Database: SQL, MySQL, DBI and ADO.NET

Outline

Database: SQL, MySQL, DBI and ADO.NET

Objectives

- In this lesson, you will learn:
  - To understand the relational database model.
  - To be able to write database queries using SQL (Structured Query Language).
  - To understand the MySQL database server.
  - To learn various database interfaces.
  - To understand ADO.NET's object model.

22.1 Introduction

- Database
  - Integrated collection of data
  - Database management system (DBMS)
    - Store and organize data consistent with database's format
    - Relational database
      - SQL (Structured Query Language)
        - Queries
        - Manipulate data

22.2 Relational Database Model

- Composed of tables
- Row
  - Number column
  - Primary key
    - Reference data in the table
    - A column or set of columns in table contains unique data

22.2 Relational Database Model

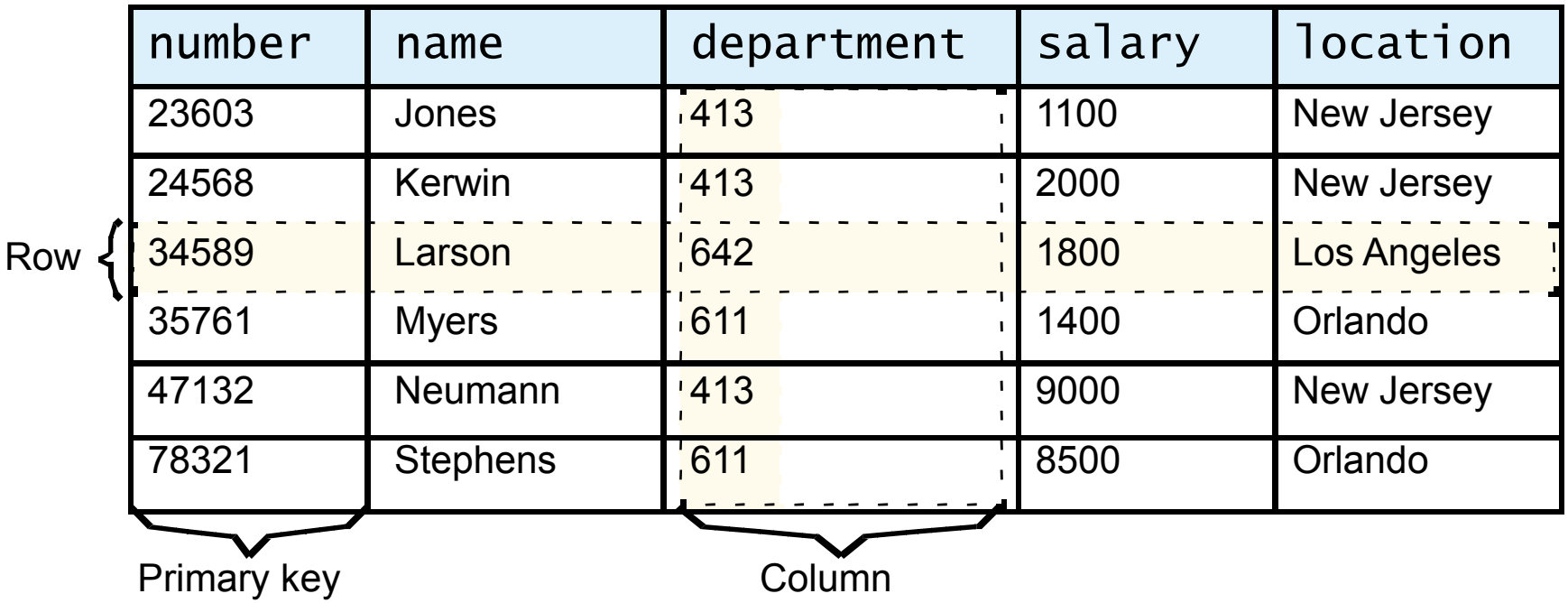

Fig. 22.1 Relational database structure of an `Employee` table.

22.2 Relational Database Model

department	location
413	New Jersey
611	Orlando
642	Los Angeles

Fig. 22.2 Table formed by selecting `department` and `location` data from the `Employee` table.

