

- (2) Does this design eliminate the potential for data integrity problems that occur in the spreadsheet? Why or why not?
 - (3) Design a database for the data model that uses *Work_Version2*. Specify key and foreign key columns.
 - (4) Design a database for the data models that uses *Work_Version3*. Specify key and foreign key columns.
 - (5) Is the design with *Work_Version2* better than the design for *Work_Version3*? Why or why not?
- d. Figure 5-30c shows a third alternative data model for the sheet-music-tracking problem. In this data model, use either *Work_Version2* or *Work_Version3*, whichever you think is better.
- (1) Select identifiers for each entity in your data model. Justify your selection.
 - (2) Summarize the differences between this data model and that in Figure 5-30b. Which data model is better? Why?
 - (3) Design a database for this data model. Specify key and foreign key columns.
- e. Which of the three data models is the best? Justify your answer.

CASE STUDY 5

Fail Away with Dynamo, Bigtable, and Cassandra

As you learned in Case Study 1, Amazon.com processed more than 306 order items per second on its peak day of the 2012 holiday sales season. To do that, it processed customer transactions on tens of thousands of servers. With that many computers, failure is inevitable. Even if the probability of any one server failing is .0001, the likelihood that not one out of 10,000 of them fails is .9999 raised to the 10,000 power, which is about .37. Thus, for these assumptions the likelihood of at least one failure is 63 percent. For reasons that go beyond the scope of this discussion, the likelihood of failure is actually much greater.

Amazon.com must be able to thrive, even in the presence of such constant failure. Or, as Amazon.com engineers stated: "Customers should be able to view and add items to their shopping cart even if disks are failing, network routes are flapping, or data centers are being destroyed by tornados."⁹

The only way to deal with such failure is to replicate the data on multiple servers. When a customer stores a Wish List, for example, that Wish List needs to be stored on different, geographically separated servers. Then, when (notice *when*, not *if*) a server with one copy of the Wish List fails, Amazon.com applications obtain it from another server.

Such data replication solves one problem but introduces another. Suppose that the customer's Wish List is stored on servers A, B, and C and server A fails. While server A is down, server B or C can provide a copy of the Wish List, but if the

customer changes it, that Wish List can only be rewritten to servers B and C. It cannot be written to A, because A is not running. When server A comes back into service, it will have the old copy of the Wish List. The next day, when the customer reopens his or her Wish List, two different versions exist: the most recent one on servers B and C and an older one on server A. The customer wants the most current one. How can Amazon.com ensure that it will be delivered? Keep in mind that 15.6 million orders are being shipped while this goes on.

None of the current relational DBMS products was designed for problems like this. Consequently, Amazon.com engineers developed Dynamo, a specialized data store for reliably processing massive amounts of data on tens of thousands of servers. Dynamo provides an always-open experience for Amazon.com's retail customers; Amazon.com also sells Dynamo store services to others via its S3 Web Services product offering.

Meanwhile, Google was encountering similar problems that could not be met by commercially available relational DBMS products. In response, Google created Bigtable, a data store for processing petabytes of data on hundreds of thousands of servers.¹⁰ Bigtable supports a richer data model than Dynamo, which means that it can store a greater variety of data structures.

Both Dynamo and Bigtable are designed to be **elastic**; this term means that the number of servers can dynamically increase and decrease without disrupting performance.

⁹DeCandia, et al., "Dynamo: Amazon's Highly Available Key-Value Store," Proceedings of the 21st ACM Symposium on Operating Systems Principles, Stevenson, WA, October 2007.

¹⁰Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber, "Bigtable: A Distributed Storage System for Structured Data," OSDI 2006: Seventh Symposium on Operating System Design and Implementation, Seattle, WA, last modified November 2006, <http://labs.google.com/papers/bigtable.html>.

In 2007, Facebook encountered similar data storage problems: Massive amounts of data, the need to be elastically scalable, tens of thousands of servers, and high volumes of traffic. In response to this need, Facebook began development on Cassandra, a data store that provides storage capabilities like Dynamo with a richer data model like Bigtable.^{11,12} Initially, Facebook used Cassandra to power its Inbox Search. By 2008, Facebook realized that it had a bigger project on its hands than it wanted and gave the source code to the open source community. As of 2012, Cassandra is used by Facebook, Twitter, Digg, Reddit, Cisco, and many others.

Cassandra, by the way, is a fascinating name for a data store. In Greek mythology, Cassandra was so beautiful that Apollo fell in love with her and gave her the power to see the future. Alas, Apollo's love was unrequited and he cursed her so that no one would ever believe her predictions. The name was apparently a slam at Oracle.

