22

Database: SQL

OBJECTIVES

In this chapter you will learn:

- Relational database concepts.
- To use Structured Query Language (SQL) to retrieve data from and manipulate data in a database.

- Introduction
- Relational Databases
- Relational Database Overview: The `books` Database
- SQL
  - Basic `SELECT` Query
  - `WHERE` Clause
  - `ORDER BY` Clause
  - Combining Data from Multiple Tables: `INNER JOIN`
  - `INSERT` Statement
  - `UPDATE` Statement
  - `DELETE` Statement

Introduction

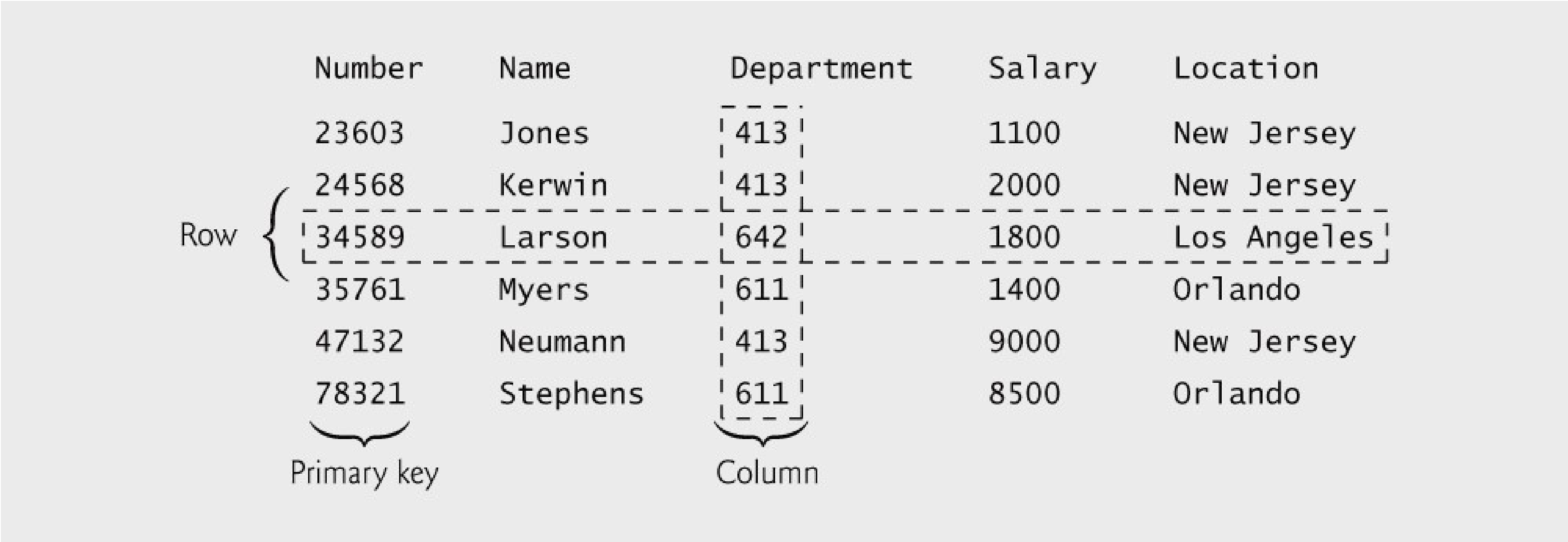
- A database is an integrated collection of data. A database management system (DBMS) provides mechanisms for storing, organizing, retrieving and modifying data.
- SQL is the international standard language used almost universally with relational database systems to perform queries and manipulate data.
- Programs connect to, and interact with, relational databases systems via an interface—software that facilitates communications between a database management system and a program.

Relational Data

- 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 is 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 can be composed of more than one column.
- SQL provides a rich set of language constructs that enable you to define complex queries to retrieve data from a database.

Relational Data (Cont.)

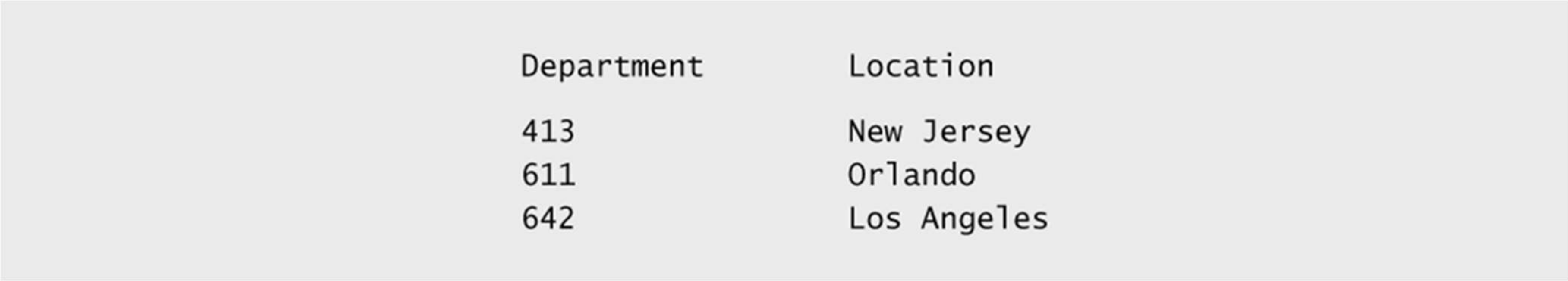
- Every column in a primary key must have a value, and the value of the primary key must be unique. This is known as the Rule of Entity Integrity.
- A one-to-many relationship between tables indicates that a row in one table can have many related rows in a separate table.
- A foreign key is a column in a table that matches the primary-key column in another table.
- 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 can be used to combine information from multiple tables. There is a one-to-many relationship between a primary key and its corresponding foreign key.



Number	Name	Department	Salary	Location
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

Row
Primary key
Column

Employee table sample data.