22.3 Relational Database Overview: Books.mdb Database

- Primary key uniquely identifies each row
  - Rule of Entity Integrity
- Composite primary key
- Lines connecting tables
  - Relationships
    - One-to-many relationship
- Foreign key
  - Join multiple tables
  - Rule of Referential Integrity

22.3 Relational Database Overview: Books.mdb Database

Field	Description
authorID	Author's ID number in the database. In the Books.mdb database, this Integer column is defined as auto-increment. For each new row inserted in this table, the database increments the authorID value, ensuring 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. 22.3 Authors table from Books.mdb.

22.3 Relational Database Overview: Books.mdb Database

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Kate	Steinbuhler
5	Sean	Santry
6	Ted	Lin
7	Praveen	Sadhu
8	David	McPhie
9	Cheryl	Yaeger
10	Marina	Zlatkina
11	Ben	Wiedermann
12	Jonathan	Liperi

Fig. 22.4 Data from the Authors table of Books.mdb.

22.3 Relational Database Overview: Books.mdb Database

Field	Description
publisherID	The publisher's ID number in the database. This auto-incremented Integer is the table's primary key.
publisherName	The name of the publisher (a String).

Fig. 22.5 Publishers table from Books.mdb.

publisherID	publisherName
1	Prentice Hall
2	Prentice Hall PTG

Fig. 22.6 Data from the Publishers table of Books.mdb.

Field	Description
authorID	The author's ID number, which allows the database to associate each book with a specific author. The integer ID number in this column must also appear in the Authors table.
isbn	The ISBN number for a book (a String).

Fig. 22.7 AuthorISBN table from Books.mdb.

22.3 Relational Database Overview: Books.mdb Database

authorID	isbn	authorID	isbn
1	0130895725	2	0139163050
1	0132261197	2	013028419x
1	0130895717	2	0130161438
1	0135289106	2	0130856118
1	0139163050	2	0130125075
1	013028419x	2	0138993947
1	0130161438	2	0130852473
1	0130856118	2	0130829277
1	0130125075	2	0134569555
1	0138993947	2	0130829293
1	0130852473	2	0130284173
1	0130829277	2	0130284181
1	0134569555	2	0130895601
1	0130829293	3	013028419x
1	0130284173	3	0130161438
1	0130284181	3	0130856118
1	0130895601	3	0134569555
2	0130895725	3	0130829293
2	0132261197	3	0130284173
2	0130895717	3	0130284181
2	0135289106	4	0130895601

Fig. 22.8 Data from AuthorISBN table in Books.mdb.

22.3 Relational Database Overview: Books.mdb Database

Field	Description
isbn	ISBN number of the book (a String).
title	Title of the book (a String).
editionNumber	Edition number of the book (a String).
copyright	Copyright year of the book (an Integer).
description	Description of the book (a String).
publisherID	Publisher's ID number (an Integer). This value must correspond to an ID number in the Publishers table.
imageFile	Name of the file containing the book's cover image (a String).
price	Suggested retail price of the book (a real number). [*Note:* The prices shown in this database are for example purposes only.]

Fig. 22.9 Titles table from Books.mdb.

22.3 Relational Database Overview: Books.mdb Database

isbn	title	edition-Number	publisherID	copy-right	price
0130923613	Python How to Program	1	1	2002	$69.95
0130622214	C# How to Program	1	1	2002	$69.95
0130341517	Java How to Program	4	1	2002	$69.95
0130649341	The Complete Java Training Course	4	2	2002	$109.95
0130895601	Advanced Java 2 Platform How to Program	1	1	2002	$69.95
0130308978	Internet and World Wide Web How to Program	2	1	2002	$69.95
0130293636	Visual Basic .NET How to Program	2	1	2002	$69.95
0130895636	The Complete C++ Training Course	3	2	2001	$109.95
0130895512	The Complete e-Business & e-Commerce Programming Training Course	1	2	2001	$109.95

Fig. 22.10 Portion of the data from the Titles table of Books.mdb.

