System Analysis and Design

Eighth Edition

Alan Dennis, Barbara Wixom, Roberta M. Roth

Chapter 10

Data Storage Design

Objectives

- Become familiar with several file and database formats.
- Describe several goals of data storage.
- Be able to revise a logical ERD into a physical ERD.
- Be able to optimize a relational database for data storage and data access.
- Become familiar with indexes.
- Be able to estimate the size of a database.

Introduction

- ***Data storage function***: how data is stored and handled by programs that run the system.
- ***Data storage design***:
  - Select the data storage format
  - Convert the logical data model into a physical data model to reflect implementation decisions
  - Ensure that DFDs and ERDs balance
  - Design the selected data storage format to optimize its processing efficiency

Data Storage Formats

- Two main types of data storage formats:
    - Files: electronic lists of data, optimized to perform a particular transaction
    - Database: a collection of groupings of information that are related to each other in some way
- Database Management System (DBMS): software that creates and manipulates the databases

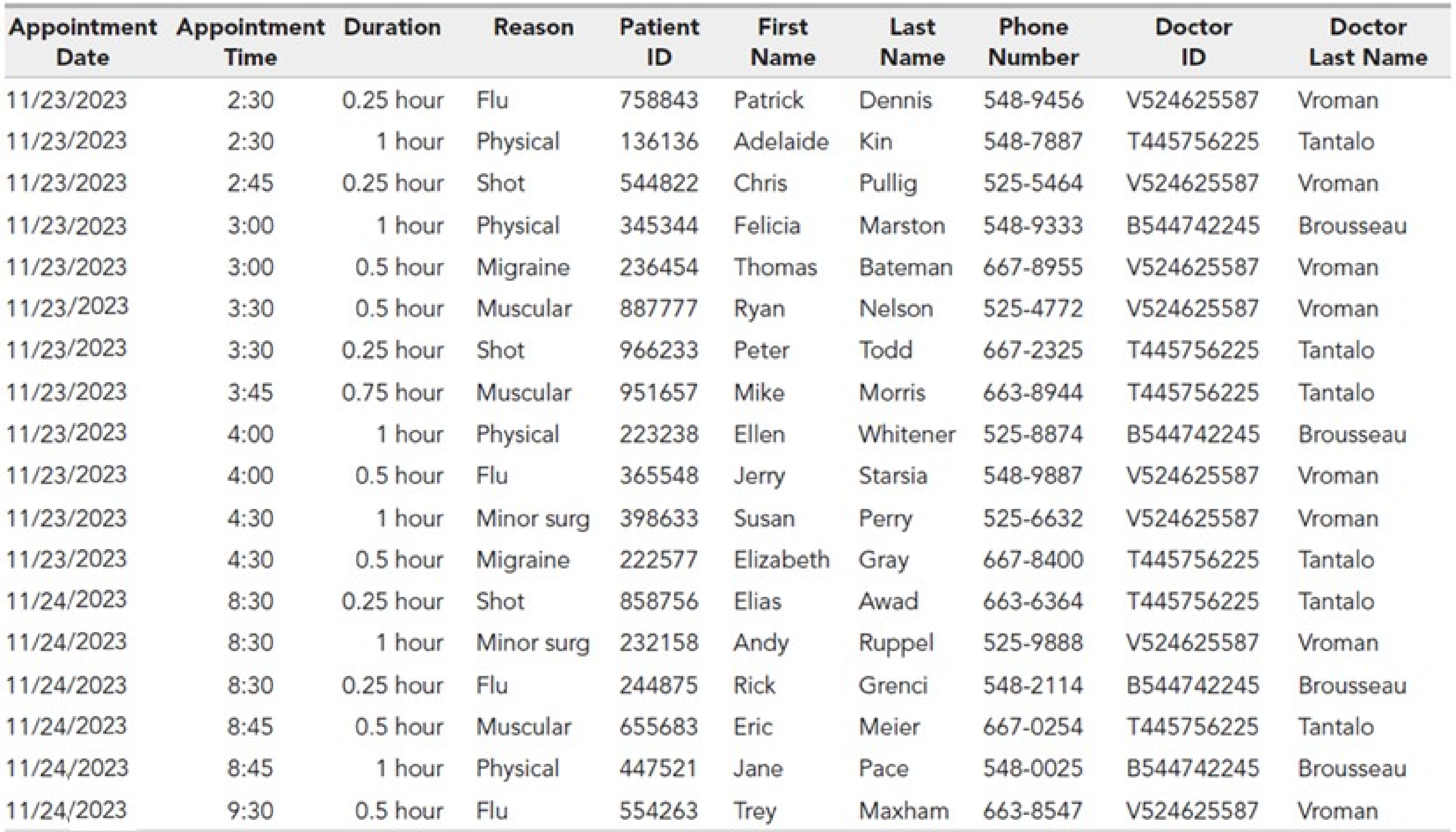
Files

- ***Data file***: an electronic list of information that is formatted for a particular transaction
- Sequential organization is typical
- Record associations with other records created by pointers
- Also called linked lists because of the way the records are linked together using pointers

Types of Files

- ***Master files*** – store core information that is important to the application
- ***Look-up files*** – contain static values
- ***Transaction files*** – store information that can be used to update a master file
- ***Audit files*** – record “before” and “after” images of data as the data is altered
- ***History files*** (or archive files) – store past transactions

