**Section 24.1 Introduction**

• A database (p. [1046](http://proquest.safaribooksonline.com/9780133813036/ch24_html#page_1046)) is an integrated collection of data. A database management system (DBMS; p.[1046](http://proquest.safaribooksonline.com/9780133813036/ch24_html#page_1046)) provides mechanisms for storing, organizing, retrieving and modifying data.

• Today’s most popular database management systems are relational database (p. [1046](http://proquest.safaribooksonline.com/9780133813036/ch24_html#page_1046)) systems.

• SQL (p. [1046](http://proquest.safaribooksonline.com/9780133813036/ch24_html#page_1046)) is the international standard language used to query (p. [1046](http://proquest.safaribooksonline.com/9780133813036/ch24_html#page_1046)) and manipulate relational data.

• Programs connect to, and interact with, relational databases via an interface—software that facilitates communications between a database management system and a program.

• A JDBC driver (p. [1046](http://proquest.safaribooksonline.com/9780133813036/ch24_html#page_1046)) enables Java applications to connect to a database in a particular DBMS and allows you to retrieve and manipulate database data.

#### Section 24.2 Relational Databases

• A relational database (p. [1047](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec3_html#page_1047)) stores data in tables (p. [1047](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec3_html#page_1047)). Tables are composed of rows (p. [1047](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec3_html#page_1047)), and rows are composed of columns in which values are stored.

• A table’s primary key (p. [1047](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec3_html#page_1047)) has a unique in each row.

• Each column (p. [1047](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec3_html#page_1047)) of a table represents a different attribute.

• The primary key can be composed of more than one column.

• An identity column is the SQL standard way to represent an autoincremented (p. [1048](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec5_html#page_1048)) column. The SQLIDENTITY keyword (p. [1048](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec5_html#page_1048)) marks a column as an identity column.

• A foreign key is a column in a table that must match the primary-key column in another table. This is known as the Rule of Referential Integrity (p. [1050](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec7_html#page_1050)).

• A one-to-many relationship (p. [1051](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec7_html#page_1051)) between tables indicates that a row in one table can have many related rows in a separate table.

• Every column in a primary key must have a value, and the value of the primary key must be unique. This is known as the Rule of Entity Integrity (p. [1051](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec7_html#page_1051)).

• Foreign keys enable information from multiple tables to be joined together. There’s a one-to-many relationship between a primary key and its corresponding foreign key.

#### Section 24.4.1 Basic SELECT Query

• The basic form of a query (p. [1052](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec4_html#page_1052)) is

  SELECT \* FROM *tableName*

where the asterisk (\*; p. [1052](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec4_html#page_1052)) indicates that all columns from tableName should be selected, andtableName specifies the table in the database from which rows will be retrieved.

• To retrieve specific columns, replace the \* with a comma-separated list of column names.

#### Section 24.4.2 WHERE Clause

• The optional WHERE clause (p. [1053](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec9_html#page_1053)) in a query specifies the selection criteria for the query. The basic form of a query with selection criteria (p. [1052](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec4_html#page_1052)) is

[**Click here to view code image**](http://proquest.safaribooksonline.com/9780133813036/app06_html#p1100pro01a)

SELECT *columnName1*, *columnName2*, *...* FROM *tableName* WHERE *criteria*

• The WHERE clause can contain operators <, >, <=, >=, =, <> and LIKE. LIKE (p. [1053](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec9_html#page_1053)) is used for string pattern matching (p. [1053](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec9_html#page_1053)) with wildcard characters percent (%) and underscore (\_).

• A percent character (%; p. [1053](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec9_html#page_1053)) in a pattern indicates that a string matching the pattern can have zero or more characters at the percent character’s location in the pattern.

• An underscore (\_ ; p. [1053](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec9_html#page_1053)) in the pattern string indicates a single character at that position in the pattern.

#### Section 24.4.3 ORDER BY Clause

• A query’s result can be sorted with the ORDER BY clause (p. [1055](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec11_html#page_1055)). The simplest form of an ORDER BY clause is

[**Click here to view code image**](http://proquest.safaribooksonline.com/9780133813036/app06_html#p1100pro02a)

SELECT *columnName1*, *columnName2*, *...* FROM *tableName* ORDER BY *column* ASC  
SELECT *columnName1*, *columnName2*, *...* FROM *tableName* ORDER BY *column* DESC

where ASC specifies ascending order, DESC specifies descending order and column specifies the column on which the sort is based. The default sorting order is ascending, so ASC is optional.