22.3 Relational Database Overview: Books.mdb Database

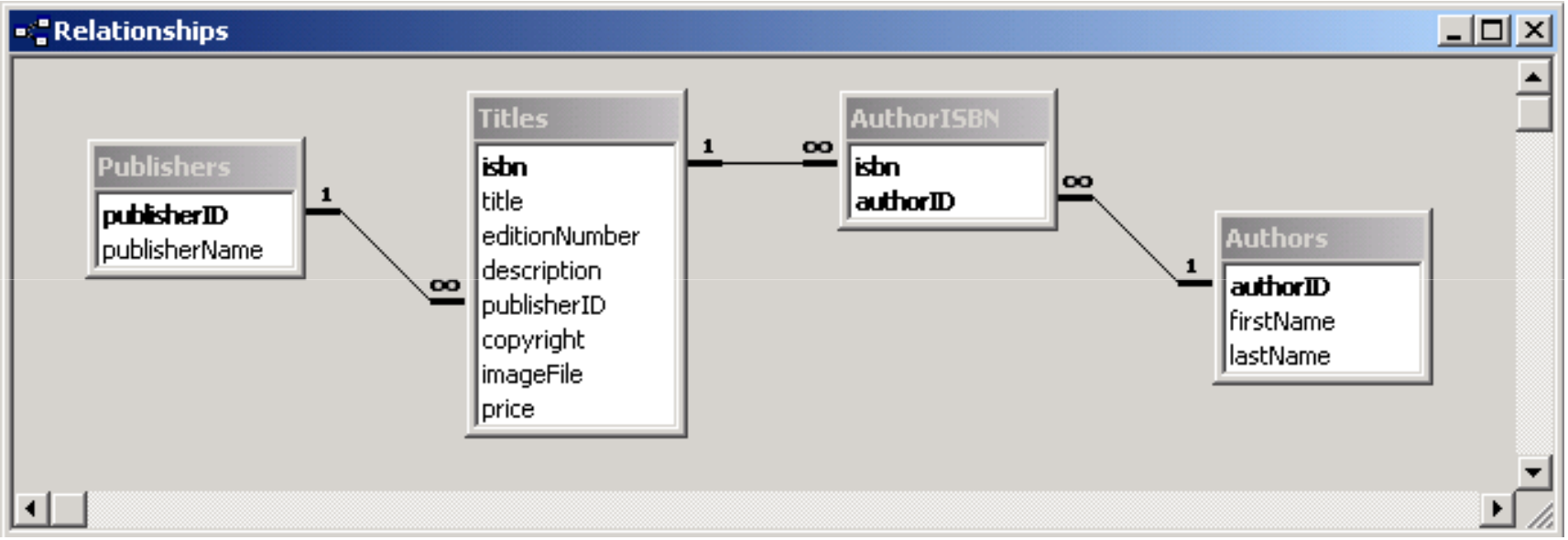

Fig. 22.11 Table relationships in Books.mdb.

22.4 SQL (Structured Query Language)

SQL keyword	Description
`SELECT`	Selects (retrieves) columns from one or more tables.
`FROM`	Specifies tables from which to get columns or delete rows. Required in every `SELECT` and `DELETE` statement.
`WHERE`	Specifies criteria that determine the rows to be retrieved.
`INNER JOIN`	Joins rows from multiple tables to produce a single set of rows.
`GROUP BY`	Specifies criteria for grouping rows.
`ORDER BY`	Specifies criteria for ordering rows.
`INSERT`	Inserts data into a specified table.
`UPDATE`	Updates data in a specified table.
`DELETE`	Deletes data from a specified table.
`CREATE`	Creates a new table.
`DROP`	Deletes an existing table.
`COUNT`	Returns the number of records that satisfy given search criteria.

Fig. 22.12 SQL keywords.