Appointment File Example



Appointment Date	Appointment Time	Duration	Reason	Patient ID	First Name	Last Name	Phone Number	Doctor ID	Doctor Last Name
11/23/2023	2:30	0.25 hour	Flu	758843	Patrick	Dennis	548-9456	V524625587	Vroman
11/23/2023	2:30	1 hour	Physical	136136	Adelaide	Kin	548-7887	T445756225	Tantalo
11/23/2023	2:45	0.25 hour	Shot	544822	Chris	Pullig	525-5464	V524625587	Vroman
11/23/2023	3:00	1 hour	Physical	345344	Felicia	Marston	548-9333	B544742245	Brousseau
11/23/2023	3:00	0.5 hour	Migraine	236454	Thomas	Bateman	667-8955	V524625587	Vroman
11/23/2023	3:30	0.5 hour	Muscular	887777	Ryan	Nelson	525-4772	V524625587	Vroman
11/23/2023	3:30	0.25 hour	Shot	966233	Peter	Todd	667-2325	T445756225	Tantalo
11/23/2023	3:45	0.75 hour	Muscular	951657	Mike	Morris	663-8944	T445756225	Tantalo
11/23/2023	4:00	1 hour	Physical	223238	Ellen	Whitener	525-8874	B544742245	Brousseau
11/23/2023	4:00	0.5 hour	Flu	365548	Jerry	Starsia	548-9887	V524625587	Vroman
11/23/2023	4:30	1 hour	Minor surg	398633	Susan	Perry	525-6632	V524625587	Vroman
11/23/2023	4:30	0.5 hour	Migraine	222577	Elizabeth	Gray	667-8400	T445756225	Tantalo
11/24/2023	8:30	0.25 hour	Shot	858756	Elias	Awad	663-6364	T445756225	Tantalo
11/24/2023	8:30	1 hour	Minor surg	232158	Andy	Ruppel	525-9888	V524625587	Vroman
11/24/2023	8:30	0.25 hour	Flu	244875	Rick	Grenci	548-2114	B544742245	Brousseau
11/24/2023	8:45	0.5 hour	Muscular	655683	Eric	Meier	667-0254	T445756225	Tantalo
11/24/2023	8:45	1 hour	Physical	447521	Jane	Pace	548-0025	B544742245	Brousseau
11/24/2023	9:30	0.5 hour	Flu	554263	Trey	Maxham	663-8547	V524625587	Vroman

Types of Databases

- Legacy database
- Relational database
- Object database
- Multidimensional database
- NoSQL database

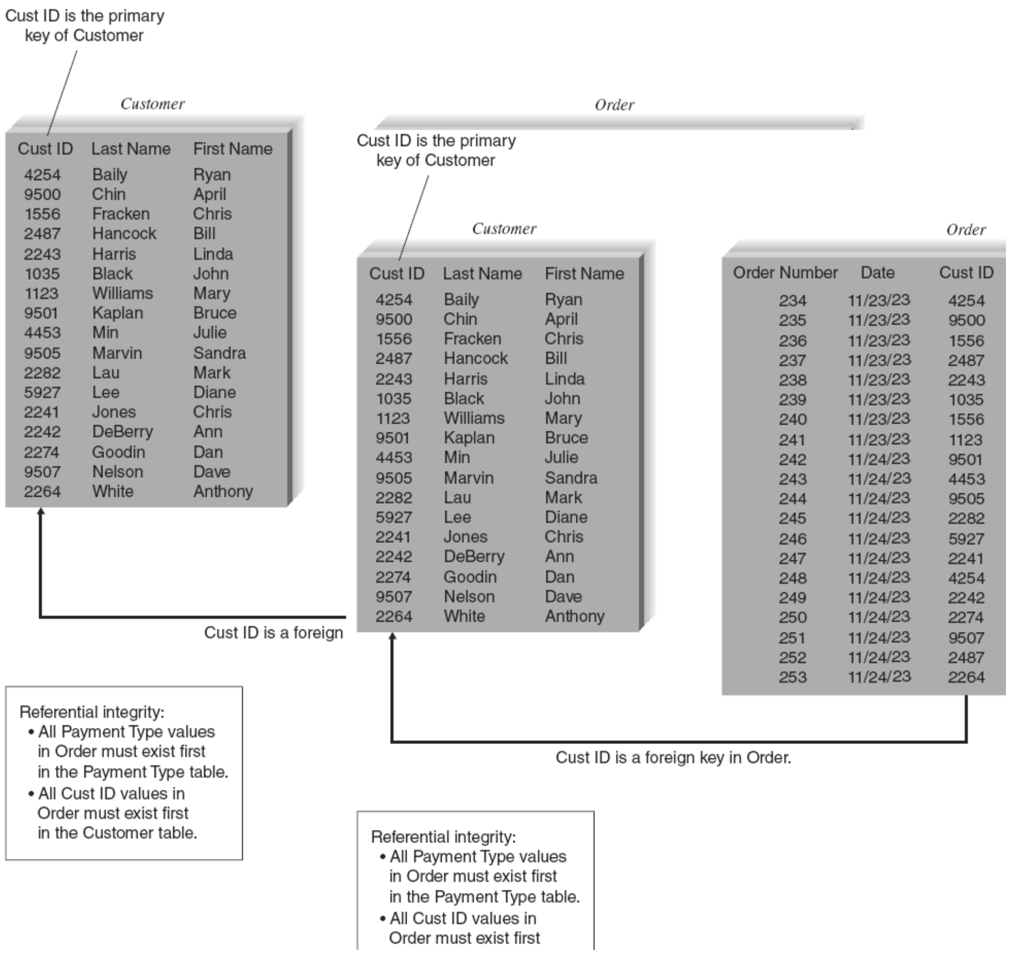
Legacy Databases

- Databases which are based on older technology; seldom used to develop new applications
- Two major types:
  - ***Hierarchical databases*** use hierarchies, or inverted trees, to represent relationships
  - ***Network databases*** are collections of records that are related to each other through pointers

Relational Databases

- The most popular kind of database for application development today
- Based on collections of ***tables***, each of which has a ***primary key***
- Tables are related to each other by the placing the primary key from one table into the related table as a ***foreign key***
- Most relational database management systems (RDBMS) support ***referential integrity***
  - Ensures that values linking the tables together are valid and correctly synchronized
- ***Structured Query Language*** (SQL) is the standard language for accessing the data in the tables

Appointment Database



Cust ID is the primary key of Customer

Customer

Cust ID	Last Name	First Name
4254	Baily	Ryan
9500	Chin	April
1556	Fracken	Chris
2487	Hancock	Bill
2243	Harris	Linda
1035	Black	John
1123	Williams	Mary
9501	Kaplan	Bruce
4453	Min	Julie
9505	Marvin	Sandra
2282	Lau	Mark
5927	Lee	Diane
2241	Jones	Chris
2242	DeBerry	Ann
2274	Goodin	Dan
9507	Nelson	Dave
2264	White	Anthony

Order

Cust ID is a foreign

Referential integrity:
- All Payment Type values in Order must exist first in the Payment Type table.
- All Cust ID values in Order must exist first in the Customer table.