Cassandra is elastic and fault-tolerant; it supports massive amounts of data on thousands of servers and provides **durability**, meaning that once data is committed to the data store, it won't be lost, even in the presence of failure. One of the most interesting characteristics of Cassandra is that clients (meaning the programs that run Facebook, Twitter, etc.) can select the level of consistency that they need. If a client requests that all servers always be current, Cassandra will ensure that that happens, but performance will be slow. At the other end of the trade-off spectrum, clients can require no consistency, whereby performance is maximized. In between, clients can require that a majority of the servers that store a data item be consistent.

Cassandra's performance is vastly superior to relational DBMS products. In one comparison, Cassandra was found to be 2,500 times faster than MySQL for write operations and 23 times faster for read operations¹³ on massive amounts of data on hundreds of thousands of possibly failing computers!

QUESTIONS

- 5-5. Clearly, Dynamo, Bigtable, and Cassandra are critical technology to the companies that created them. Why did they allow their employees to publish academic papers about them? Why did they not keep them as proprietary secrets?
- 5-6. What do you think this movement means to the existing DBMS vendors? How serious is the NoSQL threat? Justify your answer. What responses by existing DBMS vendors would be sensible?
- 5-7. Is it a waste of your time to learn about the relational model and Microsoft Access? Why or why not?
- 5-8. Given what you know about AllRoad Parts, should it use a relational DBMS, such as Oracle Database or MySQL, or should it use Cassandra?
- 5-9. Suppose that AllRoad decides to use a NoSQL solution, but a battle emerges among the employees in the IT department. One faction wants to use Cassandra, but another faction wants to use a different NoSQL data store, named MongoDB (www.mongodb.org). Assume that you're Kelly, and Lucas asks for your opinion about how he should proceed. How do you respond?

¹¹ "Welcome to Apache Cassandra," The Apache Software Foundation, accessed June 2011, <http://cassandra.apache.org>.


¹² "The Cassandra Distributed Database," Parleys, accessed July 16, 2013, <http://www.parleys.com/#st=5&id=1866&sl=20>.

¹³ "The Cassandra Distributed Database," Slide 21.

KEY TERMS AND CONCEPTS

- Access 166
- Attributes 175
- BigData 185
- Bigtable 184
- Byte 160
- Cassandra 184
- Columns 160
- Crow's feet 177
- Crow's-foot diagram 177
- Data integrity problem 179
- Data model 175
- Database 160
- Database administration 168
- Database application 168
- Database management system (DBMS) 166
- DB2 166
- Durability 195
- Dynamo 184
- Elastic 194
- Entity 175
- Entity-relationship (E-R) data model 175
- Entity-relationship (E-R) diagrams 177
- Fields 160
- File 160
- Foreign keys 163
- Graphical queries 172
- Identifier 175
- Key 162
- Lost-update problem 174
- Many-to-many (N:M) relationships 177
- Maximum cardinality 178
- Metadata 163
- Minimum cardinality 178
- MongoDB 184
- Multi-user processing 174
- MySQL 166
- Normal forms 180
- Normalization 178
- NoSQL databases 184
- One-to-many (1:N) relationships 177
- Oracle Database 166
- Primary key 162
- Records 160
- Relation 163
- Relational databases 163
- Relationships 176
- Rows 160
- SQL Server 166
- Structured Query Language (SQL) 168
- Table 160
- Unified Modeling Language (UML) 175

MyMISLab

Go to mymislab.com to complete the problems marked with this icon .

USING YOUR KNOWLEDGE

- ★ 5-1. Draw an entity-relationship diagram that shows the relationships among a database, database applications, and users.
- ★ 5-2. Consider the relationship between *Adviser* and *Student* in Figure 5-21. Explain what it means if the maximum cardinality of this relationship is:
 - a. N:1
 - b. 1:1
 - c. 5:1
 - d. 1:5
- ★ 5-3. Identify two entities in the data entry form in Figure 5-28. What attributes are shown for each? What do you think are the identifiers?
- 5-4. Visit www.acxiom.com. Navigate the site to answer the following questions.
 - a. According to the Web site, what is Acxiom's privacy policy? Are you reassured by its policy? Why or why not?
 - b. Make a list of 10 different products that Acxiom provides.
 - c. Describe Acxiom's top customers.
 - d. Examine your answers in parts b and c and describe, in general terms, the kinds of data that Acxiom must be collecting to be able to provide those products to those customers.
 - e. What is the function of InfoBase?
 - f. What is the function of PersoniX?
 - g. In what ways might companies like Acxiom need to limit their marketing so as to avoid a privacy outcry from the public?
 - h. Should there be laws that govern companies like Acxiom? Why or why not?
 - i. Should there be laws that govern the types of data services that governmental agencies can buy from companies like Acxiom? Why or why not?