22.4.1 Basic SELECT Query

- SELECT * FROM *tableName*
  - SELECT * FROM Authors
  - SELECT authorID, lastName FROM Authors

22.4.1 Basic SELECT Query

authorID	lastName	authorID	lastName
1	Deitel	7	Sadhu
2	Deitel	8	McPhie
3	Nieto	9	Yaeger
4	Steinbuhler	10	Zlatkina
5	Santry	11	Wiedermann
6	Lin	12	Liperi

Fig. 22.13 authorID and lastName from the Authors table.

22.4.2 WHERE Clause

- Specify selection criteria for query
  - SELECT *columnName1*, *columnName2*, … FROM *tableName* WHERE *criteria*
    - `SELECT title, editionNumber, copyright FROM Titles WHERE copyright > 1999`
  - LIKE
    - Pattern matching
      - Asterisk (*)
        - `SELECT authorID, firstName, lastName FROM Authors WHERE lastName LIKE 'D*'`
      - Question mark (?)
        - `SELECT authorID, firstName, lastName FROM Authors WHERE lastName LIKE '?I*'`

22.4.2 WHERE Clause

Title	editionNumber	copyright
Internet and World Wide Web How to Program	2	2002
Java How to Program	4	2002
The Complete Java Training Course	4	2002
The Complete e-Business & e-Commerce Programming Training Course	1	2001
The Complete Internet & World Wide Web Programming Training Course	2	2001
The Complete Perl Training Course	1	2001
The Complete XML Programming Training Course	1	2001
C How to Program	3	2001
C++ How to Program	3	2001
The Complete C++ Training Course	3	2001
e-Business and e-Commerce How to Program	1	2001
Internet and World Wide Web How to Program	1	2000
The Complete Internet and World Wide Web Programming Training Course	1	2000

22.4.2 WHERE Clause

Java How to Program (Java 2)	3	2000
The Complete Java 2 Training Course	3	2000
XML How to Program	1	2001
Perl How to Program	1	2001
Advanced Java 2 Platform How to Program	1	2002
e-Business and e-Commerce for Managers	1	2000
Wireless Internet and Mobile Business How to Program	1	2001
C# How To Program	1	2002
Python How to Program	1	2002
Visual Basic .NET How to Program	2	2002

Fig. 22.14 Titles with copyrights after 1999 from table `Titles`.

22.4.2 WHERE Clause

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel

Fig. 22.15 Authors from the Authors table whose last names start with D.

authorID	firstName	lastName
3	Tem	Nieto
6	Ted	Lin
11	Ben	Wiedermann
12	Jonathan	Liperi

Fig. 22.16 Authors from table Authors whose last names contain i as their second letter.

22.4.3 ORDER BY Clause

- Arranged in ascending or descending order
  - SELECT *columnName1*, *columnName2*, … FROM *tableName* ORDER BY *column* ASC
    - `SELECT authorID, firstName, lastName FROM Authors ORDER BY lastName ASC`
  - SELECT *columnName1*, *columnName2*, … FROM *tableName* ORDER BY *column* DESC
    - `SELECT authorID, firstName, lastName FROM Authors ORDER BY lastName DESC`

22.4.3 ORDER BY Clause

authorID	firstName	lastName
2	Paul	Deitel
1	Harvey	Deitel
6	Ted	Lin
12	Jonathan	Liperi
8	David	McPhie
3	Tem	Nieto
7	Praveen	Sadhu
5	Sean	Santry
4	Kate	Steinbuhler
11	Ben	Wiedermann
9	Cheryl	Yaeger
10	Marina	Zlatkina

Fig. 22.17 Authors from table `Authors` in ascending order by `lastName`.

22.4.3 ORDER BY Clause

authorID	firstName	lastName
10	Marina	Zlatkina
9	Cheryl	Yaeger
11	Ben	Wiedermann
4	Kate	Steinbuhler
5	Sean	Santry
7	Praveen	Sadhu
3	Tem	Nieto
8	David	McPhie
12	Jonathan	Liperi
6	Ted	Lin
2	Paul	Deitel
1	Harvey	Deitel

Fig. 22.18 Authors from table `Authors` in descending order by `lastName`.

