Reference data*

54

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

DESIGNING PHYSICAL TABLES (CONT.)

Partitioning: splitting a table into different physical files, perhaps stored on different disks or computer.

Helps speed up system performance.

Three types of table partitioning:

- *Range partitioning*: partitions are defined by nonoverlapping ranges of values for a specified attribute
- *Hash partitioning*: a table row is assigned to a partition by an algorithm and then maps the specified attribute value to a partition
- *Composite partitioning*: combines range and hash partitioning by first segregating data by ranges on the designated attribute, and then within each of these partitions

55

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

FILE ORGANIZATIONS

File organization: a technique for physically arranging the records of a file

Physical file: a named set of table rows stored in a contiguous section of secondary memory

56

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

ARRANGING TABLE ROWS

Objectives for choosing file organization

- Fast data retrieval
- High throughput for processing transactions
- Efficient use of storage space
- Protection from failures or data loss
- Minimizing need for reorganization
- Accommodating growth
- Security from unauthorized use

57

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

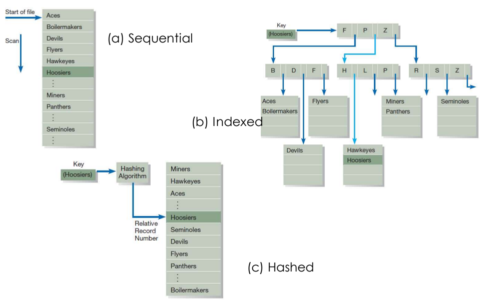
FILE ORGANIZATIONS (CONT.)

Three common file organizations:

1. **Sequential**: rows are stored in sequence according to a primary key value
2. **Indexed:** rows can be stored sequentially or nonsequentially; an index allows quick access to rows
3. **Hashed file organization**: rows usually stored nonsequentially; the address for each row is determined using an algorithm

58

Comparison of file organizations



59

File Organizations (Cont.)

	File Organization		
Factor	**Sequential**	**Indexed**	**Hashed**
Storage space	No wasted space	No wasted space for data, but extra space for index	Extra space may be needed to allow for addition and deletion of records
Sequential retrieval on primary key	Very fast	Moderately fast	Impractical
Random retrieval on primary key	Impractical	Moderately fast	Very fast
Multiple key retrieval	Possible, but requires scanning whole file	Very fast with multiple indexes	Not possible
Deleting rows	Can create wasted space or require reorganizing	If space can be dynamically allocated, this is easy, but requires maintenance of indexes	Very easy
Adding rows	Requires rewriting file	If space can be dynamically allocated, this is easy, but requires maintenance of indexes	Very easy, except multiple keys with same address require extra work
Updating rows	Usually requires rewriting file	Easy, but requires maintenance of indexes	Very easy

60

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

INDEXED FILE ORGANIZATION

Indexed file organization: a file organization in which rows are stored either sequentially or nonsequentially, and an index is created that allows software to locate individual rows

Index: a table used to determine the location of rows in a file that satisfy some condition

Secondary keys: one or a combination of fields for which more than one row may have the same combination of values

61

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

INDEXED FILE ORGANIZATION (CONT.)

Main disadvantages:
- Extra space required to store the indexes
- Extra time necessary to access and maintain indexes

Main advantage:
- Allows for both random and sequential processing

Guidelines for choosing indexes
- Specify a unique index for the primary key of each table.
- Specify an index for foreign keys.
- Specify an index for nonkey fields that are referenced in qualification, sorting and grouping commands for the purpose of retrieving data.

62

Introduction
Database Design
Normalization
Transforming E-R Diagrams into Relations
Merging Relations
Physical File and Database Design
Electronic Commerce Application

DESIGNING CONTROLS FOR FILES

Two of the goals of physical table design are *protection from failure or data loss* and *security from unauthorized use.*

These goals are achieved primarily by implementing controls on each file.

Two other important types of controls *address file backup* and *security*.

63

Introduction

Database Design

Normalization

Transforming E-R Diagrams into Relations

Merging Relations

Physical File and Database Design

Electronic Commerce Application

DESIGNING CONTROLS FOR FILES (CONT.)

Techniques for file restoration include:

- Periodically making a backup copy of a file.
- Storing a copy of each change to a file in a transaction log or audit trail.
- Storing a copy of each row before or after it is changed.

Means of building data security into a file include:

- Coding, or encrypting, the data in the file.
- Requiring data file users to identify themselves by entering user names and passwords.
- Prohibiting users from directly manipulating any data in the file by forcing users to work with a copy (real or virtual).

64

SUMMARY

In this unit you learned how to:

- ✓Describe the database design process, its outcomes, and the relational database model.
- ✓Describe normalization and the rules for second and third normal form.
- ✓Transform an entity-relationship (E-R) diagram into an equivalent set of well-structured (normalized) relations.
- ✓Merge normalized relations from separate user views into a consolidated set of well-structured relations.

✓Describe physical database design concepts:

- ✓Choose storage formats for fields in database tables.
- ✓Translate well-structured relations into efficient database tables.
- ✓Explain when to use different types of file organizations to store computer files.
- ✓Describe the purpose of indexes and the important considerations in selecting attributes to be indexed.

65

SUMMARY (CONT.)

66