Cust ID is the primary key of Customer

Customer

Cust ID	Last Name	First Name
4254	Baily	Ryan
9500	Chin	April
1556	Fracken	Chris
2487	Hancock	Bill
2243	Harris	Linda
1035	Black	John
1123	Williams	Mary
9501	Kaplan	Bruce
4453	Min	Julie
9505	Marvin	Sandra
2282	Lau	Mark
5927	Lee	Diane
2241	Jones	Chris
2242	DeBerry	Ann
2274	Goodin	Dan
9507	Nelson	Dave
2264	White	Anthony

Order

Order Number	Date	Cust ID
234	11/23/23	4254
235	11/23/23	9500
236	11/23/23	1556
237	11/23/23	2487
238	11/23/23	2243
239	11/23/23	1035
240	11/23/23	1556
241	11/23/23	1123
242	11/24/23	9501
243	11/24/23	4453
244	11/24/23	9505
245	11/24/23	2282
246	11/24/23	5927
247	11/24/23	2241
248	11/24/23	4254
249	11/24/23	2242
250	11/24/23	2274
251	11/24/23	9507
252	11/24/23	2487
253	11/24/23	2264

Cust ID is a foreign key in Order.

Referential integrity:
- All Payment Type values in Order must exist first in the Payment Type table.
- All Cust ID values in Order must exist first

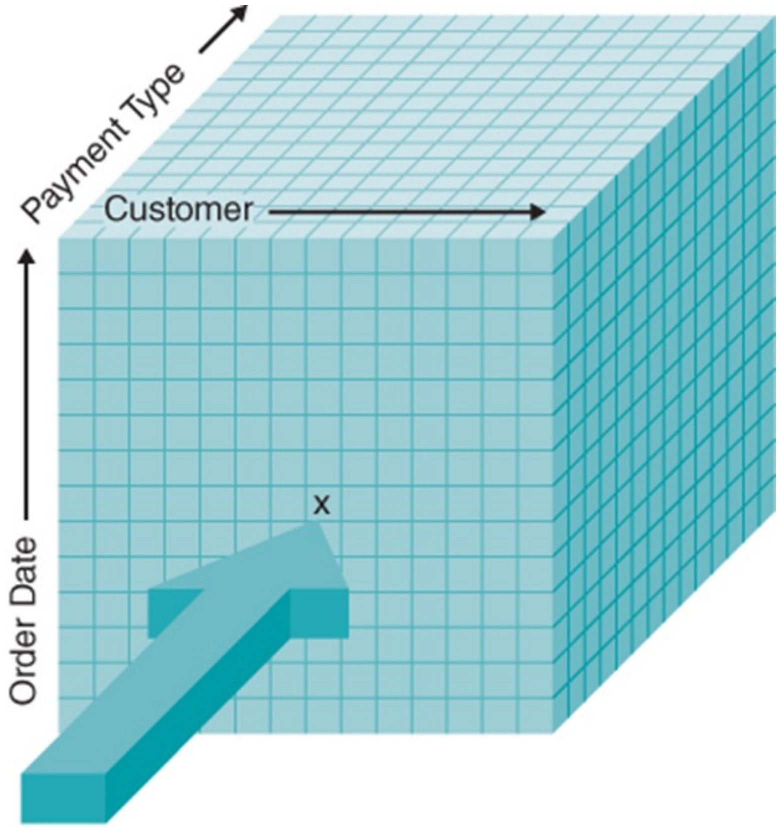
Object Databases

- Based on object orientation
  - All things should be treated as ***objects*** having both data (attributes) and processes (behaviors)
- ***Object-oriented database management systems*** (OODBMs) are mainly used to support multimedia applications or systems that involve complex data.
- Play a minor role in the DBMS market currently

Multidimensional Databases

- A type of relational database used extensively in data warehousing.
- ***Data warehousing*** is the practice of taking and storing data in a data warehouse (that is a large database) that supports ***business intelligence*** (BI) systems
- Data marts are smaller databases based on data warehouse data; support BI for specific departments or functional areas of the organization
- Stores data to support ***aggregations*** of data on multiple dimensions
- When the data are first loaded into a multidimensional database, the database precalculates the data across the

Multidimensional Databases Example



Last quarter, how many customers placed more than one order, using an American Express card?