22.4.3 ORDER BY Clause

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
6	Ted	Lin
12	Jonathan	Liperi
8	David	McPhie
3	Tem	Nieto
7	Praveen	Sadhu
5	Sean	Santry
4	Kate	Steinbuhler
11	Ben	Wiedermann
9	Cheryl	Yaeger
10	Marina	Zlatkina

Fig. 22.19 Authors from table `Authors` in ascending order by `lastName` and by `firstName`.

22.4.3 ORDER BY Clause

isbn	title	edition-Number	copy-right	price
0130895601	Advanced Java 2 Platform How to Program	1	2002	$69.95
0131180436	C How to Program	1	1992	$69.95
0130895725	C How to Program	3	2001	$69.95
0132261197	C How to Program	2	1994	$49.95
0130622214	C# How To Program	1	2002	$69.95
0135289106	C++ How to Program	2	1998	$49.95
0131173340	C++ How to Program	1	1994	$69.95
0130895717	C++ How to Program	3	2001	$69.95
013028419X	e-Business and e-Commerce How to Program	1	2001	$69.95
0130308978	Internet and World Wide Web How to Program	2	2002	$69.95
0130161438	Internet and World Wide Web How to Program	1	2000	$69.95

22.4.3 ORDER BY Clause

0130341517	Java How to Program	4	2002	$69.95
0136325890	Java How to Program	1	1998	$69.95
0130284181	Perl How to Program	1	2001	$69.95
0130923613	Python How to Program	1	2002	$69.95
0130293636	Visual Basic .NET How to Program	2	2002	$69.95
0134569555	Visual Basic 6 How to Program	1	1999	$69.95
0130622265	Wireless Internet and Mobile Business How to Program	1	2001	$69.95
0130284173	XML How to Program	1	2001	$69.95

Fig. 22.20 Books from table `Titles` whose titles end with `How to Program` in ascending order by `title`.

22.4.4 Merging Data from Multiple Tables: INNER JOIN

- Normalize databases
  - Ensure database does not store data redundantly
  - SELECT *columnName1*, *columnName2*, … FROM *table1* INNER JOIN *table2* ON *table1*, *columnName* = *table2.columnName*

22.4.4 Merging Data from Multiple Tables: INNER JOIN

firstName	lastName	isbn	firstName	lastName	isbn
Harvey	Deitel	0130895601	Harvey	Deitel	0130856118
Harvey	Deitel	0130284181	Harvey	Deitel	0130161438
Harvey	Deitel	0130284173	Harvey	Deitel	013028419x
Harvey	Deitel	0130829293	Harvey	Deitel	0139163050
Harvey	Deitel	0134569555	Harvey	Deitel	0135289106
Harvey	Deitel	0130829277	Harvey	Deitel	0130895717
Harvey	Deitel	0130852473	Harvey	Deitel	0132261197
Harvey	Deitel	0138993947	Harvey	Deitel	0130895725
Harvey	Deitel	0130125075	Paul	Deitel	0130895601
Paul	Deitel	0130284181	Paul	Deitel	0135289106
Paul	Deitel	0130284173	Paul	Deitel	0130895717
Paul	Deitel	0130829293	Paul	Deitel	0132261197
Paul	Deitel	0134569555	Paul	Deitel	0130895725
Paul	Deitel	0130829277	Tem	Nieto	0130284181
Paul	Deitel	0130852473	Tem	Nieto	0130284173
Paul	Deitel	0138993947	Tem	Nieto	0130829293
Paul	Deitel	0130125075	Tem	Nieto	0134569555
Paul	Deitel	0130856118	Tem	Nieto	0130856118
Paul	Deitel	0130161438	Tem	Nieto	0130161438
Paul	Deitel	013028419x	Tem	Nieto	013028419x
Paul	Deitel	0139163050	Sean	Santry	0130895601

Fig. 22.21 Authors from table `Authors` and ISBN numbers of the authors' books, sorted in ascending order by `lastName` and `firstName`.

```
1  SELECT Titles.title, Titles.isbn, Authors.firstName,
2         Authors.lastName, Titles.copyright,
3         Publishers.publisherName
4  FROM
5     ( Publishers INNER JOIN Titles
6        ON Publishers.publisherID = Titles.publisherID )
7     INNER JOIN
8     ( Authors INNER JOIN AuthorISBN
9        ON Authors.authorID = AuthorISBN.authorID )
10     ON Titles.isbn = AuthorISBN.isbn
11 ORDER BY Titles.title
```

