

Database: SQL, MySQL and Java DB

18.1 Introduction

- A database is an organized collection of data
- A database management system (DBMS) provides mechanisms for storing, organizing, retrieving and modifying data for many users
 - Today's most popular database management systems are relational databases
- SQL is the international standard language used almost universally with relational database systems to perform queries and manipulate data
- Programs connect to, and interact with, relational databases systems via an interface
 - Between a database management system and a program

18.2 Relational Databases

- A relational database stores data in tables
- Tables are composed of rows, and rows are composed of columns in which values are stored
- A primary key provides unique values that cannot be duplicated in other rows of the same table
- Each column of a table represents a different attribute in a row of data
- The primary key is a column (or group of columns) with a unique value that cannot be duplicated in other rows

	Number	Name	Department	Salary	Location
Row {	23603	Jones	413	1100	New Jersey
	24568	Kerwin	413	2000	New Jersey
	34589	Larson	642	1800	Los Angeles
	35761	Myers	611	1400	Orlando
	47132	Neumann	413	9000	New Jersey
	78321	Stephens	611	8500	Orlando
	Primary key		Column		

Fig. 18.1 | Employee table sample data.

Department	Location
413	New Jersey
611	Orlando
642	Los Angeles

Fig. 18.2 | Result of selecting distinct Department and Location data from table Employee.

18.3 Relational Database

Overview: A books Database

- The combined columns of the AuthorISBN table represent the table's primary key
 - Each row in this table must be a unique combination of an AuthorID and an ISBN
 - This is known as the Rule of Entity Integrity
- A one-to-many relationship between tables indicates that a row in one table can have many related rows in a separate table
- A foreign key is a column in a table that matches the primary-key column in another table

18.3 Relational Database

Overview: A books Database

- The foreign key helps maintain the Rule of Referential Integrity
 - Every foreign-key value must appear as another table's primary-key value
- Foreign keys also allow related data in multiple tables to be selected from those tables for analytic purposes
 - This is known as joining the data

Column	Description
AuthorID	Author's ID number in the database. In the books database, this integer column is defined as autoincremented—for each row inserted in this table, the AuthorID value is increased by 1 automatically to ensure that each row has a unique AuthorID. This column represents the table's primary key.
FirstName	Author's first name (a string).
LastName	Author's last name (a string).

Fig. 18.3 | Authors table from the books database.

AuthorID	FirstName	LastName
1	Paul	Deitel
2	Harvey	Deitel
3	Abbey	Deitel
4	Michael	Morgano
5	Eric	Kern

Fig. 18.4 | Sample data from the Authors table.

Column	Description
AuthorID	The author's ID number, a foreign key to the Authors table.
ISBN	The ISBN for a book, a foreign key to the Titles table.

Fig. 18.5 | AuthorISBN table from the books database.

AuthorID	ISBN	AuthorID	ISBN
1	0132152134	2	0132575663
2	0132152134	1	0132662361
1	0132151421	2	0132662361
2	0132151421	1	0132404168
1	0132575663	2	0132404168
1	013705842X	1	0132121360
2	013705842X	2	0132121360
3	013705842X	3	0132121360
4	013705842X	4	0132121360
5	013705842X		

Fig. 18.6 | Sample data from the AuthorISBN table of books.

Column	Description
ISBN	ISBN of the book (a string). The table's primary key. ISBN is an abbreviation for "International Standard Book Number"—a numbering scheme that publishers use to give every book a unique identification number.
Title	Title of the book (a string).
EditionNumber	Edition number of the book (an integer).
Copyright	Copyright year of the book (a string).

Fig. 18.7 | Titles table from the books database.

ISBN	Title	EditionNumber	Copyright
0132152134	Visual Basic 2010 How to Program	5	2011
0132151421	Visual C# 2010 How to Program	4	2011
0132575663	Java How to Program	9	2012
0132662361	C++ How to Program	8	2012
0132404168	C How to Program	6	2010
013705842X	iPhone for Programmers: An App-Driven Approach	1	2010
0132121360	Android for Programmers: An App-Driven Approach	1	2012

Fig. 18.8 | Sample books from the Titles table of the books database.