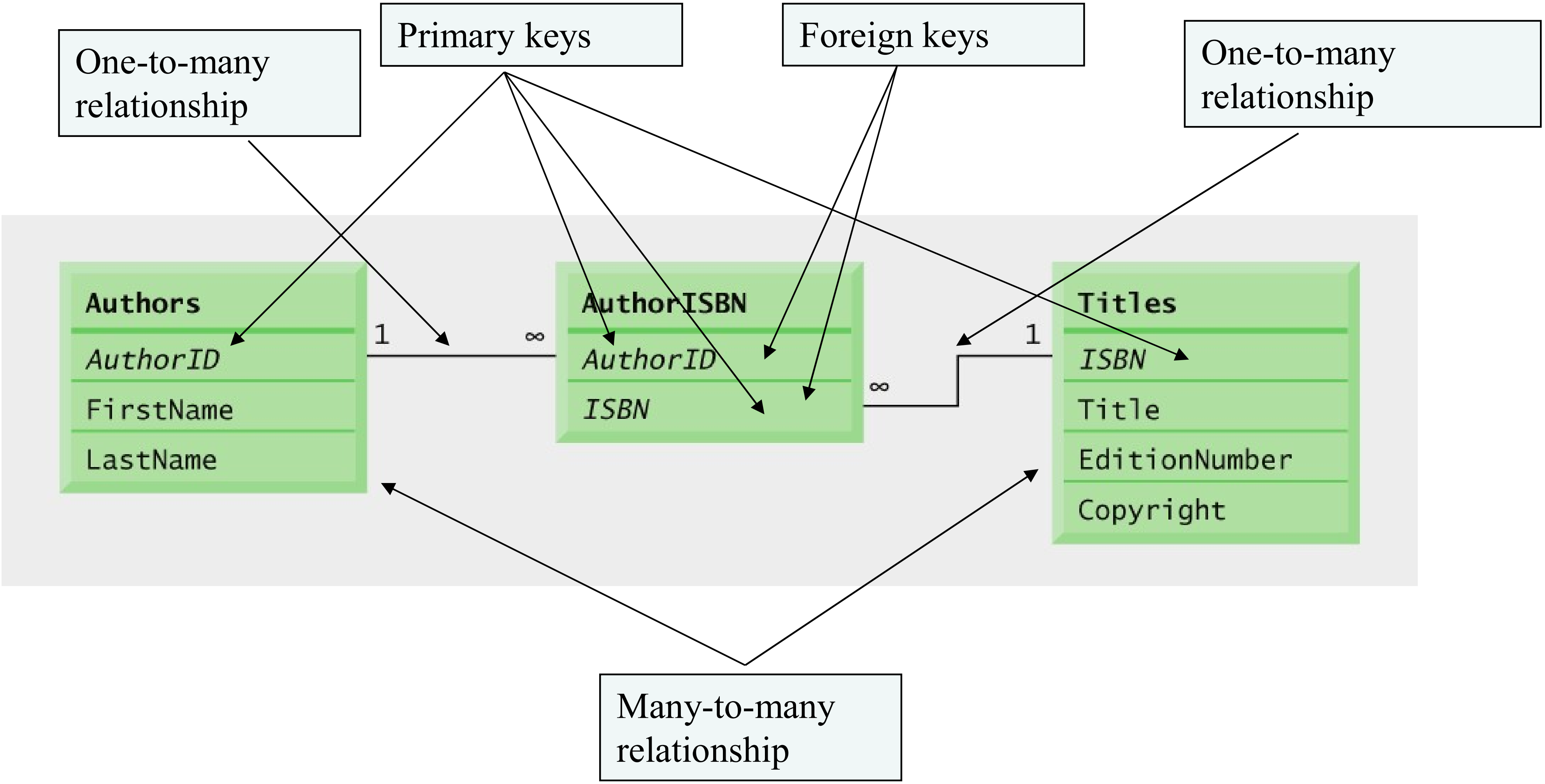


Department	Location
413	New Jersey
611	Orlando
642	Los Angeles

Result of selecting distinct `Department` and `Location` data from table `Employee`.

Relational Database Overview: A books Database

- 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.



One-to-many relationship

Primary keys

Foreign keys

One-to-many relationship

Many-to-many relationship

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).

authors table from the books database.

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Andrew	Goldberg
4	David	Choffnes

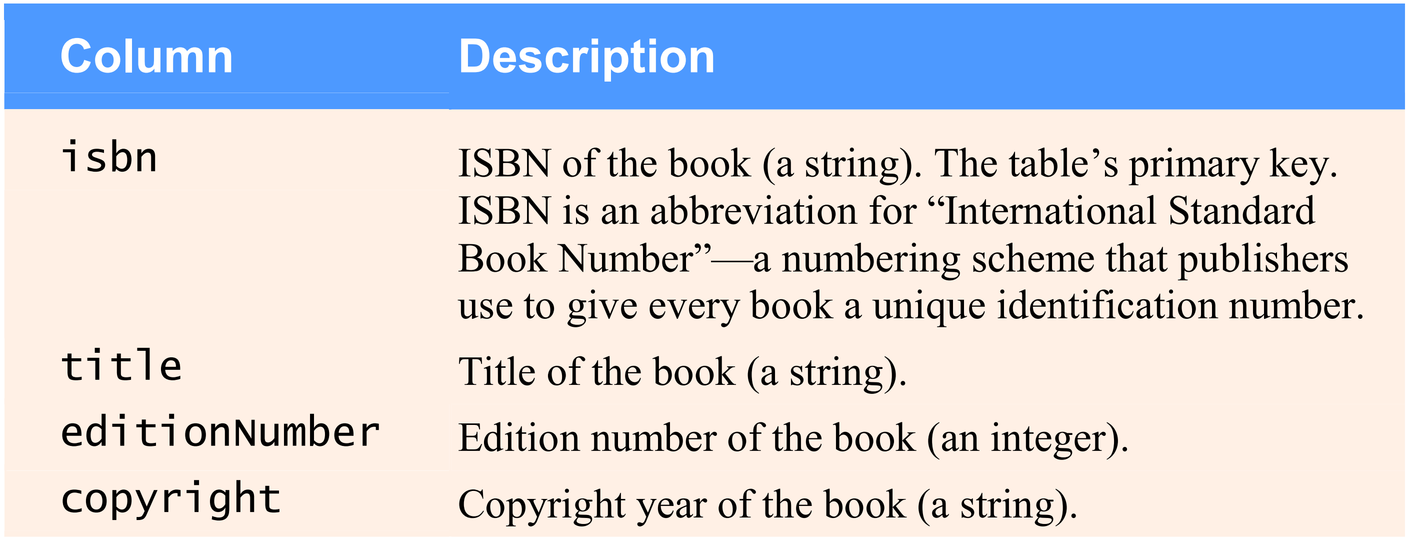
Sample data from the authors table.