Fig. 22.22
(1 of 1)

22.4.5 Joining Data from Tables Authors, AuthorISBN, Titles and Publishers

Title	isbn	first-Name	last-Name	copy-right	publisher-Name
Advanced Java 2 Platform How to Program	0130895601	Paul	Deitel	2002	Prentice Hall
Advanced Java 2 Platform How to Program	0130895601	Harvey	Deitel	2002	Prentice Hall
Advanced Java 2 Platform How to Program	0130895601	Sean	Santry	2002	Prentice Hall
C How to Program	0131180436	Harvey	Deitel	1992	Prentice Hall
C How to Program	0131180436	Paul	Deitel	1992	Prentice Hall
C How to Program	0132261197	Harvey	Deitel	1994	Prentice Hall
C How to Program	0132261197	Paul	Deitel	1994	Prentice Hall
C How to Program	0130895725	Harvey	Deitel	2001	Prentice Hall
C How to Program	0130895725	Paul	Deitel	2001	Prentice Hall
C# How To Program	0130622214	Tem	Nieto	2002	Prentice Hall
C# How To Program	0130622214	Paul	Deitel	2002	Prentice Hall
C# How To Program	0130622214	Cheryl	Yaeger	2002	Prentice Hall
C# How To Program	0130622214	Marina	Zlatkina	2002	Prentice Hall
C# How To Program	0130622214	Harvey	Deitel	2002	Prentice Hall
C++ How to Program	0130895717	Paul	Deitel	2001	Prentice Hall
C++ How to Program	0130895717	Harvey	Deitel	2001	Prentice Hall

22.4.5 Joining Data from Tables Authors, AuthorISBN, Titles and Publishers

C++ How to Program	0131173340	Paul	Deitel	1994	Prentice Hall
C++ How to Program	0131173340	Harvey	Deitel	1994	Prentice Hall
C++ How to Program	0135289106	Harvey	Deitel	1998	Prentice Hall
C++ How to Program	0135289106	Paul	Deitel	1998	Prentice Hall
e-Business and e-Commerce for Managers	0130323640	Harvey	Deitel	2000	Prentice Hall
e-Business and e-Commerce for Managers	0130323640	Kate	Steinbuhler	2000	Prentice Hall
e-Business and e-Commerce for Managers	0130323640	Paul	Deitel	2000	Prentice Hall
e-Business and e-Commerce How to Program	013028419X	Harvey	Deitel	2001	Prentice Hall
e-Business and e-Commerce How to Program	013028419X	Paul	Deitel	2001	Prentice Hall
e-Business and e-Commerce How to Program	013028419X	Tem	Nieto	2001	Prentice Hall

Fig. 22.23 Portion of the result set produced by the query in Fig. 22.22.

22.4.6 INSERT Statement

- Inserts new row in table
  - INSERT INTO *tableName* (*columnName1*, *columnName2*, …, *columnNameN*) VALUES (*value1* , *value2*, …, *valueN*)

22.4.6 INSERT Statement

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Kate	Steinbuhler
5	Sean	Santry
6	Ted	Lin
7	Praveen	Sadhu
8	David	McPhie
9	Cheryl	Yaeger
10	Marina	Zlatkina
11	Ben	Wiedermann
12	Jonathan	Liperi
13	Sue	Smith

Fig. 22.24 Table Authors after an INSERT operation to add a row.

22.4.7 UPDATE Statement

- Modifies data in a table
  - UPDATE *tableName* SET *columnName1* = *value1*, *columnName2* = *value2*, …, *columnNameN* = *valueN* WHERE *criteria*

22.4.7 UPDATE Statement

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Kate	Steinbuhler
5	Sean	Santry
6	Ted	Lin
7	Praveen	Sadhu
8	David	McPhie
9	Cheryl	Yaeger
10	Marina	Zlatkina
11	Ben	Wiedermann
12	Jonathan	Liperi
13	Sue	Jones

Fig. 22.25 Table `Authors` after an `UPDATE` operation to change a row.