NoSQL Databases

- Newest database approach; not based on the relational model or SQL
- Rapid processing on replicated database servers in the cloud
- Various types include:
  - ***Document-oriented databases***: manage collection of documents of varying forms and structures (example: Mongo DB)
  - ***Wide column databases***: store data in records holding very large numbers of dynamic columns (potentially billions of columns). Example: Bigtable, Cassandra, Dynamo
  - ***Graph databases***: a collection of nodes and edges using graph theory to store, map, and query relationships

Selecting a Storage Format

- Each of the file and database data storage format has its strengths and weaknesses
- Factors to consider in selecting a storage format:
  - Data types
  - Type of application system
  - Existing storage formats
  - Future needs

Comparison of Data Storage Formats

	Files	Legacy databases	Relational databases	Object databases	Multidimensional databases	NoSQL databases
Major strengths	Files can be designed for fast performance; good for short-term data storage.	Very mature products	Leader in the database market; can handle fast updating and querying needs	Able to handle complex data	Configured to answer business intelligence questions quickly	Designed for huge, varied data sets
Major weaknesses	Redundant data; data must be updated using programs.	Not able to store data as efficiently; limited future	Cannot handle complex data	Limited market acceptance; skills are hard to find.	Highly specialized use; skills are hard to find	New in the market, highly specialized use; skills are hard to find
Data types supported	Simple	Not recommended for new systems	Simple	Complex (e.g., video, audio, images)	Aggregated	Mixed data sets with structured and unstructured components
Application system types supported	Transaction processing	Not recommended for new systems	Transaction processing and decision making	Transaction processing	Business intelligence	Business intelligence; finding patterns and relationships in mixed data
Existing data formats	Organization dependent	Organization dependent	Organization dependent	Organization dependent	Organization dependent	Organization dependent
Future needs	Limited future prospects	Poor future prospects	Good future prospects	Uncertain future prospects	Uncertain future prospects	New, uncertain future prospects

Moving from a Logical to Physical Data Model

- The ***logical entity relationship diagrams*** (ERD) depicts the “business view” of the data; omits implementation details
- Having determined the data storage format, ***physical data*** models are created to show implementation details and to explain more about the “how” of the final system

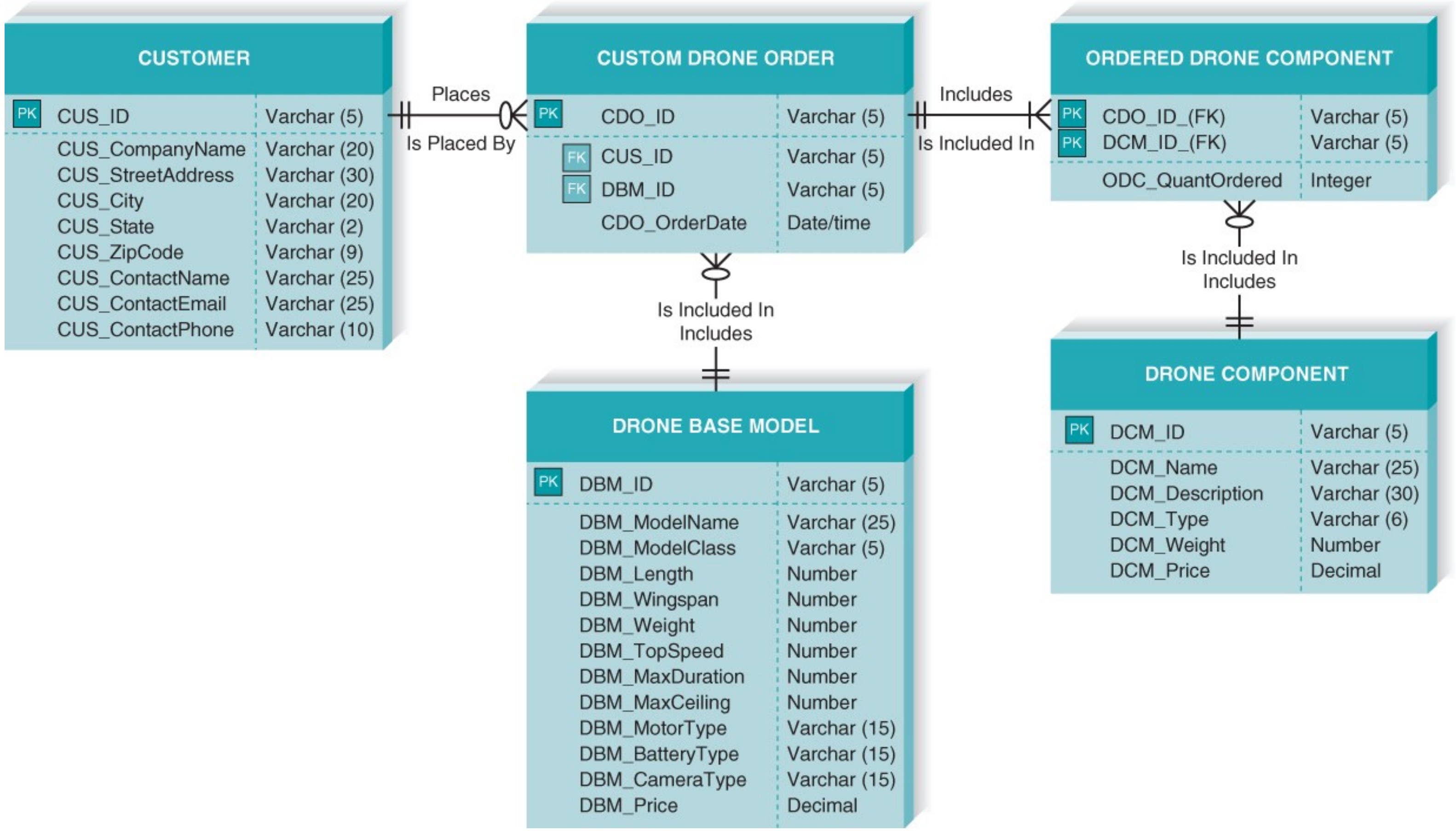
The Physical Entity Relationship Diagram

- The physical ERD includes entities, relationships, and attributes
- Adds references to how data will be stored
- Much more metadata is defined

Steps to Create the Physical ERD

Step	Explanation
1. Change entities to tables or files	Beginning with the logical ERD, change the entities to tables or files and update the metadata.
2. Change attributes to fields	Convert the attributes to fields and update the metadata.
3. Add primary keys	Assign primary keys to all entities.
4. Add foreign keys	Add foreign keys to represent the relationships among entities.
5. Add system-related components	Add system-related tables and fields.