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.

`authorISBN` table from the `books` database.

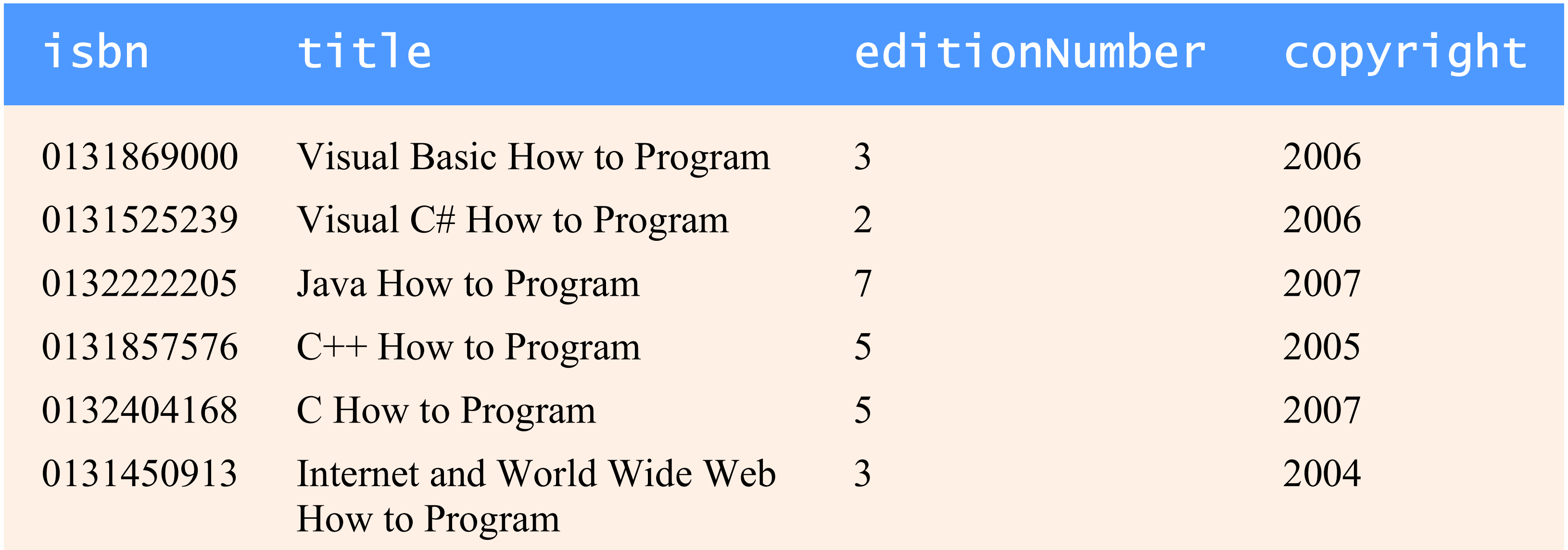
`authorID`	`isbn`	`authorID`	`isbn`
1	0131869000	2	0131450913
2	0131869000	1	0131828274
1	0131483986	2	0131828274
2	0131483986	3	0131450913
1	0131450913	4	0131828274

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).

`titles` table from the `books` database.



isbn	title	editionNumber	copyright
0131869000	Visual Basic How to Program	3	2006
0131525239	Visual C# How to Program	2	2006
0132222205	Java How to Program	7	2007
0131857576	C++ How to Program	5	2005
0132404168	C How to Program	5	2007
0131450913	Internet and World Wide Web How to Program	3	2004

Sample data from the `titles` table of the `books` database.

Relational Database Overview: A books Database (Cont.)

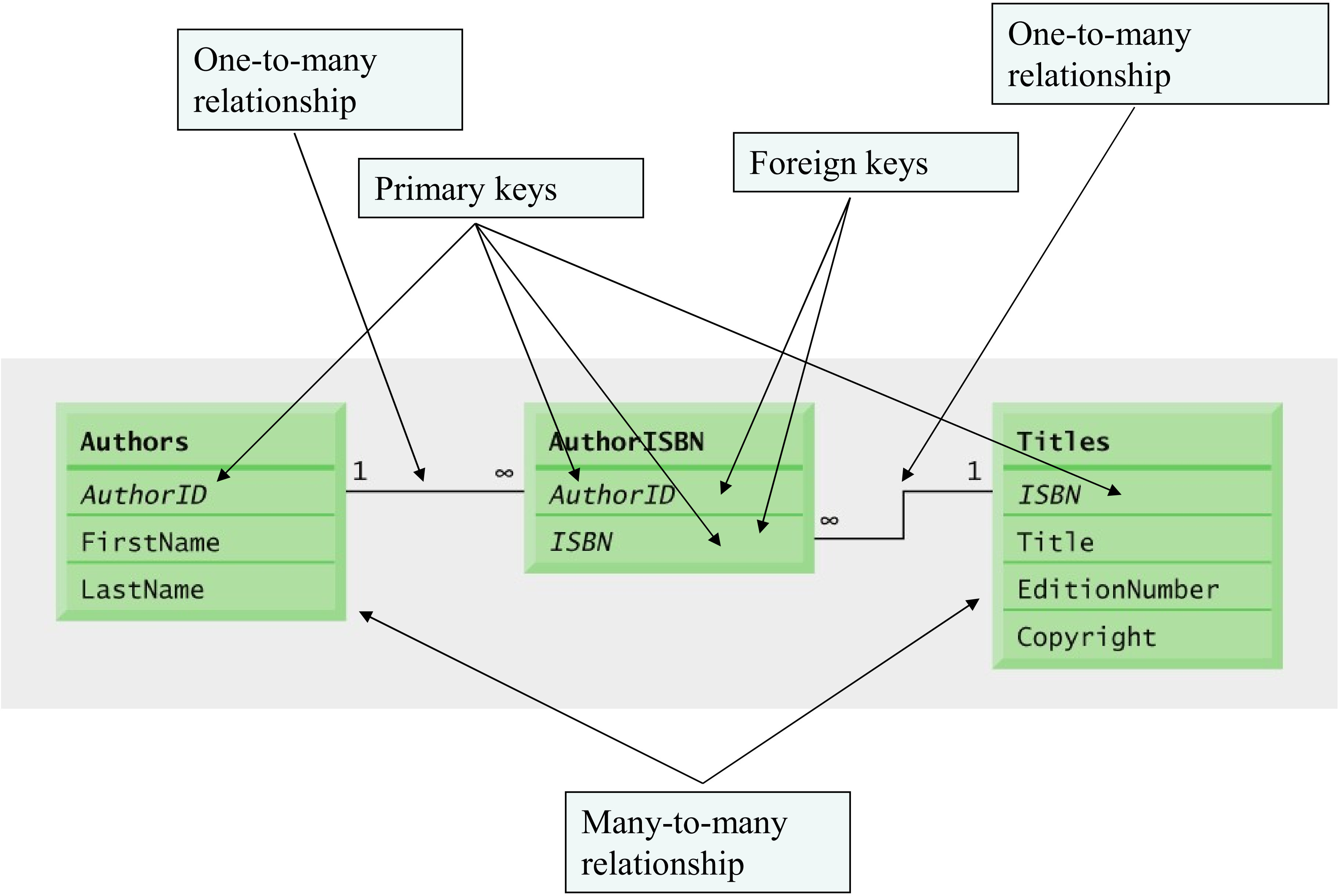
- An entity-relationship (ER) diagram shows the database tables and the relationships among them.
- Every row must have a primary-key value, and that value must be unique in the table. This is known as the Rule of Entity Integrity.
- An infinity symbol (∞) indicates a one-to-many relationship, in which 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.