• Multiple columns can be used for ordering purposes with an ORDER BY clause of the form

[**Click here to view code image**](http://proquest.safaribooksonline.com/9780133813036/app06_html#p1100pro03a)

ORDER BY column1 sortingOrder, column2 sortingOrder, ...

• The WHERE and ORDER BY clauses can be combined in one query. If used, ORDER BY must be the last clause in the query.

#### Section 24.4.4 Merging Data from Multiple Tables: INNER JOIN

• An INNER JOIN (p. [1057](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec12_html#page_1057)) merges rows from two tables by matching values in columns that are common to the tables. The basic form for the INNER JOIN operator is:

[**Click here to view code image**](http://proquest.safaribooksonline.com/9780133813036/app06_html#p1101pro01a)

SELECT columnName1, columnName2, ...  
FROM table1  
INNER JOIN table2  
   ON table1.columnName = table2.columnName

The ON clause (p. [1057](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec12_html#page_1057)) specifies the columns from each table that are compared to determine which rows are joined. If a SQL statement uses columns with the same name from multiple tables, the column names must be fully qualified (p. [1057](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec12_html#page_1057)) by prefixing them with their table names and a dot (.).

#### Section 24.4.5 INSERT Statement

• An INSERT statement (p. [1058](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec12_html#page_1058)) inserts a new row into a table. The basic form of this statement is

[**Click here to view code image**](http://proquest.safaribooksonline.com/9780133813036/app06_html#p1101pro02a)

INSERT INTO *tableName* (*columnName1*, *columnName2*, *...*, *columnNameN*)  
   VALUES (*value1*, *value2*, *...*, *valueN*)

where tableName is the table in which to insert the row. The tableName is followed by a commaseparated list of column names in parentheses. The list of column names is followed by the SQL keyword VALUES (p. [1058](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec12_html#page_1058)) and a comma-separated list of values in parentheses.

• SQL uses single quotes (') to delimit strings. To specify a string containing a single quote in SQL, escape the single quote with another single quote (i.e., '').

#### Section 24.4.6 UPDATE Statement

• An UPDATE statement (p. [1059](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec13_html#page_1059)) modifies data in a table. The basic form of an UPDATE statement is

[**Click here to view code image**](http://proquest.safaribooksonline.com/9780133813036/app06_html#p1101pro03a)

UPDATE *tableName*  
   SET *columnName1* = *value1*, *columnName2* = *value2*, *...*, *columnNameN* = *valueN*  
   WHERE *criteria*

where tableName is the table to update. Keyword SET (p. [1059](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec13_html#page_1059)) is followed by a comma-separated list ofcolumnName = value pairs. The optional WHERE clause determines which rows to update.

#### Section 24.4.7 DELETE Statement

• A DELETE statement (p. [1060](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec15_html#page_1060)) removes rows from a table. The simplest form for a DELETE statement is

[**Click here to view code image**](http://proquest.safaribooksonline.com/9780133813036/app06_html#p1101pro04a)

DELETE FROM *tableName* WHERE *criteria*

where tableName is the table from which to delete a row (or rows). The optional WHERE criteriadetermines which rows to delete. If this clause is omitted, all the table’s rows are deleted.

#### Section 24.5 Setting up a Java DB Database

• Oracle’s pure Java database Java DB (p. [1060](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec15_html#page_1060)) is installed with the JDK on Windows, Mac OS X and Linux.

• Java DB has both an embedded version and a network version.

• The Java DB software is located in the db subdirectory of your JDK’s installation directory.

• Java DB comes with several files that enable you to configure and run it. Before executing these files from a command window, you must set the environment variable JAVA\_HOME to refer to the JDK’s exact installation directory.

• You use the setEmbeddedCP.bat or setEmbeddedCP file (depending on your OS platform) to configure theCLASSPATH for working with Java DB embedded databases.

• The Java DB ij tool allows you to interact with Java DB from the command line. You can use it to create databases, run SQL scripts and perform SQL queries. Each command you enter at the ij> prompt must be terminated with a semicolon (;).

#### Section 24.6.1 Connecting to and Querying a Database

• Package java.sql contains classes and interfaces for accessing relational databases in Java.

