

The books Database

- Sample **books** database - Four tables
 - **Authors**
 - **Publishers**
 - **authorISBN**
 - **titles**
- This example is borrowed from Chapter 8 of Deitel's book *Advanced Java 2 Platform*

The authors Table

Field	Description
authorID	Author's ID number in the database. In the books database, this integer field is defined as an <i>autoincremented field</i> . For each new record inserted in this table, the database automatically increments the authorID value to ensure that each record has a unique authorID . This field represents the table's primary key.
firstName	Author's first name (a string).
lastName	Author's last name (a string).

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Sean	Santry

The publishers Table

Field	Description
publisherID	The publisher's ID number in the database. This autoincremented integer is the table's primary-key field.
publisherName	The name of the publisher (a string).

publisherID	publisherName
1	Prentice Hall
2	Prentice Hall PTG

The authorISBN Table

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 field must also appear in the authors table.
isbn	The ISBN number for a book (a string).

Example Data for the authorISBN Table

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

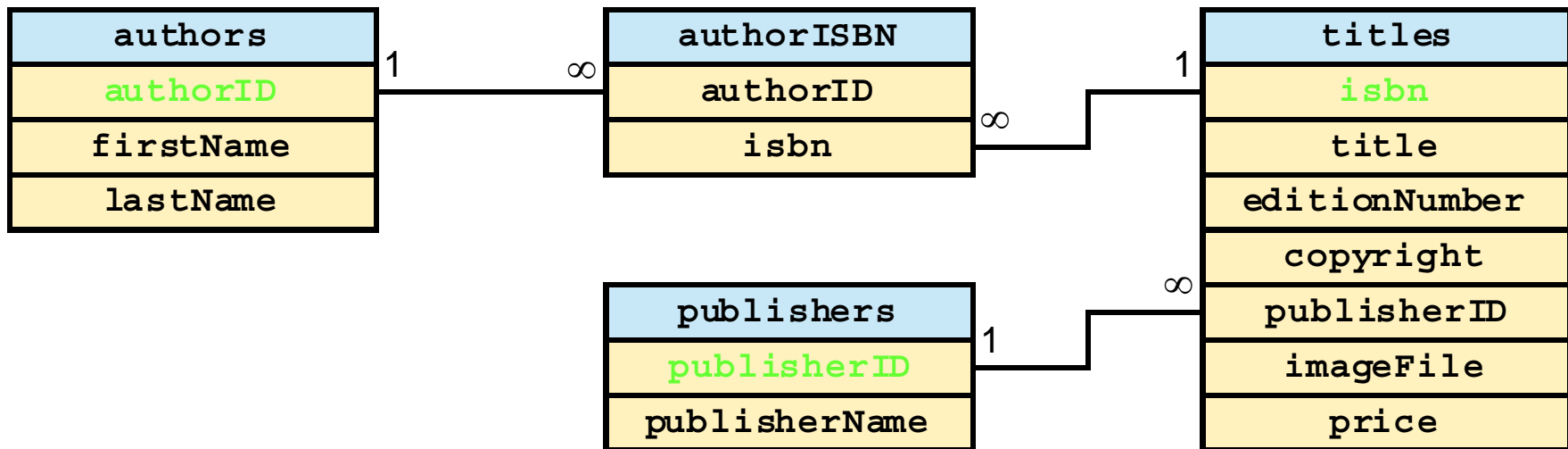
The titles Table

Field	Description
isbn	ISBN number of the book (a string).
title	Title of the book (a string).
editionNumber	Edition number of the book (an integer).
copyright	Copyright year 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). [<i>Note: The prices shown in this book are for example purposes only.</i>]

Sample Data from the titles Table

isbn	title	edition-Number	copy-right	publish-erID	image-File	price
0130895725	C How to Program	3	2001	1	chtp3.jpg	69.95
0132261197	C How to Program	2	1994	1	chtp2.jpg	49.95
0130895717	C++ How to Program	3	2001	1	cpphtp3.jpg	69.95
0135289106	C++ How to Program	2	1998	1	cpphtp2.jpg	49.95
0139163050	The Complete C++ Training Course	3	2001	2	cppctc3.jpg	109.95
013028419x	e-Business and e-Commerce How to Program	1	2001	1	ebechtp1.jpg	69.95
0130161438	Internet and World Wide Web How to Program	1	2000	1	iw3htp1.jpg	69.95
0130856118	The Complete Internet and World Wide Web Programming Training Course	1	2000	2	iw3ctc1.jpg	109.95
0130125075	Java How to Program (Java 2)	3	2000	1	jhtp3.jpg	69.95
0138993947	Java How to Program (Java 1.1)	2	1998	1	jhtp2.jpg	49.95
0130852473	The Complete Java 2 Training Course	3	2000	2	javactc3.jpg	109.95