Common Programming Error

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

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



One-to-many relationship

Primary keys

Foreign keys

One-to-many relationship

Authors

AuthorID

FirstName

LastName

1 ∞

AuthorISBN

AuthorID

ISBN

∞ 1

Titles

ISBN

Title

EditionNumber

Copyright

Many-to-many relationship

Table relationships in the books database.

Common Programming Error

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.

SQL

- The next several sections will discuss most of 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	Combine 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.

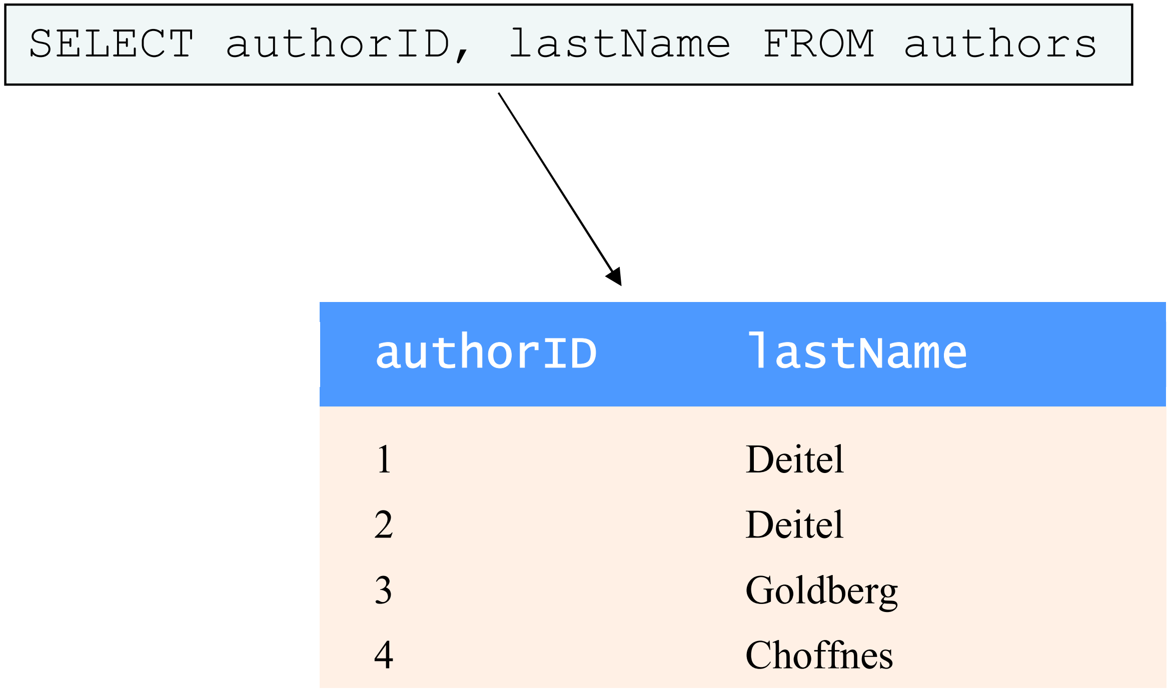
SQL query keywords.

Basic `SELECT` Query

- The basic form of a query is

  ```
  SELECT * FROM tableName
  ```

  where the asterisk (*) indicates that all columns from *`tableName`* should be selected, and *`tableName`* specifies the table in the database from which rows will be retrieved.
- To retrieve specific columns from a table, replace the asterisk (*) with a comma-separated list of column names.



```
SELECT authorID, lastName FROM authors
```

authorID	lastName
1	Deitel
2	Deitel
3	Goldberg
4	Choffnes

Sample `authorID` and `lastName` data from the `authors` table.

WHERE Clause

- The optional `WHERE` clause in a query specifies the selection criteria for the query. The basic form of a query with selection criteria is

  `SELECT` *`columnName1`*`,` *`columnName2`*`, … FROM`
  `tableName` `WHERE` *`criteria`*

- 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.