18.3 Relational Database

Overview: A books Database

- An entity-relationship (ER) diagram shows the database tables and the relationships among them
- Every row must have a primary-key value, and that value must be unique in the table
 - This is known as the Rule of Entity Integrity
- An infinity symbol (∞) indicates a one-to-many relationship, in which an entry from a table can have an arbitrary number of entries in another table
- A many-to-many relationship indicates that multiple entries can be related between tables



Fig. 18.9 | Table relationships in the books database.



Common Programming Error 18.1

Not providing a value for every column in a primary key breaks the Rule of Entity Integrity and causes the DBMS to report an error.



Common Programming Error 18.2

Providing the same primary-key value in multiple rows causes the DBMS to report an error.



Common Programming Error 18.3

Providing a foreign-key value that does not appear as a primary-key value in another table breaks the Rule of Referential Integrity and causes the DBMS to report an error.

18.4 SQL

- The next several subsections discuss the keywords listed in the following slide in the context of SQL queries and statements

SQL keyword	Description
SELECT	Retrieves data from one or more tables.
FROM	Tables involved in the query. Required in every SELECT.
WHERE	Criteria for selection that determine the rows to be retrieved, deleted or updated. Optional in a SQL query or a SQL statement.
GROUP BY	Criteria for grouping rows. Optional in a SELECT query.
ORDER BY	Criteria for ordering rows. Optional in a SELECT query.
INNER JOIN	Merge rows from multiple tables.
INSERT	Insert rows into a specified table.
UPDATE	Update rows in a specified table.
DELETE	Delete rows from a specified table.

Fig. 18.10 | SQL query keywords

18.4.1 Basic SELECT Query

- The basic form of a SELECT query is

- **SELECT * FROM tableName**

in which the asterisk (*) wildcard character indicates that all columns from the tableName should be retrieved

- **SELECT * FROM Authors**

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

- **SELECT AuthorID, LastName FROM Authors**

AuthorID	LastName
1	Deitel
2	Deitel
3	Deitel
4	Morgano
5	Kern

Fig. 18.11 | Sample AuthorID and LastName data from the Authors table.



Software Engineering Observation 18.1

In general, you process results by knowing in advance the order of the columns in the result—for example, selecting `AuthorID` and `LastName` from table `Authors` ensures that the columns will appear in the result with `AuthorID` as the first column and `LastName` as the second. Programs typically process result columns by specifying the column number in the result (starting from 1 for the first column). Selecting columns by name avoids returning unneeded columns and protects against changes to the order of the columns in the table(s) by returning the columns in the exact order specified.



Common Programming Error 18.4

If you assume that the columns are always returned in the same order from a query that uses the asterisk (*), the program may process the results incorrectly.

18.4.2 WHERE Clause

- SQL uses the optional WHERE clause in a query to specify the selection criteria for the query
- The basic form of a query with selection criteria is
 - **SELECT columnName1, columnName2, ... FROM tableName WHERE criteria**
- Strings in SQL are delimited by single (') rather than double (") quotes

- **SELECT Title, EditionNumber, Copyright FROM Titles WHERE Copyright > '2010'**

Title	EditionNumber	Copyright
Visual Basic 2010 How to Program	5	2011
Visual C# 2010 How to Program	4	2011
Java How to Program	9	2012
C++ How to Program	8	2012
Android for Programmers: An App-Driven Approach	1	2012

Fig. 18.12 | Sampling of titles with copyrights after 2005 from table `Titles`.

18.4.2 WHERE Clause (Cont.)

- The WHERE clause can contain operators <, >, <=, >=, =, <> and **LIKE**
- Operator **LIKE** is used for string pattern matching with wildcard characters percent (%) and underscore (_)
- A percent character (%) 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 (_) in the pattern string indicates a single character at that position in the pattern

- **SELECT AuthorID, FirstName, LastName FROM Authors WHERE LastName LIKE 'D%'**

AuthorID	FirstName	LastName
1	Paul	Deitel
2	Harvey	Deitel
3	Abbey	Deitel

Fig. 18.13 | Authors whose last name starts with D from the Authors table.



Portability Tip 18.1

See the documentation for your database system to determine whether SQL is case sensitive on your system and to determine the syntax for SQL keywords.



Portability Tip 18.2

Read your database system's documentation carefully to determine whether it supports the LIKE operator as discussed here.