The books Database – table relationships



Structured Query Language (SQL)

- SQL keywords

SQL keyword	Description
SELECT	Select (retrieve) fields from one or more tables.
FROM	Tables from which to get fields. Required in every SELECT .
WHERE	Criteria for selection that determine the rows to be retrieved.
GROUP BY	Criteria for grouping records.
ORDER BY	Criteria for ordering records.
INSERT INTO	Insert data into a specified table.
UPDATE	Update data in a specified table.
DELETE FROM	Delete data from a specified table.

Basic SELECT Query

- Simplest format of a SELECT query
 - **SELECT** * **FROM** tableName
 - **SELECT** * **FROM** authors
- Select specific fields from a table
 - **SELECT** authorID, lastName **FROM** authors

authorID	lastName
1	Deitel
2	Deitel
3	Nieto
4	Santry

WHERE Clause

- specify the selection criteria
 - **SELECT** fieldName1, fieldName2, ... **FROM** tableName
WHERE criteria
 - **SELECT** title, editionNumber, copyright
FROM titles
WHERE copyright > 1999
- **WHERE** clause condition operators
 - <, >, <=, >=, =, <>
 - **LIKE**
 - wildcard characters % and _

WHERE Clause (Cont.)

title	editionNumber	copyright
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
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

WHERE Clause (Cont.)

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

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel

8.4.2 WHERE Clause (Cont.)

- **SELECT** authorID, firstName, lastName
FROM authors
WHERE lastName **LIKE** '_i%'

authorID	firstName	lastName
3	Tem	Nieto

ORDER BY Clause

- Optional **ORDER BY** clause
 - **SELECT** fieldName1, fieldName2, ... **FROM** tableName
ORDER BY field **ASC**
 - **SELECT** fieldName1, fieldName2, ... **FROM** tableName
ORDER BY field **DESC**
- **ORDER BY** multiple fields
 - **ORDER BY** field1 sortingOrder, field2 sortingOrder, ...
- Combine the **WHERE** and **ORDER BY** clauses

ORDER BY Clause (Cont.)

- **SELECT** authorID, firstName, lastName
FROM authors
ORDER BY lastName **ASC**

authorID	firstName	lastName
2	Paul	Deitel
1	Harvey	Deitel
3	Tem	Nieto
4	Sean	Santry

8.4.3 ORDER BY Clause (Cont.)

- **SELECT** authorID, firstName, lastName
FROM authors
ORDER BY lastName **DESC**

authorID	firstName	lastName
4	Sean	Santry
3	Tem	Nieto
2	Paul	Deitel
1	Harvey	Deitel

ORDER BY Clause (Cont.)

- **SELECT** authorID, firstName, lastName
FROM authors
ORDER BY lastName, firstName

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Sean	Santry

ORDER BY Clause (Cont.)

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

isbn	title	edition- Number	copy- right	price
0130895601	Advanced Java 2 Platform How to Program	1	2002	69.95
0132261197	C How to Program	2	1994	49.95
0130895725	C How to Program	3	2001	69.95
0135289106	C++ How to Program	2	1998	49.95
0130895717	C++ How to Program	3	2001	69.95
0130161438	Internet and World Wide Web How to Program	1	2000	69.95
0130284181	Perl How to Program	1	2001	69.95
0134569555	Visual Basic 6 How to Program	1	1999	69.95
0130284173	XML How to Program	1	2001	69.95
013028419x	e-Business and e-Commerce How to Program	1	2001	69.95

Merging Data from Multiple Tables: Joining

- Join the tables

- Merge data from multiple tables into a single view

- **SELECT** fieldName1, fieldName2, ...

- FROM** table1, table2

- WHERE** table1.fieldName = table2.fieldName

- **SELECT** firstName, lastName, isbn

- FROM** authors, authorISBN

- WHERE** authors.authorID = authorISBN.authorID

- ORDER BY** lastName, firstName

Merging Data from Multiple Tables: Joining (Cont.)