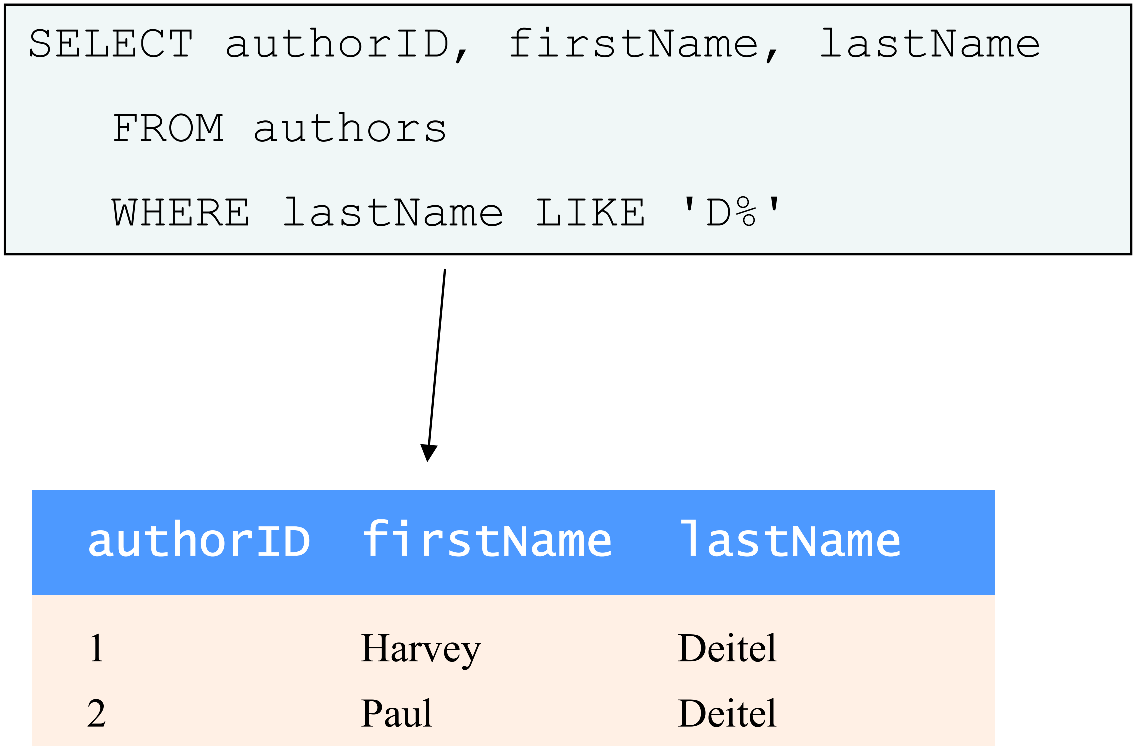
```
SELECT title, editionNumber,
copyright

   FROM titles

   WHERE copyright > '2005'
```

title	editionNumber	copyright
Visual C# How to Program	2	2006
Visual Basic 2005 How to Program	3	2006
Java How to Program	7	2007
C How to Program	5	2007

Sampling of titles with copyrights after 2005 from table `titles`.



```
SELECT authorID, firstName, lastName
   FROM authors
   WHERE lastName LIKE 'D%'
```

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel

Authors whose last name starts with `D` from the `authors` table.

```
SELECT authorID, firstName, lastName
   FROM authors
   WHERE lastName LIKE '_o%'
```

authorID	firstName	lastName
3	Andrew	Goldberg

The only author from the authors table whose last name contains o as the second letter.