Example Physical ERD



CUSTOMER		
PK	CUS_ID	Varchar (5)
	CUS_CompanyName	Varchar (20)
	CUS_StreetAddress	Varchar (30)
	CUS_City	Varchar (20)
	CUS_State	Varchar (2)
	CUS_ZipCode	Varchar (9)
	CUS_ContactName	Varchar (25)
	CUS_ContactEmail	Varchar (25)
	CUS_ContactPhone	Varchar (10)

CUSTOM DRONE ORDER		
PK	CDO_ID	Varchar (5)
FK	CUS_ID	Varchar (5)
FK	DBM_ID	Varchar (5)
	CDO_OrderDate	Date/time

ORDERED DRONE COMPONENT		
PK	CDO_ID_(FK)	Varchar (5)
PK	DCM_ID_(FK)	Varchar (5)
	ODC_QuantOrdered	Integer

DRONE BASE MODEL		
PK	DBM_ID	Varchar (5)
	DBM_ModelName	Varchar (25)
	DBM_ModelClass	Varchar (5)
	DBM_Length	Number
	DBM_Wingspan	Number
	DBM_Weight	Number
	DBM_TopSpeed	Number
	DBM_MaxDuration	Number
	DBM_MaxCeiling	Number
	DBM_MotorType	Varchar (15)
	DBM_BatteryType	Varchar (15)
	DBM_CameraType	Varchar (15)
	DBM_Price	Decimal

DRONE COMPONENT		
PK	DCM_ID	Varchar (5)
	DCM_Name	Varchar (25)
	DCM_Description	Varchar (30)
	DCM_Type	Varchar (6)
	DCM_Weight	Number
	DCM_Price	Decimal

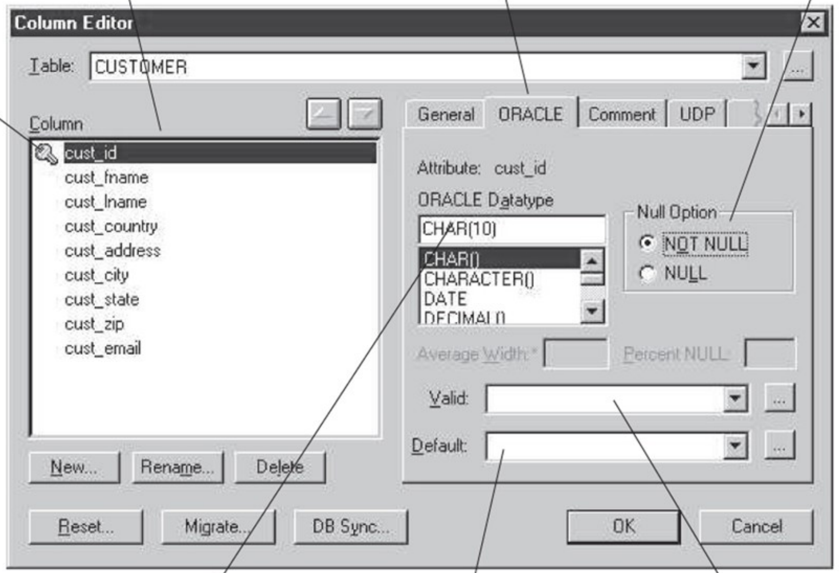
Physical Aspects of Data Element in Metadata

Naming conventions for fields: 4 digits of table name followed by the field name.

Notice that this will be implemented in Oracle.

No null, or blank, values will be accepted into the *cust_id* field.

The key signifies that *cust_id* is a primary key.



Column Editor

Table: CUSTOMER

Column:
- cust_id
- cust_fname
- cust_lname
- cust_country
- cust_address
- cust_city
- cust_state
- cust_zip
- cust_email

Attribute: cust_id

ORACLE Datatype: CHAR(10)

- CHAR()
- CHARACTER()
- DATE
- DECIMAL()

Null Option: ● NOT NULL ○ NULL

Average Width: Percent NULL:

Valid:

Default:

CHAR stands for "character" data type; the 10 stands for the number of characters.

The analyst can specify a default value that appears for this field.

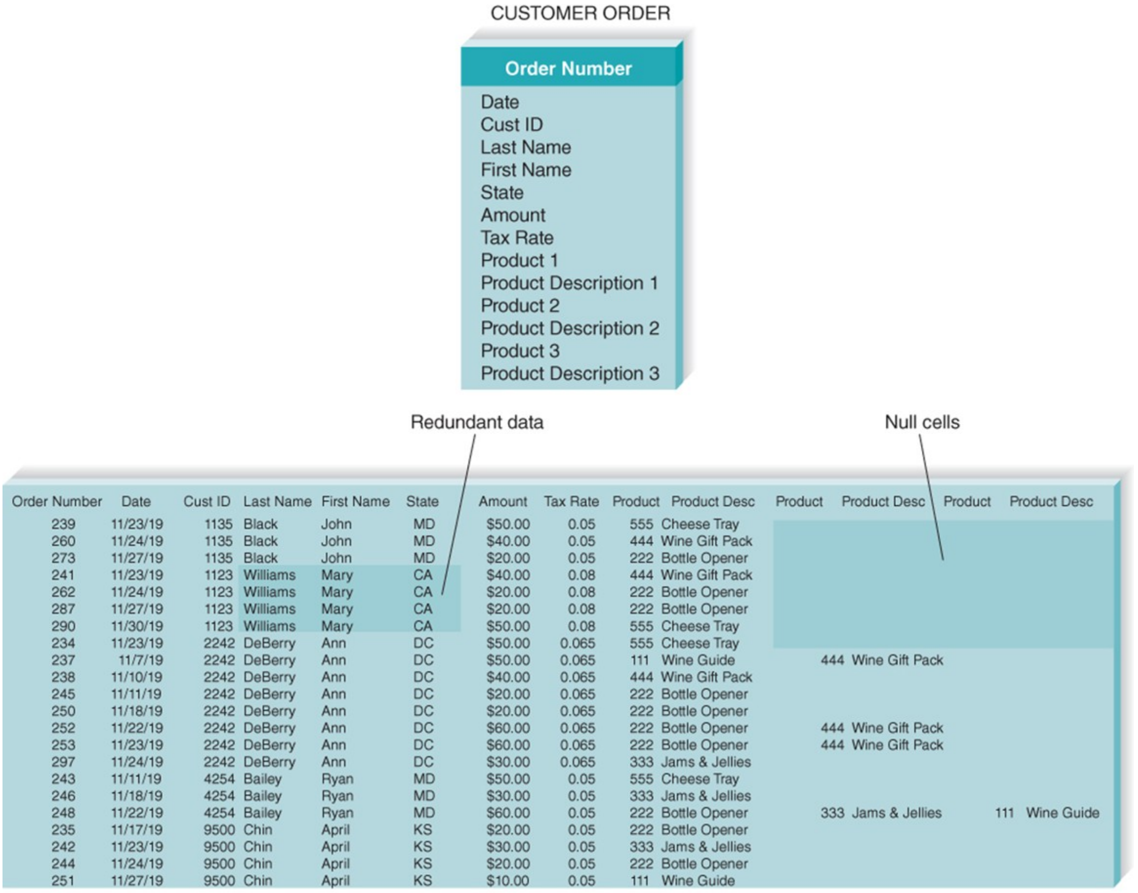
The analyst can develop a validation rule to be applied to this field.