• A Connection object (p. [1065](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1065)) manages the connection between a Java program and a database.Connection objects enable programs to create SQL statements that access data.

• DriverManager (p. [1065](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1065)) method getConnection (p. [1065](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1065)) attempts to connect to a database at a URL that specifies the protocol for communication, the subprotocol (p. [1065](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1065)) for communication and the database name.

• Connection method createStatement (p. [1066](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1066)) creates a Statement object (p. [1066](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1066)), which can be used to submit SQL statements to the database.

• Statement method executeQuery (p. [1066](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1066)) executes a query and returns a ResultSet object (p. [1066](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1066)).ResultSet methods enable a program to manipulate query results.

• A ResultSetMetaData object (p. [1066](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1066)) describes a ResultSet’s contents. Programs can use metadata programmatically to obtain information about the ResultSet column names and types.

• ResultSetMetaData method getColumnCount (p. [1066](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1066)) retrieves the number of ResultSet columns.

• ResultSet method next (p. [1066](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1066)) positions the ResultSet cursor to the next row and returns true if the row exists; otherwise, it returns false. This method must be called to begin processing a ResultSetbecause the cursor is intially positioned before the first row.

• It’s possible to extract each ResultSet column as a specific Java type. ResultSetMetaData methodgetColumnType (p. [1066](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1066)) returns a Types (p. [1067](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1067)) constant (package java.sql) indicating the column’s type.

• ResultSet get methods typically receive as an argument either a column number (as an int) or a column name (as a String) indicating which column’s value to obtain.

• ResultSet row and column numbers start at 1.

• Each Statement object can open only one ResultSet at a time. When a Statement returns a new ResultSet, the Statement closes the prior ResultSet.

#### Section 24.6.2 Querying the books Database

• TableModel (p. [1067](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1067)) method getColumnClass (p. [1068](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec22_html#page_1068)) returns a Class object that represents the superclass of all objects in a particular column. A JTable (p. [1067](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec21_html#page_1067)) uses this information to set up the default cell renderer and cell editor for that column in a JTable.

• Connection method createStatement has an overloaded version that receives the result type and concurrency. The result type specifies whether the ResultSet’s cursor is able to scroll in both directions or forward only and whether the ResultSet is sensitive to changes. The result concurrency (p. [1072](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec22_html#page_1072)) specifies whether the ResultSet can be updated.

• Some JDBC drivers (p. [1072](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec22_html#page_1072)) do not support scrollable or updatable ResultSets.

• ResultSetMetaData method getColumnClassName (p. [1073](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec22_html#page_1073)) obtains a column’s fully qualified class name.

• TableModel method getColumnCount (p. [1073](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec22_html#page_1073)) returns the number of columns in the ResultSet.

• TableModel method getColumnName (p. [1073](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec22_html#page_1073)) returns the column name in the ResultSet.

• ResultSetMetaData method getColumnName (p. [1073](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec22_html#page_1073)) obtains a ResultSet column’s name.

• TableModel method getRowCount (p. [1073](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec22_html#page_1073)) returns the number of rows in the model’s ResultSet.

• TableModel method getValueAt (p. [1073](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec22_html#page_1073)) returns the Object at a particular row and column of the model’s underlying ResultSet.

• ResultSet method absolute (p. [1073](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec22_html#page_1073)) positions the ResultSet cursor at a specific row.

• AbstractTableModel method fireTableStructureChanged (p. [1074](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec22_html#page_1074)) notifies any JTable using a particularTableModel object as its model that the data in the model has changed.

#### Section 24.7 RowSet Interface

• Interface RowSet (p. [1080](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec7_html#page_1080)) configures a database connection and executes a query automatically.

• A connected RowSet (p. [1080](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec7_html#page_1080)) remains connected to the database while the object is in use. A disconnected RowSet (p. [1080](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec7_html#page_1080)) connects, executes a query, then closes the connection.

• JdbcRowSet (p. [1080](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec7_html#page_1080)) (a connected RowSet) wraps a ResultSet object and allows you to scroll and update its rows. Unlike a ResultSet object, a JdbcRowSet is scrollable and updatable by default.

• CachedRowSet (p. [1080](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec7_html#page_1080)), a disconnected RowSet, caches a ResultSet’s data in memory. A CachedRowSet is scrollable and updatable. A CachedRowSet is also serializable.