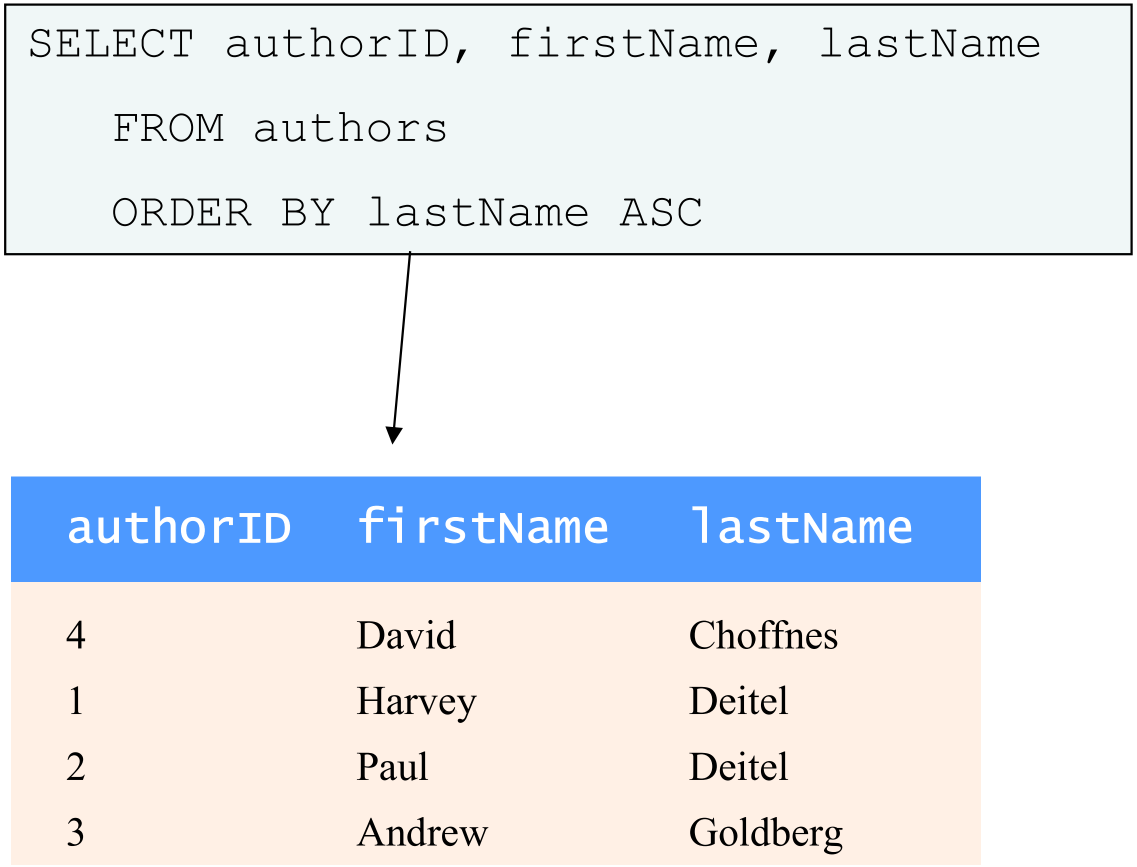
ORDER BY Clause

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

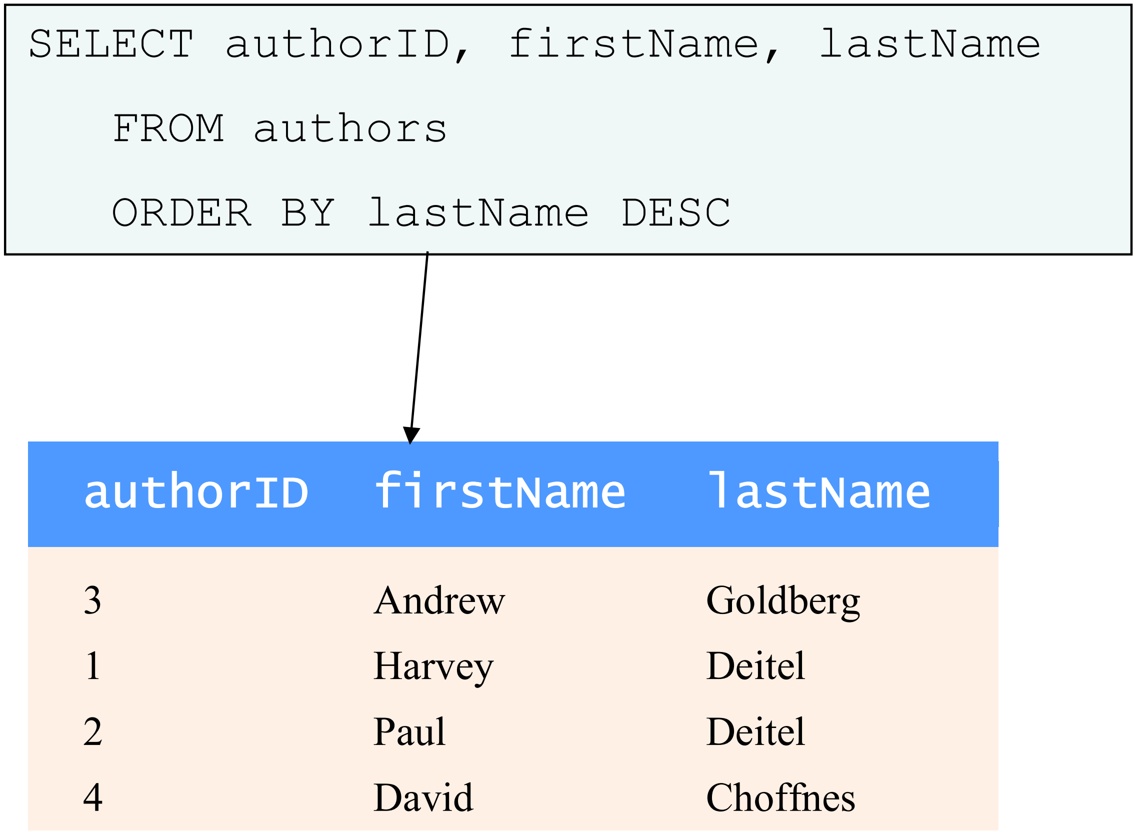
  ```
  ORDER BY column1 sortingOrder, column2
  sortingOrder, …
  ```
- The `WHERE` and `ORDER BY` clauses can be combined in one query. If used, `ORDER BY` must be the last clause in the query.



```
SELECT authorID, firstName, lastName
   FROM authors
   ORDER BY lastName ASC
```

authorID	firstName	lastName
4	David	Choffnes
1	Harvey	Deitel
2	Paul	Deitel
3	Andrew	Goldberg

`authors` sample data in ascending order by `lastName`.



```
SELECT authorID, firstName, lastName
   FROM authors
   ORDER BY lastName DESC
```

authorID	firstName	lastName
3	Andrew	Goldberg
1	Harvey	Deitel
2	Paul	Deitel
4	David	Choffnes

`authors` sample data in descending order by `lastName`.

```
SELECT authorID, firstName, lastName
   FROM authors
   ORDER BY lastName, firstName
```

authorID	firstName	lastName
4	David	Choffnes
1	Harvey	Deitel
2	Paul	Deitel
3	Andrew	Goldberg

`authors` sample data in ascending order by `lastName` and `firstName`.

```
SELECT isbn, title, editionNumber,
copyright
   FROM titles
   WHERE title LIKE '%How to Program'
   ORDER BY title ASC
```

isbn	title	editionNumber	copyright
0132404168	C How to Program	5	2007
0131857576	C++ How to Program	5	2005
0131450913	Internet and World Wide Web How to Program	3	2004
0132222205	Java How to Program	7	2007
0131869000	Visual Basic 2005 How to Program	3	2006
0131525239	Visual C# How to Program	2	2006

Sampling of books from table `titles` whose titles end with `How to Program` in ascending order by `title`.

Combining Data from Multiple Tables: `INNER JOIN`

- An `INNER JOIN` combines 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 specifies a condition that determines which rows are joined. This condition often compares columns from each table If a SQL statement uses columns with the same name from multiple tables, the column names must be fully qualified by prefixing them with their table names and a dot (`.`).

Software Engineering Observation

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

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

```
SELECT firstName, lastName, isbn
FROM authors
INNER JOIN authorISBN
   ON authors.authorID = authorISBN.authorID
ORDER BY lastName, firstName
```

firstName	lastName	isbn
David	Choffnes	0131828274
Harvey	Deitel	0131869000
Harvey	Deitel	0131525239
Harvey	Deitel	0132222205
Harvey	Deitel	0131857576
Harvey	Deitel	0132404168
Harvey	Deitel	0131450913
Harvey	Deitel	0131869000
Harvey	Deitel	0131828274
Paul	Deitel	0131869000
Paul	Deitel	0131525239
Paul	Deitel	0132222205
Paul	Deitel	0131857576
Paul	Deitel	0132404168
Paul	Deitel	0131450913
Paul	Deitel	0131869000
Paul	Deitel	0131828274
Andrew	Goldberg	0131450913

Sampling of authors and ISBNs for the books they have written in ascending order by `lastName` and `firstName`.

`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.
- SQL uses single quotes (`'`) as the delimiter for strings. To specify a string containing a single quote in SQL, the single quote must be escaped with another single quote.