Revising the CRUD Matrix

- It is important to verify that the system’s DFD and ERD models are balanced
- In design, logical models are converted into physical models
- Changes in the form of new processes, new data stores, and new data elements may occur
- The CRUD matrix should be revised

Optimizing Data Storage

- The data storage format is now optimized for processing ***efficiency***
- Two primary dimensions:
  - Storage efficiency
  - Speed of access
- Limit ***data redundancy***; very few null values
- Best way to achieve efficiency is ***normalization***



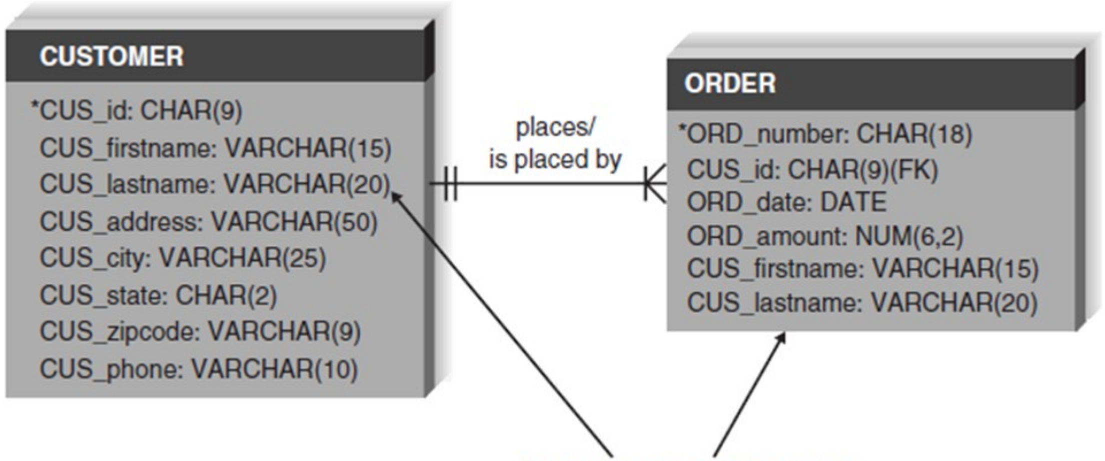
Order Number	Date	Cust ID	Last Name	First Name	State	Amount	Tax Rate	Product	Product Desc	Product	Product Desc	Product	Product Desc
239	11/23/19	1135	Black	John	MD	$50.00	0.05	555	Cheese Tray				
260	11/24/19	1135	Black	John	MD	$40.00	0.05	444	Wine Gift Pack				
273	11/27/19	1135	Black	John	MD	$20.00	0.05	222	Bottle Opener				
241	11/23/19	1123	Williams	Mary	CA	$40.00	0.08	444	Wine Gift Pack				
262	11/24/19	1123	Williams	Mary	CA	$20.00	0.08	222	Bottle Opener				
287	11/27/19	1123	Williams	Mary	CA	$20.00	0.08	222	Bottle Opener				
290	11/30/19	1123	Williams	Mary	CA	$50.00	0.08	555	Cheese Tray				
234	11/23/19	2242	DeBerry	Ann	DC	$50.00	0.065	555	Cheese Tray				
237	11/7/19	2242	DeBerry	Ann	DC	$50.00	0.065	111	Wine Guide	444	Wine Gift Pack		
238	11/10/19	2242	DeBerry	Ann	DC	$40.00	0.065	444	Wine Gift Pack				
245	11/11/19	2242	DeBerry	Ann	DC	$20.00	0.065	222	Bottle Opener				
250	11/18/19	2242	DeBerry	Ann	DC	$20.00	0.065	222	Bottle Opener				
252	11/22/19	2242	DeBerry	Ann	DC	$60.00	0.065	222	Bottle Opener	444	Wine Gift Pack		
253	11/23/19	2242	DeBerry	Ann	DC	$60.00	0.065	222	Bottle Opener	444	Wine Gift Pack		
297	11/24/19	2242	DeBerry	Ann	DC	$30.00	0.065	333	Jams & Jellies				
243	11/11/19	4254	Bailey	Ryan	MD	$50.00	0.05	555	Cheese Tray				
246	11/18/19	4254	Bailey	Ryan	MD	$30.00	0.05	333	Jams & Jellies				
248	11/22/19	4254	Bailey	Ryan	MD	$60.00	0.05	222	Bottle Opener	333	Jams & Jellies	111	Wine Guide
235	11/17/19	9500	Chin	April	KS	$20.00	0.05	222	Bottle Opener				
242	11/23/19	9500	Chin	April	KS	$30.00	0.05	333	Jams & Jellies				
244	11/24/19	9500	Chin	April	KS	$20.00	0.05	222	Bottle Opener				
251	11/27/19	9500	Chin	April	KS	$10.00	0.05	111	Wine Guide				

