

A.I.S. Class 14: Outline

- Learning Objectives for Chapter 9
- Designing the Data Repository Structure
- Database Normalization
- Group Work for Chapter 9
- CLASSROOM PRESENTATION 2

Learning Objectives for Chapter 9

- After studying this chapter you should be able to:
 - ◆ convert an extended entity relationship diagram to relational tables
 - ◆ articulate the conversion rules for mandatory relationships
 - ◆ articulate the conversion rules for optional relationships
 - ◆ explain the concept of database normalization

Learning Objectives for Chapter 9

- ◆ describe the rules for determining whether a table conforms to first, second, or third normal form
- ◆ explain the process of implementing tables in a relational database system such as Microsoft Access
- ◆ explain how forms in a relational database management system are used to implement information processes

Designing the Data Repository Structure for RDBMS

- There are three main issues involved in this step:
 - * Identifying the required tables
 - * Linking the tables
 - * Specifying the attributes for the tables

Linking Tables

■ 1:1 Links

- * collapse the two entities into a single table
or
- * post the primary key of either table as a foreign key in the other

■ 1:M and M:1 Links

- * post the primary key for the '1' table as a foreign key in the 'M' table

■ M:M Links

- * create a new Link Table with the primary keys from the original tables forming a composite key; add any uniquely defined attributes

Linking Tables

■ Except:

- * For optional entities

- ◆ always treat as if their cardinality were M
- ◆ i.e., treat both (0,1) and (0,*) entities as (0,*)

- * When modeling two events linked 1:1

- ◆ post the key of the first event as a foreign key in the table for the second

- * When modeling two events linked 1:M where the '1' event follows the 'M' event

- ◆ treat the link as a M:M link - i.e., create an additional Link Table

- * *Why do these exceptions almost always produce the same results as the methods set out in the chapter – because the second of two related events is almost always optional in the relationship with the first, and so is in any case treated as many*

Chapter 8 Problem 10: Data Repository Structure

- Submit (Submission#, Date, [Scientist#], [Proposal#])
- Check (Check#, Date, [Proposal#], [Submission#], Accepted?)
- Approve (Approval#, Date, [Director#], [Proposal#], [Check#], Approved/Rejected?)
- Return (Return#, Date, [Director#], [Proposal#], [Approval#], ReasonForRejection)
- Chemicals (Chemical#, Description, . . .)
- Equipment (Equipment#, Description, . . .)
- Proposals (Proposal#, Title?, Abstract?)
- Scientists (Scientist#, Name, Office, Phone#, . . .)
- Directors (Director#, Name, Office, Phone#, . . .)
- *No secretary table*

Chapter 8 Problem 10: Data Repository Structure

- Proposal-Scientist ([Proposal#], [Scientist#])
- Proposal-Equipment-Requested ([Proposal#], [Equipment#])
- Proposal-Chemicals-Requested ([Proposal#], [Chemical#], Quantity, Procedure)
- Proposal-Equipment-Used ([Proposal#], [Equipment#])
- Proposal-Chemicals-Used ([Proposal#], [Chemical#], Quantity, Procedure)
- *No reference tables needed*
- *Consider:*
SubmitProposal (Proposal#, [Scientist#], Date, Title, Abstract)

Chapter 8 Problem 9: Data Repository Structure

■ Mr. Video

- * EER
- * Data Repository Structure
- * See Handout

Data Repository Structures

■ Storing Balances

- * Technically wrong
- * Sometimes still done for convenience
- * Imposes extra processing burden to ensure correct

■ Recording agents: customers/vendors

- * Can you be certain who you shipped to and who you paid?
- * Omit if you can
- * For now (and next week!) assume not! . . .