Good Programming Practice

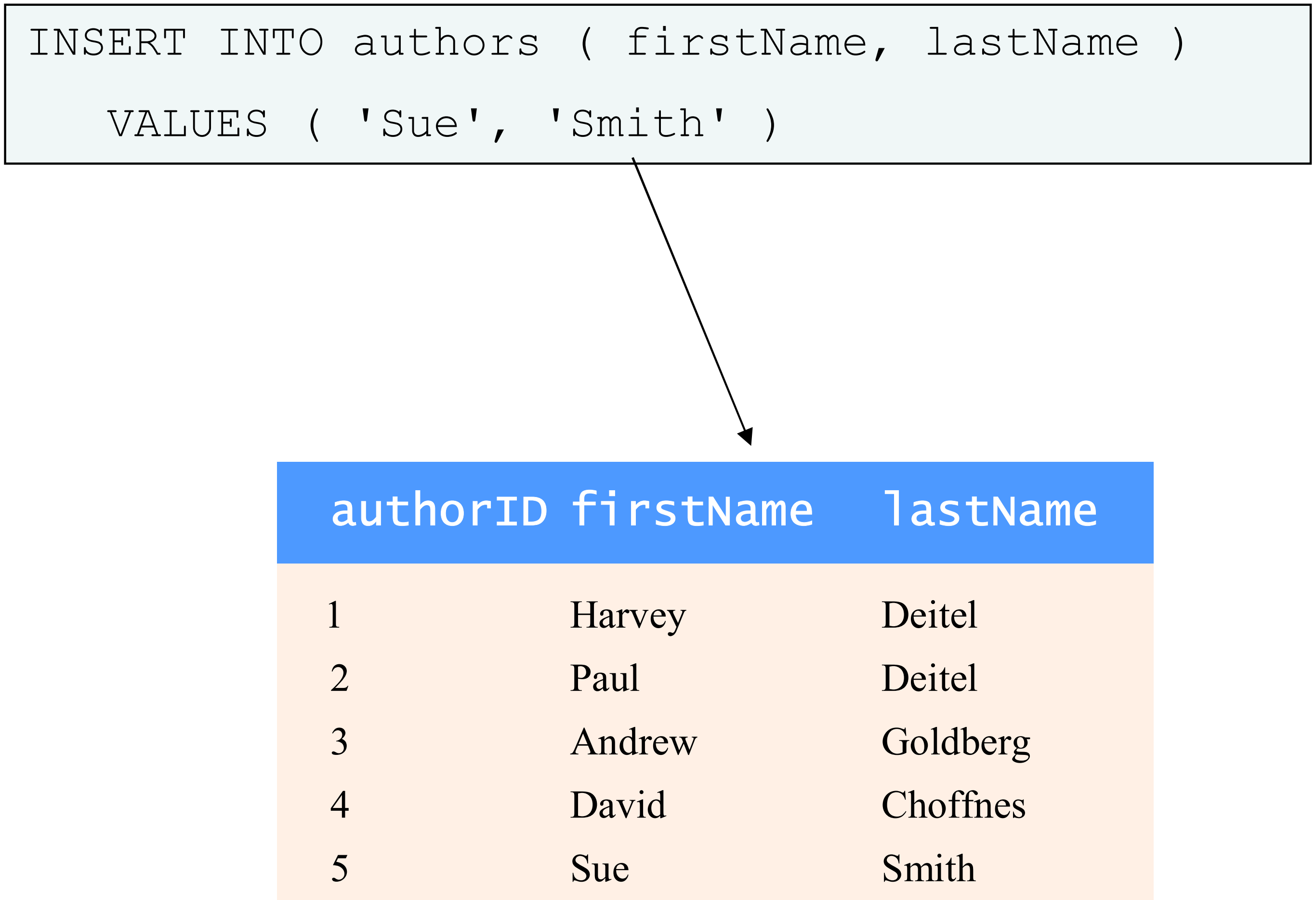
Always explicitly list the columns when inserting rows. If the table's column order changes or a new column is added, omitting the columns list may cause an error.

Common Programming Error

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

Common Programming Error

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



```
INSERT INTO authors ( firstName, lastName )
   VALUES ( 'Sue', 'Smith' )
```

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Andrew	Goldberg
4	David	Choffnes
5	Sue	Smith

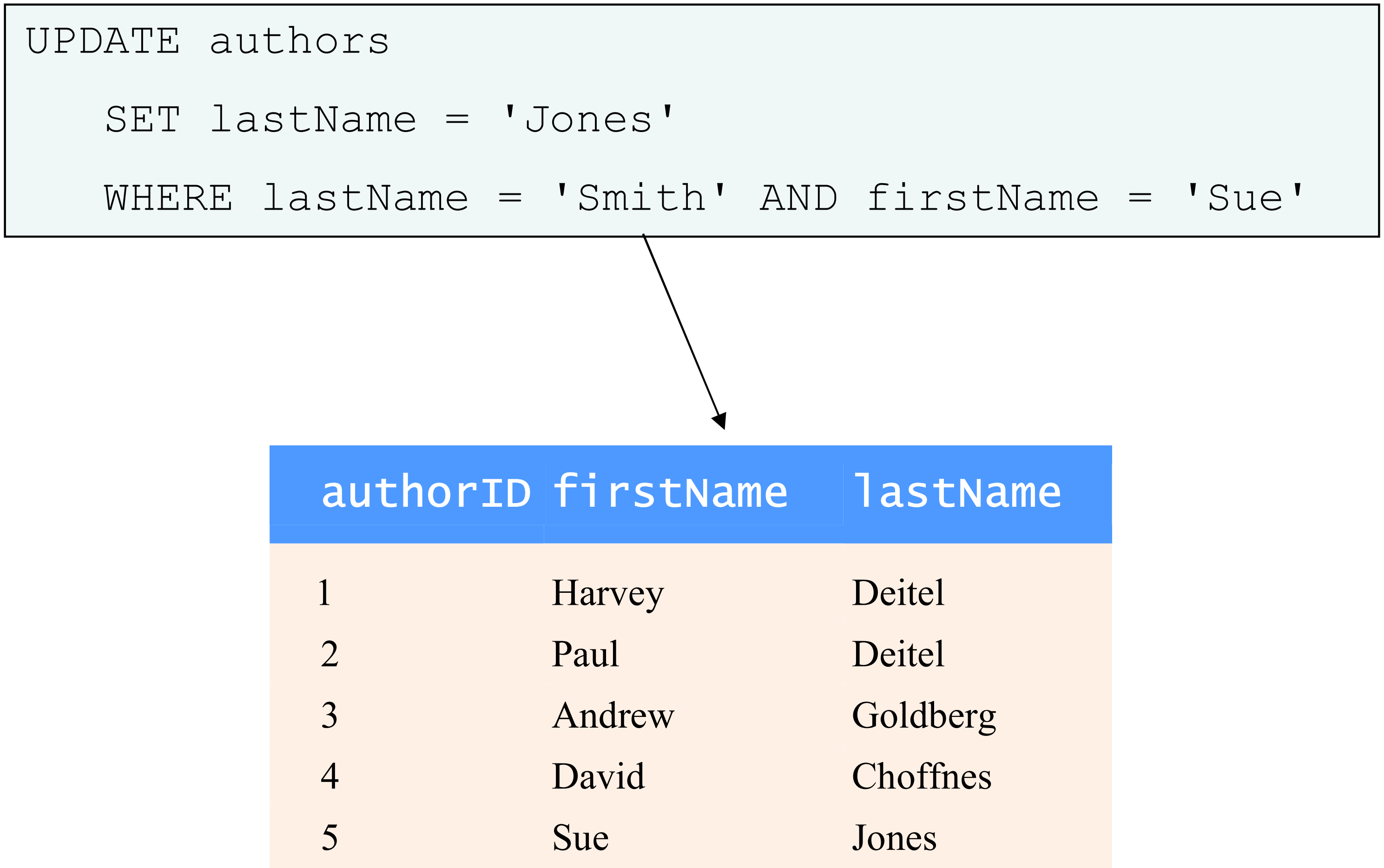
Sample data from table `Authors` after an `INSERT` operation.