Optimizing Access Speed

- After optimizing for data storage efficiency, data are spread out across many tables
- For a large relational database, it is necessary to optimize access speed
- Techniques of optimizing access speed:
  - Denormalization
  - Clustering
  - Indexing
  - Estimating the size of data for hardware planning

Denormalization

- Add redundancy back into the design
- Reduce the number of joins required during processing to enhance data access speed



CUSTOMER

*CUS_id: CHAR(9)
CUS_firstname: VARCHAR(15)
CUS_lastname: VARCHAR(20)
CUS_address: VARCHAR(50)
CUS_city: VARCHAR(25)
CUS_state: CHAR(2)
CUS_zipcode: VARCHAR(9)
CUS_phone: VARCHAR(10)

ORDER

*ORD_number: CHAR(18)
CUS_id: CHAR(9)(FK)
ORD_date: DATE
ORD_amount: NUM(6,2)
CUS_firstname: VARCHAR(15)
CUS_lastname: VARCHAR(20)

Customer name will be stored in both tables.

Reasons to Denormalize

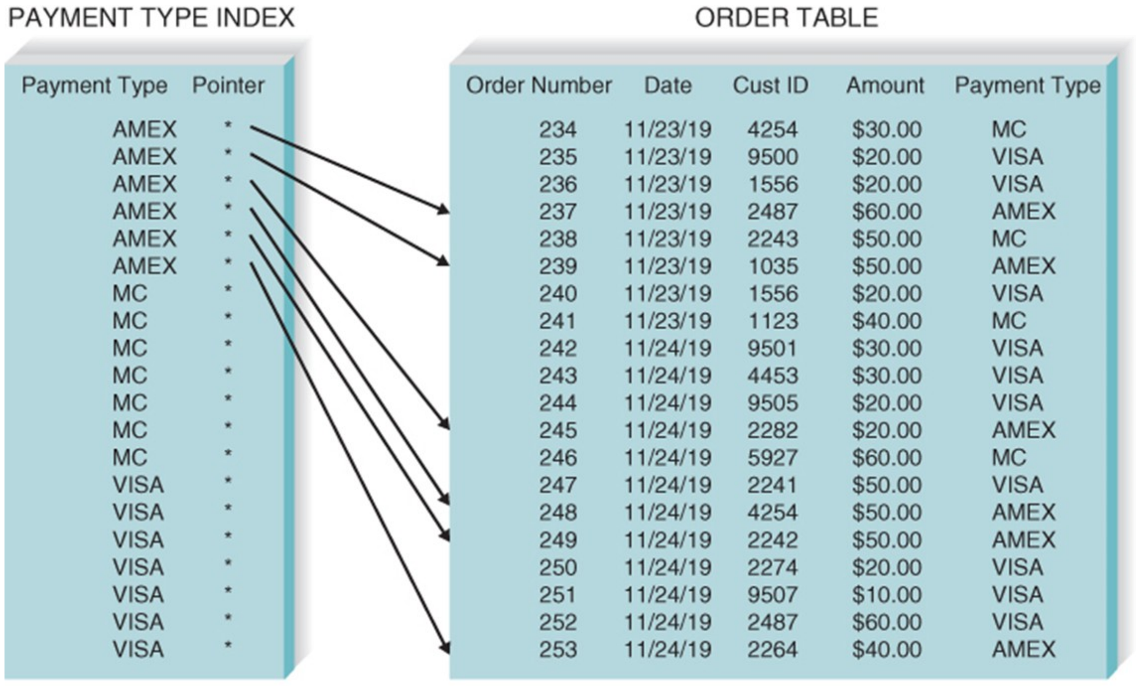
Reason	Description
Look-up table	Include a code’s description in the table using that code if the description is often used.
1:1 Relationships	Combine tables if they are related 1:1 and if they usually are accessed together.
1:N Relationships	Place fields from the parent (1) table into the child (N) table if the parent fields are used frequently with child information.
Star schema design	Data marts often are modeled with star schema design, which uses denormalization to maximize BI query performance.

Clustering

- Placing records together physically so that like records are stored close together.
- ***Intrafile clustering***: Similar records in the table are stored together
- ***Interfile clustering***: Combining records from more that one table that typically are retrieved together

Indexing

- A data storage ***index*** is a minitable (similar to an index of a book) containing values from one or more columns in a table and the location of the values within the table
- Indexes require overhead; they take up storage space



PAYMENT TYPE INDEX

Payment Type	Pointer
AMEX	*
AMEX	*
AMEX	*
AMEX	*
AMEX	*
AMEX	*
MC	*
MC	*
MC	*
MC	*
MC	*
MC	*
MC	*
VISA	*
VISA	*
VISA	*
VISA	*
VISA	*
VISA	*
VISA	*