18.4.2 WHERE Clause (Cont.)

- An underscore (`_`) in the pattern string indicates a single wildcard character at that position in the pattern
- The following query locates the rows of all the authors whose last names start with any character (specified by `_`), followed by the letter `o`, followed by any number of additional characters (specified by `%`)

- **SELECT AuthorID, FirstName, LastName
FROM Authors
WHERE LastName LIKE '_o%'**

AuthorID	FirstName	LastName
4	Michael	Morgano

Fig. 18.14 | The only author from the Authors table

18.4.3 ORDER BY Clause

- The result of a query can be sorted in ascending or descending order using the optional ORDER BY clause
- The simplest form of an ORDER BY clause is
 - **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

- **SELECT AuthorID, FirstName, LastName FROM Authors ORDER BY LastName ASC**

AuthorID	FirstName	LastName
1	Paul	Deitel
2	Harvey	Deitel
3	Abbey	Deitel
5	Eric	Kern
4	Michael	Morgano

Fig. 18.15 | Sample data from table Authors in ascending order by LastName.

- **SELECT AuthorID, FirstName, LastName FROM Authors ORDER BY LastName DESC**

AuthorID	FirstName	LastName
4	Michael	Morgano
5	Eric	Kern
1	Paul	Deitel
2	Harvey	Deitel
3	Abbey	Deitel

Fig. 18.16 | Sample data from table Authors in descending order by LastName.

18.4.3 ORDER BY Clause

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

- **SELECT AuthorID, FirstName, LastName FROM Authors ORDER BY LastName, FirstName**

AuthorID	FirstName	LastName
3	Abbey	Deitel
2	Harvey	Deitel
1	Paul	Deitel
5	Eric	Kern
4	Michael	Morgano

Fig. 18.17 | Sample data from Authors in ascending order by LastName and FirstName.

Combining the WHERE and ORDER BY Clauses

- **SELECT ISBN, Title, EditionNumber, Copyright
FROM Titles WHERE Title LIKE '%How to Program'
ORDER BY Title ASC**

ISBN	Title	Edition- Number	Copy- right
0132404168	C How to Program	6	2010
0132662361	C++ How to Program	8	2012
0132575663	Java How to Program	9	2012
0132152134	Visual Basic 2005 How to Program	5	2011
0132151421	Visual C# 2005 How to Program	4	2011

Fig. 18.18 | Sampling of books from table `Titles` whose titles end with `How to Program` in ascending order by `Title`.

18.4.4 Merging Data from Multiple Tables: INNER JOIN

- An INNER JOIN operator 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
 - **SELECT columnName1, columnName2, ... FROM table1 INNER JOIN table2 ON table1.columnName = table2.columnName**
- The ON clause of the INNER JOIN specifies the columns from each table that are compared to determine which rows are merged

18.4.4 Merging Data from Multiple Tables: INNER JOIN

- **SELECT FirstName, LastName, ISBN FROM Authors
INNER JOIN AuthorISBN ON Authors.AuthorID =
AuthorISBN.AuthorID ORDER BY LastName,
FirstName**

FirstName	LastName	ISBN	FirstName	LastName	ISBN
Abbey	Deitel	013705842X	Paul	Deitel	0132151421
Abbey	Deitel	0132121360	Paul	Deitel	0132575663
Harvey	Deitel	0132152134	Paul	Deitel	0132662361
Harvey	Deitel	0132151421	Paul	Deitel	0132404168
Harvey	Deitel	0132575663	Paul	Deitel	013705842X
Harvey	Deitel	0132662361	Paul	Deitel	0132121360
Harvey	Deitel	0132404168	Eric	Kern	013705842X
Harvey	Deitel	013705842X	Michael	Morgano	013705842X
Harvey	Deitel	0132121360	Michael	Morgano	0132121360
Paul	Deitel	0132152134			

18.4.4 Merging Data from Multiple Tables: INNER JOIN (Cont.)

- Note the use of the syntax `tableName.columnName` in the ON clause
 - This syntax, called a qualified name, specifies the columns from each table that should be compared to join the tables
- The “`tableName.`” syntax is required if the columns have the same name in both tables
- The same syntax can be used in any SQL statement to distinguish columns in different tables that have the same name
- In some systems, table names qualified with the database name can be used to perform cross-database queries
- As always, the query can contain an ORDER BY clause



Software Engineering Observation 18.2

If a SQL statement includes columns with the same name from multiple tables, the statement must precede those column names with their table names and a dot (e.g., Authors .AuthorID).