UPDATE Statement

- An `UPDATE` statement modifies data in a table. The basic form of an `UPDATE` statement is

  ```
  UPDATE tableName
       SET columnName1 = value1,
           columnName2 = value2, …,
           columnNameN = valueN
       WHERE criteria
  ```

  where `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 = 'Jones'
   WHERE lastName = 'Smith' AND firstName = 'Sue'
```

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Andrew	Goldberg
4	David	Choffnes
5	Sue	Jones

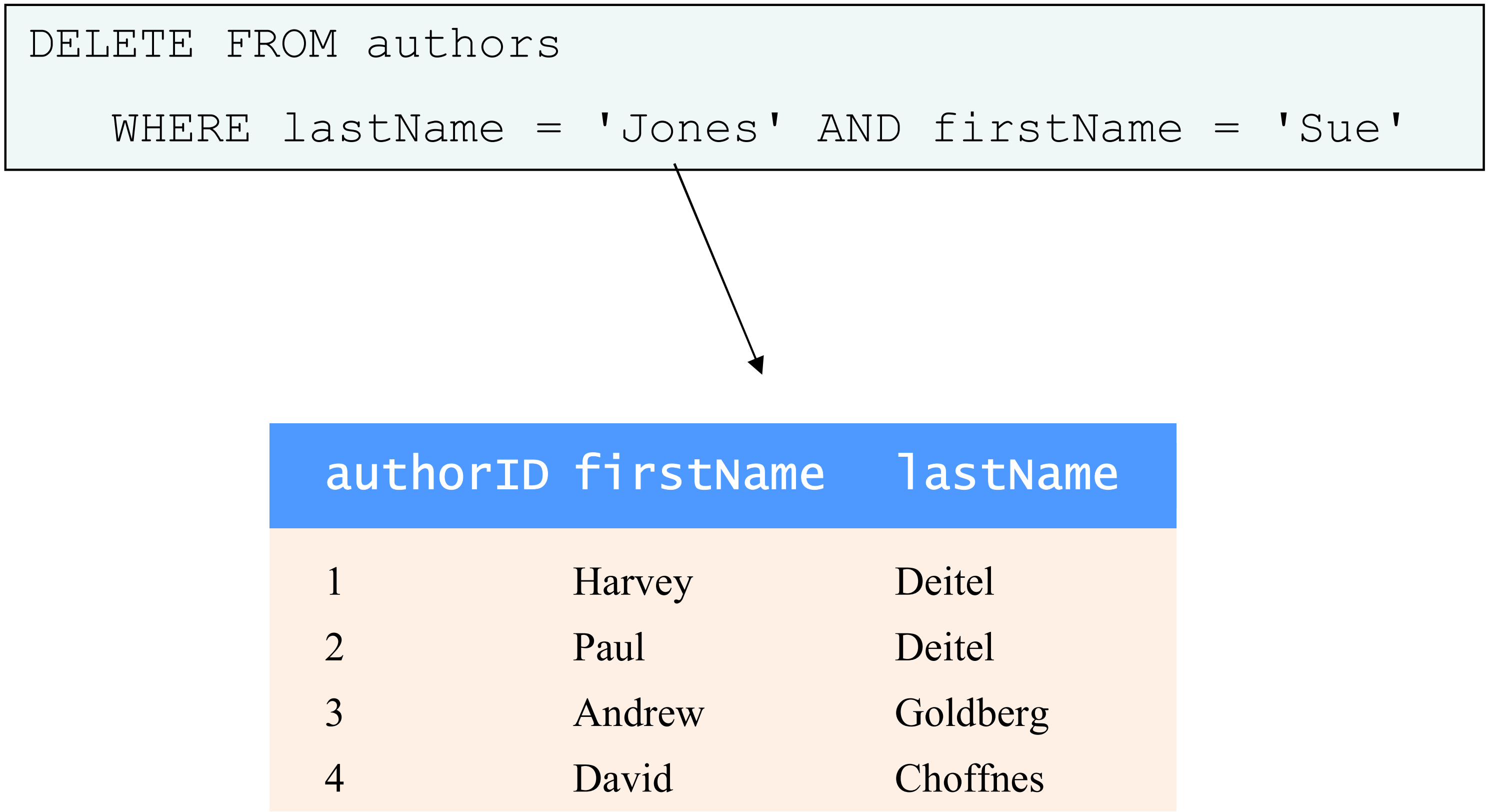
Sample data from table **`authors`** after an `UPDATE` operation.

DELETE Statement

- A DELETE statement removes rows from a table. The simplest form for a DELETE statement is

  ```
  DELETE FROM tableName WHERE
          criteria
  ```

  where *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 = 'Jones' AND firstName = 'Sue'
```

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Andrew	Goldberg
4	David	Choffnes

Sample data from table `authors` after a `DELETE` operation.

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 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.