ORDER TABLE

Order Number	Date	Cust ID	Amount	Payment Type
234	11/23/19	4254	$30.00	MC
235	11/23/19	9500	$20.00	VISA
236	11/23/19	1556	$20.00	VISA
237	11/23/19	2487	$60.00	AMEX
238	11/23/19	2243	$50.00	MC
239	11/23/19	1035	$50.00	AMEX
240	11/23/19	1556	$20.00	VISA
241	11/23/19	1123	$40.00	MC
242	11/24/19	9501	$30.00	VISA
243	11/24/19	4453	$30.00	VISA
244	11/24/19	9505	$20.00	VISA
245	11/24/19	2282	$20.00	AMEX
246	11/24/19	5927	$60.00	MC
247	11/24/19	2241	$50.00	VISA
248	11/24/19	4254	$50.00	AMEX
249	11/24/19	2242	$50.00	AMEX
250	11/24/19	2274	$20.00	VISA
251	11/24/19	9507	$10.00	VISA
252	11/24/19	2487	$60.00	VISA
253	11/24/19	2264	$40.00	AMEX

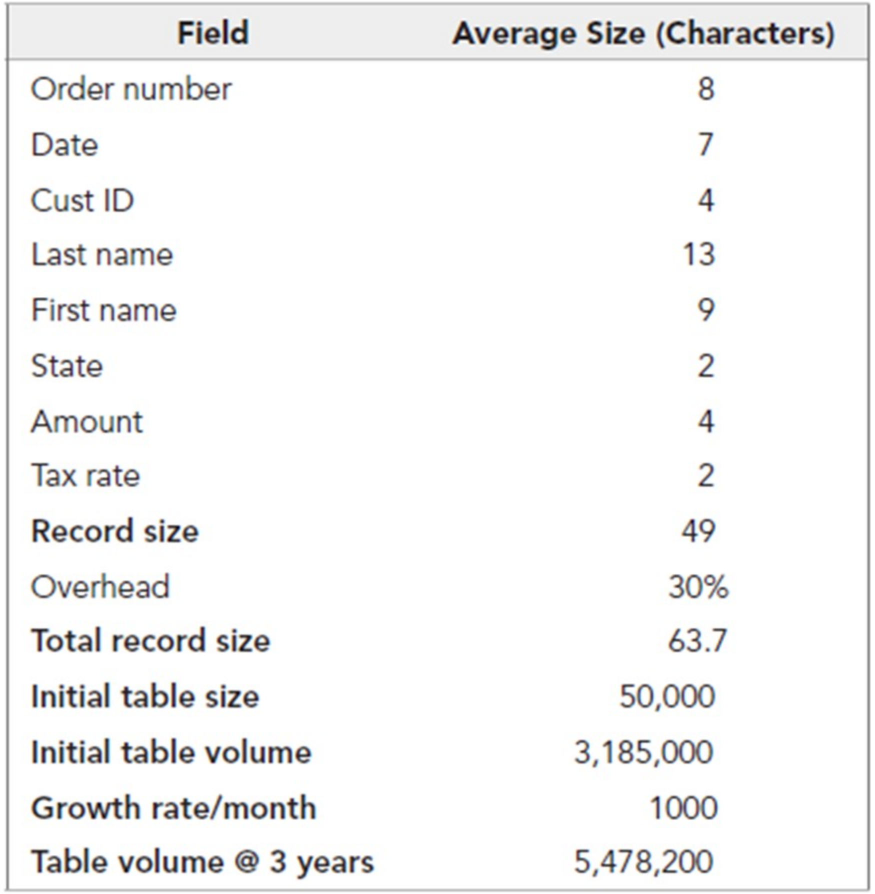
Indexing Guidelines

- Use indexes sparingly for transaction systems
- Use many indexes to improve response times in business intelligence systems
- For each table, create a unique index that is based on the primary key
- For each table, create an index that is based on the foreign key to improve the performance of joins
- Create an index for fields that are used frequently for grouping, sorting, or criteria

Estimating Storage Size

- ***Volumetrics*** – technique of estimating the amount of data that the hardware must support
- Steps:
  1. Calculate the amount of raw data - all the data stored within the database tables
  2. Calculate the overhead requirements based on the DBMS vendor’s recommendations
  3. Record the number of initial records loaded into the table, as well as the expected growth per month

Sample Volumetrics Calculation



Field	Average Size (Characters)
Order number	8
Date	7
Cust ID	4
Last name	13
First name	9
State	2
Amount	4
Tax rate	2
Record size	49
Overhead	30%
Total record size	63.7
Initial table size	50,000
Initial table volume	3,185,000
Growth rate/month	1000
Table volume @ 3 years	5,478,200

Chapter Review

- Identify and describe the purpose of the five type of files that are used to store business information.
- Identify and describe the purpose of the five type of databases that are used to store business information.
- Discuss the considerations to be made when selecting a data storage format.
- Discuss the five steps involved in converting the logical data model to a physical data model.
- Explain the modifications that may need to be made when updating the CRUD matrix

Chapter Review Continued

- Explain how to optimize the data storage design for storage efficiency.
- Explain the several reasons to apply denormalization to the data storage design.
- Explain the purpose and types of clustering that can be applied to the data storage design.
- Explain the purpose of indexing when applied to the data storage design.

Key Terms

- Aggregated
- Audit file
- Business intelligence (BI)
- Clustering
- Database
- Database management system (DBMS)
- Data mart
- Data warehousing
- Default value
- Denormalization
- End-user
- DBMS
- Enterprise DBMS
- Executive information system (EIS)
- Expert system (ES)
- Files
- Foreign key
- Hierarchical databases
- History file
- Index
- Interfile cluster
- Intrafile clustering
- Legacy database
- Linked list
- Logical entity relationship diagram
- Look-up table
- Look-up files

Key Terms Continued

- Management information system (MIS)
- Master files
- Multidimensional database
- Network databases
- Normalization
- NoSQL databases
- Object database
- Object-oriented
- (DBMS)
- Overhead
- Physical data model
- Physical entity relationship diagram
- Pointer
- Primary key
- Raw data
- Referential integrity
- Relational database
- Set
- Star schema design
- Structured Query Language
- (SQL)
- Table scan
- Transaction file
- Transaction processing systems
- Valid value
- Volumetrics