Common Programming Error 18.5

Failure to qualify names for columns that have the same name in two or more tables is an error.

18.4.5 INSERT Statement

- An INSERT statement inserts a new row into a table. The basic form of this statement is
 - **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 comma-separated list of column names in parentheses
- The list of column names is followed by the SQL keyword VALUES and a comma-separated list of values in parentheses
- Always explicitly list the columns when inserting rows

18.4.5 INSERT Statement (Cont.)

- The INSERT statement inserts a row into the Authors table
 - **INSERT INTO Authors (FirstName, LastName)
VALUES ('Sue', 'Red')**
- The statement indicates that values are provided for the FirstName and LastName columns
- The corresponding values are 'Sue' and 'Red'
- We do not specify an AuthorID in this example because AuthorID is an autoincremented column
- For every row added to this table, the DBMS assigns a unique AuthorID value that is the next value in the autoincremented sequence (i.e., 1, 2, 3 and so on)
- In this case, Sue Red would be assigned AuthorID number 6

AuthorID	FirstName	LastName
1	Paul	Deitel
2	Harvey	Deitel
3	Abbey	Deitel
4	Michael	Morgano
5	Eric	Kern
6	Sue	Red

Fig. 18.20 | Sample data from table Authors after an INSERT operation.



Common Programming Error 18.6

It's normally an error to specify a value for an autoincrement column.



Common Programming Error 18.7

SQL delimits strings with single quotes ('). A string containing a single quote (e.g., O'Malley) must have two single quotes in the position where the single quote appears (e.g., 'O' 'Malley'). The first acts as an escape character for the second. Not escaping single-quote characters in a string that's part of a SQL statement is a SQL syntax error.

18.4.6 UPDATE Statement

- An UPDATE statement modifies data in a table
 - **UPDATE tableName SET columnName1 = value1, columnName2 = value2, ..., columnNameN = valueN WHERE criteria**
- tableName is the table in which to update data
- The tableName is followed by keyword SET and a comma-separated list of column name/value pairs in the format columnName = value
- The optional WHERE clause criteria determines which rows to update

- **UPDATE Authors SET LastName = 'Black' WHERE LastName = 'Red' AND FirstName = 'Sue'**
- **UPDATE Authors SET LastName = 'Black' WHERE AuthorID = 6**

AuthorID	FirstName	LastName
1	Paul	Deitel
2	Harvey	Deitel
3	Abbey	Deitel
4	Michael	Morgano
5	Eric	Kern
6	Sue	Black

Fig. 18.21 | Sample data from table Authors after an UPDATE operation.

18.4.7 DELETE Statement

- A SQL DELETE statement removes rows from a table
 - **DELETE FROM tableName WHERE criteria**
- tableName is the table from which to delete a row (or rows)
- The optional WHERE criteria determines which rows to delete
- If this clause is omitted, all the table's rows are deleted