■ Inventory

- * Are items uniquely identified (like cars) or not (like books)

■ Cash as a resource

- * Cash account numbers

Implementing the Design

- 1 Create the Access tables required by the design
- 2 Designate the primary keys
- 3 Establish relationships between tables
- 4 Create forms to maintain the tables for each resource and agent
- 5 Create (multi-table) forms for event recording processes
- 6 Create queries to generate desired information
- 7 Develop report formats for the desired reports
- 8 Build a custom menu system

Database Normalization

- Relational databases and other data processing systems can suffer from a number of anomalies
- Proper application of REA(L) modeling should avoid these problems by automatically generating 3NF data
- Problems may be encountered, though, with data designed in other ways

Database Normalization

■ Data anomalies

* Insertion anomalies

- ◆ inability to add certain data

* Deletion anomalies

- ◆ deleting data in one place causes a loss of other data that needs to be retained

* Update anomalies

- ◆ changes must be made in multiple locations

Database Normalization

■ Functional dependency

- * If in a table with two attributes, X and Y, there is only one possible value of Y for each possible value of X, Y is said to be functionally dependent on X
- * Thus secondary keys are functionally dependent on the primary key (although this is not the only kind of functional dependency)

Database Normalization

- Unnormalized tables
- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)

- Boyce/Codd Normal Form (BCNF)
- Fourth Normal Form (4NF)
- Fifth Normal Form (5NF)
- Domain/Key Normal Form (DK/NF)

Database Normalization

■ 1NF

- * No repeating groups

■ 2NF

- * 1NF + Full dependence on the entire key

■ 3NF

- * 2NF + No transitive dependencies (i.e., functional dependencies not originating from the primary key)

Fabulous Furniture Forum

Fun Family Furniture

Sales Invoice

Customer:

Benjamin Sisko
Deep Space Nine
Alpha Quadrant

Customer ID: 9876

Invoice #: 12345

Date: 12/31/2350

Salesman: Jean-Luc Picard

1 x Executive Desk @ \$30,000	30,000
-------------------------------	--------

2 x Office Chairs @ \$500	1,000
---------------------------	-------

Subtotal:	31,000
------------------	--------

Intergalactic Shipping	14,000
------------------------	--------

Total:	\$55,000
---------------	----------

Database Normalization

- What data is on this invoice?
- (Customer#, Customer, Address, Invoice#, Date, Salesman, {Qty, Item, Price, Value}, Shipping, Total)
 - * This is unnormalized data because of the recurring items
 - * Value and Total should not be stored because they can be calculated from other data

Database Normalization

- (Invoice#, Customer#, Customer, Address, Date, Salesman, Shipping)
- ([Invoice#], Line#, Qty, Item, Price)
 - * These tables are 1NF
 - * The Invoice table is 2NF but not 3NF because Customer and Address depend on Customer# which depends on Invoice#
 - * Salesman names and Item descriptions are repeated on these tables unnecessarily
 - * On the Invoice-Line table Price is not determined by the key at all - we still need to know the Item number

Database Normalization

- (Invoice#, [Customer#], Date, [Salesman#], Shipping)
- (Customer#, Customer, Address, . . .)
- (Salesman#, Salesman, . . .)
- (Item#, Item, . . .)
- ([Invoice#], Line#, [Item#], Qty, Price)
 - * But now the final table is not 2NF because Price depends only on [Item#] - and Line# is useless

Database Normalization

- (Invoice#, [Customer#], Date, [Salesman#], Shipping)
- (Customer#, Customer, . . .)
- (Salesman#, Salesman, . . .)
- (Item#, Item, Price, . . .)
- ([Invoice#], [Item#], Qty)

* These tables are 3NF

Chapter 9 Appendix

- Reading the Appendix at this stage helps orient you with respect to what needs to be done
- However, we will be studying ACCESS much more thoroughly in the second half of the semester
- Consequently, I have decided not to hold you responsible for this Appendix on the Mid-Term examination

Group Work for Chapter 9

- Problems 8, 9 & 12
- Remember to look at the Discussion

Questions

CLASSROOM PRESENTATION

■ Group Project Stage 2