22.4.8 DELETE Statement

- Removes data from a table
  - DELETE FROM *tableName* WHERE *criteria*

22.4.8 DELETE Statement

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Kate	Steinbuhler
5	Sean	Santry
6	Ted	Lin
7	Praveen	Sadhu
8	David	McPhie
9	Cheryl	Yaeger
10	Marina	Zlatkina
11	Ben	Wiedermann
12	Jonathan	Liperi

Fig. 22.26 Table `Authors` after a `DELETE` operation to remove a row.

22.5 MySQL

- Multi-user and multi-threaded RDBMS server
- Uses SQL to interact with and manipulate data
- Supports various programming languages
- Access tables from different databases
- Handle large databases

22.6 Introduction to DBI

- Uniform access to all database systems
- Access relational databases

22.6.1 Perl Database Interface

- Access relational databases from Perl programs
- Database independent
- Handles
  - Driver handles
  - Database handles
  - Statement handles

22.6.1 Perl Database Interface

Data object handle	Description
Driver handle	Encapsulates the driver for the database; rarely used in a Perl script.
Database handle	Encapsulates a specific connection to a database; can send SQL statements to a database.
Statement handle	Encapsulates specific SQL statements and the results returned from them.

Fig. 22.27 Data object handles for Perl DBI.

22.6.2 PHP dbx module

- Seven functions that interface to database modules
- MySQL
- PostgreSQL
- Microsoft SQL Server
- Oracle
- Sybase
- FrontBase
- ODBC (Open Database Connectivity)

22.6.2 PHP dbx module

dbx function	Description
`dbx_connect`	Opens a connection/database.
`dbx_close`	Closes an open connection/database.
`dbx_error`	Reports any error messages from the last function call in the module.
`dbx_query`	Executes a query and returns the results.
`dbx_sort`	Sorts a result by a custom sort function.
`dbx_compare`	Compares two rows and sorts them.
`dbx_escape_string`	Escapes a string for use in an SQL query.

Fig. 22.28 Data objects for PHP `dbx` modules.

22.6.3 Python DB-API

- Consists of `Connection` data objects and `Cursor` data objects
- Portable across several databases

22.6.3 Python DB-API

Connection data object	Description
`close`	Closes the connection to the database.
`commit`	Commits (saves) a transaction (i.e., database update operation).
`rollback`	Exits a pending transaction without saving changes.
`cursor`	Returns a new `Cursor` object or the current connection.

Fig. 22.29 `Connection` data objects for Python DB-API.

Cursor data object	Description
`rowcount`	Returns the number of rows affected by the last execute method call.
`close`	Closes the `Cursor` object.
`execute(` *operation* `)`	Executes a database query or statement. Return values not defined.
`executemany(` *operation*`,` *parameters* `)`	Executes a database query or statement against a set of parameters. Return values not defined.
`fetchone`	Returns the next row of a query result.
`fetchmany(` *size* `)`	Returns a set of rows—defined in the parameter—from a query result.
`fetchall`	Returns all the rows of a query result.

Fig. 22.30 Some `Cursor` data objects for Python-API.

22.7 ADO.NET Object Model

- API
  - Access database systems programmatically
  - Created for the .NET framework
  - Primary namespaces
    - `System.Data`
    - `System.Data.OleDb`
    - `System.Data.SqlClient`

22.8 Web Resources

- www.sql.org
- www.mysql.com
- www.microsoft.com/sql
- www.microsoft.com/sql/downloads/default.asp
- www.postgresql.org
- www.interbase.com
- www.maverick-dbms.org
- www.devshed.com
- www.cql.com
- leap.sourceforge.net
- www.voicenet.com/~gray/Home.html
- msdn.microsoft.com/library/devprods/vs6/vstudio/mdac200/mdac3sc7.htm
- www.w3schools.com/sql
- www.sqlmag.com