- **DELETE FROM Authors WHERE LastName = 'Black' AND FirstName = 'Sue'**

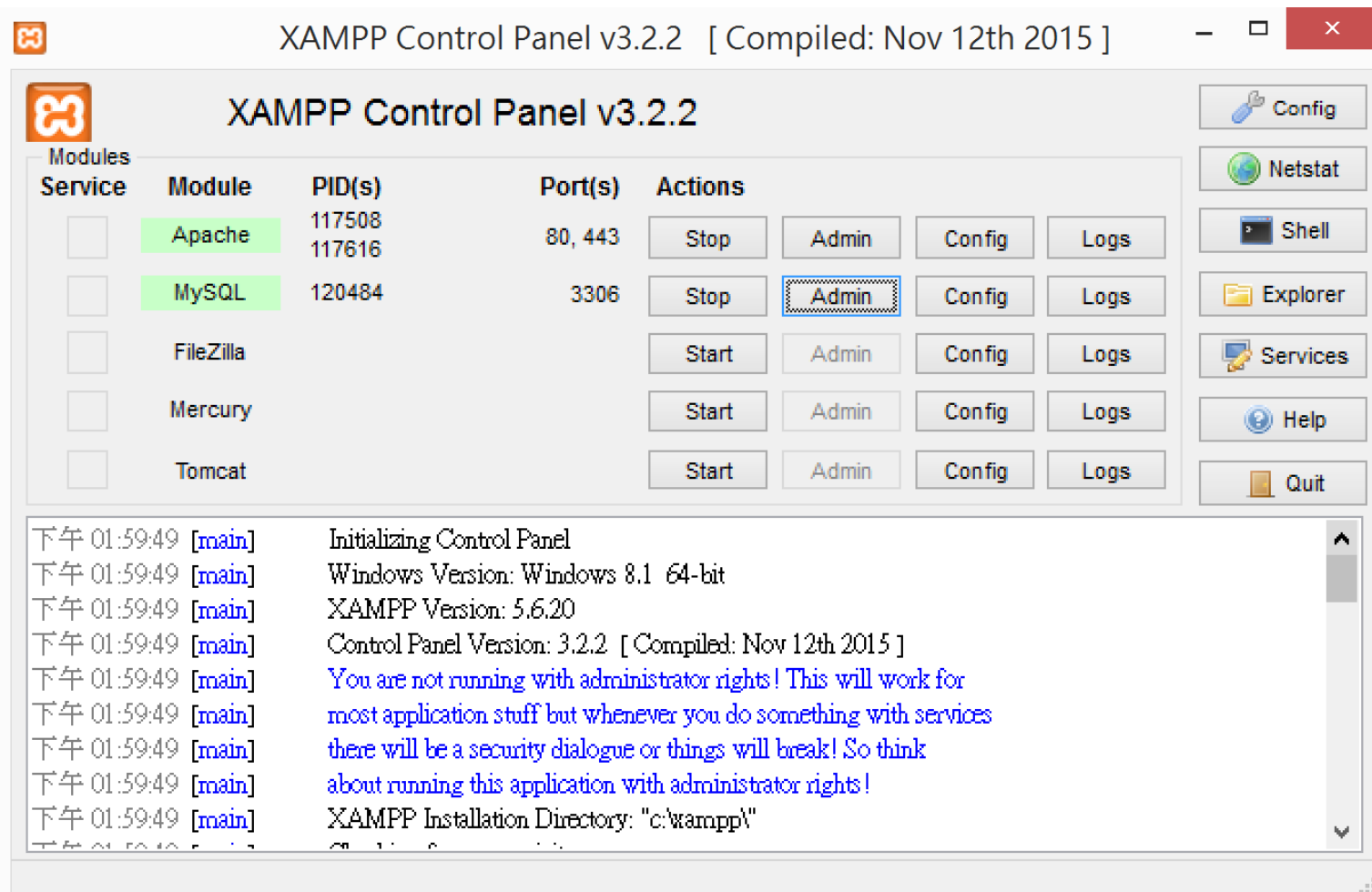
AuthorID	FirstName	LastName
1	Paul	Deitel
2	Harvey	Deitel
3	Abbey	Deitel
4	Michael	Morgano
5	Eric	Kern

Fig. 18.22 | Sample data from table Authors after a DELETE operation.

18.5 MySQL

- MySQL (pronounced “my sequel”) is a robust and scalable relational database management system (RDBMS) that was created by the Swedish consulting firm TcX in 1994
- MySQL, now owned by Oracle, is a multiuser, multithreaded RDBMS server that uses SQL to interact with and manipulate data
- Multithreading capabilities enable MySQL database to perform multiple tasks concurrently, allowing the server to process client requests efficiently
- Implementations of MySQL are available for Windows, Mac OS X, Linux and UNIX

18.5 MySQL



18.6 Java DB/Apache Derby

- The Java SE 6 and 7 Development Kits (JDKs) come bundled with the open source, pure Java database Java DB (the Oracle branded version of Apache Derby)
- Similar to MySQL, Java DB has both an embedded version and a network (client/server) version
- The tools we use in Chapters 27–28 come with Java DB
- For those examples, we use Java DB's network version, and we provide all the information you need to configure each example's database
- You can learn more about Java DB at <http://www.oracle.com/technetwork/java/javadb/overview/index.html>