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

INSERT INTO Statement

- Insert a new record into a table
 - **INSERT INTO** tableName (fieldName1, ... , fieldNameN)
VALUES (value1, ... , valueN)
 - **INSERT INTO** authors (firstName, lastName)
VALUES ('Sue', 'Smith')

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Sean	Santry
5	Sue	Smith

UPDATE Statement

- Modify data in a table

- **UPDATE** tableName

- SET** fieldName1 = value1, ... , fieldNameN = valueN

- WHERE** criteria

- **UPDATE** authors

- SET** lastName = 'Jones'

- WHERE** lastName = 'Smith' **AND** firstName = 'Sue'

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Sean	Santry
5	Sue	Jones

DELETE FROM Statement

- Remove data from a table
 - **DELETE FROM** tableName **WHERE** criteria
 - **DELETE FROM** authors
 - WHERE** lastName = 'Jones' **AND** firstName = 'Sue'

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Sean	Santry

Seven Basic Steps in Using JDBC

1. Load the driver
2. Define the Connection URL
3. Establish the Connection
4. Create a Statement object
5. Execute a query
6. Process the results
7. Close the connection

JDBC: Details of Process

1. Load the driver

```
try {  
    Class.forName("oracle.jdbc.driver.OracleDriver");  
    Class.forName("org.gjt.mm.mysql.Driver");  
} catch (ClassNotFoundException cnfe) {  
    System.out.println("Error loading driver: " + cnfe);  
}
```

2. Define the Connection URL

```
String host = "dbhost.yourcompany.com";  
String dbName = "someName";  
int port = 1234;  
String oracleURL = "jdbc:oracle:thin:@" + host +  
    ":" + port + ":" + dbName;  
String mysqlURL = "jdbc:mysql://" + host +  
    ":" + port + "/" + dbName;
```

JDBC: Details of Process, cont.

3. Establish the Connection

```
String username = "jay_debese";
String password = "secret";
Connection connection =
    DriverManager.getConnection(oracleURL,
                                username,
                                password);
```

- Optionally, look up information about the database

```
DatabaseMetaData dbMetaData =
connection.getMetaData();
String productName =
    dbMetaData.getDatabaseProductName();
System.out.println("Database: " + productName);
String productVersion =
    dbMetaData.getDatabaseProductVersion();
System.out.println("Version: " + productVersion);
```

JDBC: Details of Process, cont.

4. Create a Statement

```
Statement statement = connection.createStatement();
```

5. Execute a Query

```
String query = "SELECT col1, col2, col3 FROM  
sometable";
```

```
ResultSet resultSet = statement.executeQuery(query);
```

- To modify the database, use `executeUpdate`, supplying a string that uses `UPDATE`, `INSERT`, or `DELETE`
- Use `setQueryTimeout` to specify a maximum delay to wait for results

JDBC: Details of Process, cont.

6. Process the Result

```
while(resultSet.next()) {  
    System.out.println(resultSet.getString(1) + " " +  
                        resultSet.getString(2) + " " +  
                        resultSet.getString(3));  
}
```

- First column has index 1, not 0
- ResultSet provides various getXxx methods that take a column index or name and returns the data

7. Close the Connection

```
connection.close();
```

- As opening a connection is expensive, postpone this step if additional database operations are expected

Basic JDBC Example

```
import java.sql.*;

public class TestDB {
    public static void main(String[] args) {

        // Use driver from Connect SW.
        String driver = "connect.microsoft.MicrosoftDriver";
        try {
            Class.forName(driver);
            String url = "jdbc:ff-microsoft://" +           // FastForward
                        "dbtest.apl.jhu.edu:1433/" +      // Host:port
                        "pubs";                            // Database name
            String user = "sa", password="";

            Connection connection =
                DriverManager.getConnection(url, user, password);
            Statement statement = connection.createStatement();
            String query =
                "SELECT col1, col2, col3 FROM testDB";

            // Execute query and save results.
            ResultSet results = statement.executeQuery(query);
```

Basic JDBC Example, cont.

```
// Print column names.
String divider = "-----+-----+-----";
System.out.println("Col1 | Col2 | Col3\n" + divider);

// Print results
while(results.next()) {
    System.out.println
        (pad(results.getString(1), 4) + " | " +
         pad(results.getString(2), 4) + " | " +
         results.getString(3) + "\n" + divider);
}
connection.close();
} catch(ClassNotFoundException cnfe) {
    System.out.println("No such class: " + driver);
} catch(SQLException se) {
    System.out.println("SQLException: " + se);
}
}
...

```