• Class RowSetProvider (package javax.sql.rowset; p. [1080](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec7_html#page_1080)) provides static method newFacto ry which returns an object that implements interface RowSetFactory (package javax.sql.row set) that can be used to create various types of RowSets.

• RowSetFactory (p. [1080](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec7_html#page_1080)) method createJdbcRowSet returns a JdbcRowSet object.

• JdbcRowSet method setUrl specifies the database URL.

• JdbcRowSet method setUsername specifies the username.

• JdbcRowSet method setPassword specifies the password.

• JdbcRowSet method setCommand specifies the SQL query that will be used to populate a RowSet.

• JdbcRowSet method execute executes the SQL query. Method execute performs establishes a Connectionto the database, prepares the query Statement, executes the query and stores the ResultSet returned by query. The Connection, Statement and ResultSet are encapsulated in the JdbcRowSet object.

#### Section 24.8 PreparedStatements

• PreparedStatements (p. [1082](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec24_html#page_1082)) are compiled, so they execute more efficiently than Statements.

• PreparedStatements can have parameters, so the same query can execute with different arguments.

• A parameter is specified with a question mark (?) in the SQL statement. Before executing aPreparedStatement, you must use PreparedStatement’s set methods to specify the arguments.

• PreparedStatement method setString’s (p. [1082](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec24_html#page_1082)) first argument represents the parameter number being set and the second argument is that parameter’s value.

• Parameter numbers are counted from 1, starting with the first question mark (?).

• Method setString automatically escapes String parameter values as necessary.

• Interface PreparedStatement provides set methods for each supported SQL type.

#### Section 24.9 Stored Procedures

• JDBC enables programs to invoke stored procedures (p. [1098](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec31_html#page_1098)) using CallableStatement (p. [1098](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec31_html#page_1098)) objects.

• CallableStatement can specify input parameters. CallableStatement can specify output parameters (p.[1098](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec31_html#page_1098)) in which a stored procedure can place return values.

#### Section 24.10 Transaction Processing

• Transaction processing (p. [1098](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec31_html#page_1098)) enables a program that interacts with a database to treat a database operation (or set of operations) as a single operation—known as an atomic operation (p. [1099](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec10_html#page_1099)) or a transaction (p. [1099](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec10_html#page_1099)).

• At the end of a transaction, a decision can be made to either commit or roll back the transaction.

• Committing a transaction (p. [1099](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec10_html#page_1099)) finalizes the database operation(s)—inserts, updates and deletes cannot be reversed without performing a new database operation.

• Rolling back a transaction (p. [1099](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec10_html#page_1099)) leaves the database in its state prior to the database operation.

• Java provides transaction processing via methods of interface Connection.

• Method setAutoCommit (p. [1099](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec10_html#page_1099)) specifies whether each SQL statement commits after it completes (a trueargument) or whether several SQL statements should be grouped as a transaction.

• When autocommit is disabled, the program must follow the last SQL statement in the transaction with a call to Connection method commit (to commit the changes to the database; p. [1099](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec10_html#page_1099)) or Connection methodrollback (to return the database to its state prior to the transaction; p. [1099](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec10_html#page_1099)).

• Method getAutoCommit (p. [1099](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec10_html#page_1099)) determines the autocommit state for the Connection.

### Self-Review Exercise

[**24.1**](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec14_html#ch24ans1) Fill in the blanks in each of the following statements:

a) The international standard database language is \_\_\_\_\_\_\_\_\_.

b) A table in a database consists of \_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_.

c) Statement objects return SQL query results as \_\_\_\_\_\_\_\_\_ objects.

d) The \_\_\_\_\_\_\_\_\_ uniquely identifies each row in a table.

e) SQL keyword \_\_\_\_\_\_\_\_\_ is followed by the selection criteria that specify the rows to select in a query.

f) SQL keywords \_\_\_\_\_\_\_\_\_ specify the order in which rows are sorted in a query.

g) Merging rows from multiple database tables is called \_\_\_\_\_\_\_\_\_ the tables.

h) A(n) \_\_\_\_\_\_\_\_\_ is an organized collection of data.

i) A(n) \_\_\_\_\_\_\_\_\_ is a set of columns whose values match the primary-key values of another table.

j)  \_\_\_\_\_\_\_\_\_ method \_\_\_\_\_\_\_\_\_ is used to obtain a Connection to a database.

k) Interface \_\_\_\_\_\_\_\_\_ helps manage the connection between a Java program and a database.

l) A(n) \_\_\_\_\_\_\_\_\_ object is used to submit a query to a database.

m) Unlike a ResultSet object, \_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_ objects are scrollable and updatable by default.

n) \_\_\_\_\_\_\_\_\_, a disconnected RowSet, caches the data of a ResultSet in memory.

### Answers to Self-Review Exercise

[**24.1**](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec13_html#ch24que1)

a) SQL.

b) rows, columns.

c) ResultSet.

d) primary key.

e) WHERE.

f) ORDER BY.

g) joining.

h) database.

i) foreign key.

j) DriverManager, getConnection.

k) Connection.

l) Statement.

m) JdbcRowSet, CachedRowSet.

n) CachedRowSet.

### Exercises

**24.2****(Query Application for the** ***books*** **Database)** Using the techniques shown in this chapter, define a complete query application for the books database. Provide the following predefined queries:

a) Select all authors from the Authors table.

b) Select a specific author and list all books for that author. Include each book’s title, year and ISBN. Order the information alphabetically by the author’s last name then by first name.

c) Select a specific title and list all authors for that title. Order the authors alphabetically by last name then by first name.

d) Provide any other queries you feel are appropriate.

Display a JComboBox with appropriate names for each predefined query. Also allow users to supply their own queries.

**24.3 (Data-Manipulation Application for the** ***books*** **Database)** Define a data-manipulation application for the books database. The user should be able to edit existing data and add new data to the database (obeying referential and entity integrity constraints). Allow the user to edit the database in the following ways:

a) Add a new author.

b) Edit the existing information for an author.

c) Add a new title for an author. (Remember that the book must have an entry in the AuthorISBNtable.).

d) Add a new entry in the AuthorISBN table to link authors with titles.

**24.4 (Employee Database)** In [Section 10.5](http://proquest.safaribooksonline.com/9780133813036/ch10lev1sec5_html#ch10lev1sec5), we introduced an employee-payroll hierarchy to calculate each employee’s payroll. In this exercise, we provide a database of employees that corresponds to the employee-payroll hierarchy. (A SQL script to create the employees database is provided with the examples for this chapter.) Write an application that allows the user to:

a) Add employees to the employee table.

b) Add payroll information to the appropriate table for each new employee. For example, for a salaried employee add the payroll information to the salariedEmployees table.

[Figure 24.33](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec15_html#ch24fig33) is the entity-relationship diagram for the employees database.

**24.5 (Employee Database Query Application)** Modify [Exercise 24.4](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec15_html#ch24que4) to provide a JComboBox and aJTextArea to allow the user to perform a query that is either selected from the JComboBox or defined in theJTextArea. Sample predefined queries are:

a) Select all employees working in Department SALES.

b) Select hourly employees working over 30 hours.

c) Select all commission employees in descending order of the commission rate.

**24.6 (Employee Database Data-Manipulation Application)** Modify [Exercise 24.5](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec15_html#ch24que5) to perform the following tasks:

a) Increase base salary by 10% for all base-plus-commission employees.

b) If the employee’s birthday is in the current month, add a $100 bonus.

c) For all commission employees with gross sales over $10,000, add a $100 bonus.

**24.7****(Address Book Modification: Update an Existing Entry)** Modify the program in [Figs. 24.30](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec26_html#ch24fig30)–[24.32](http://proquest.safaribooksonline.com/9780133813036/ch24lev2sec31_html#ch24fig32)to provide a JButton that allows the user to call a method named updatePerson in PersonQueries class to update the current entry in the AddressBook database.

**24.8 (Address Book Modification: Delete an Existing Entry)** Modify the program of [Exercise 24.7](http://proquest.safaribooksonline.com/9780133813036/ch24lev1sec15_html#ch24que7) to provide a JButton that allows the user to call a method named deletePerson in PersonQueries class to delete the current entry in the AddressBook database.

**24.9 (Optional Project: ATM Case Study with a Database)** Modify the optional ATM Case Study (online Chapters 33–34) to use an actual database to store the account information. We provide a SQL script to create BankDatabase, which has a single table consisting of four columns—AccountNumber (an int), PIN (anint), AvailableBalance (a double) and TotalBalance (a double).

**Fig. 24.33** | Table relationships in the employees database.