

30. Structured Query Language (SQL)

Java

Fall 2009

Instructor: Dr. Masoud Yaghini

Outline

- SQL query keywords
- Basic **SELECT** Query
- **WHERE** Clause
- **ORDER BY** Clause
- **INNER JOIN** Clause
- **INSERT** Statement
- **UPDATE** Statement
- **DELETE** Statement
- References

SQL query keywords



SQL query keywords

- **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.

SQL query keywords

- **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.

Basic **SELECT** Query



Basic SELECT Query

- **SELECT** query selects rows and columns from one or more tables in a database.
- The basic form of a **SELECT** query is:
`SELECT * FROM tableName`
- Example:
`SELECT * FROM authors`
- The asterisk (*) indicates that all columns from the `tableName` table should be retrieved.

Basic SELECT Query

- To retrieve only specific columns from a table, replace the asterisk (*) with a comma-separated list of the column names.
- Example:
`SELECT authorID, lastName FROM authors`

<code>authorID</code>	<code>lastName</code>
1	Deitel
2	Deitel
3	Nieto
4	Santry

Basic SELECT Query

- Selecting columns by name avoids returning unneeded columns and protects against changes in the actual order of the columns in the table(s).

WHERE Clause



WHERE Clause

- In most cases, Only rows that satisfy the **selection criteria** (formally called **predicates**) are selected.
- 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
```

WHERE Clause

- Example:
SELECT title, editionNumber, copyright
FROM titles
WHERE copyright > '2002'

title	editionNumber	copyright
The Complete C++ Training Course	4	2003
Java How to Program	5	2003
C How to Program	4	2004
Internet and World Wide Web How to Program	3	2004
Java How to Program	6	2005
C# How to Program	1	2003

WHERE Clause

- WHERE clause condition operators
 - <, >, <=, >=, =, <>
 - LIKE
 - wildcard characters * and ?

WHERE Clause

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

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel

WHERE Clause

- For example, the following query locates the rows of all the authors whose last names start with any character (specified by ?), followed by the letter i, followed by any number of additional characters (specified by *):

```
SELECT authorID, firstName, lastName  
FROM authors  
WHERE lastName LIKE '?i*'
```

authorID	firstName	lastName
3	Tem	Nieto



ORDER BY Clause



ORDER BY Clause

- The rows in the result of a query can be sorted into ascending or descending order by using the optional **ORDER BY** clause.
- The basic form of a query with an **ORDER BY** clause is:

```
SELECT columnName1, columnName2, ...  
FROM tableName ORDER BY column ASC
```

```
SELECT columnName1, columnName2, ...  
FROM tableName ORDER BY column DESC
```

ORDER BY Clause

- Example:

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

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

ORDER BY Clause

- Example:

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

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

ORDER BY Clause

- Multiple columns can be used for sorting with an **ORDER BY** clause of the form
`ORDER BY column1 sortingOrder, column2 sortingOrder, ...`

- Example:

```
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

- The **WHERE** and **ORDER BY** clauses can be combined in one query. For example, the query **SELECT isbn, title, editionNumber, copyright, price FROM titles WHERE title LIKE '*How to Program' ORDER BY title ASC**

isbn	title	editionNumber	copyright	price
0130895601	Advanced Java 2 Platform How to Program	1	2002	69.95
0131426443	C How to Program	4	2004	85.00
0130384747	C++ How to Program	4	2003	85.00



INNER JOIN Clause



INNER JOIN Clause

- Often, it is necessary to merge data from multiple tables into a single result.
- Referred to as joining the tables, this is specified by an **INNER JOIN** operator in the query.
- An **INNER JOIN** merges rows from two tables by matching values in columns that are common to the tables. The basic form of an **INNER JOIN** is:

```
SELECT columnName1, columnName2,..
```

```
FROM table1
```

```
INNER JOIN table2
```

```
ON table1.columnName = table2.columnName
```

INNER JOIN Clause

- For example, the following query produces a list of authors accompanied by the ISBNs for books written by each author:

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


INNER JOIN Clause

- The result:

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

INNER JOIN Clause

- If a SQL statement includes columns from multiple tables that have the same name, the statement must precede those column names with their table names and a dot (e.g., `authors.authorID`).

INSERT Statement



INSERT Statement

- The **INSERT** statement inserts a row into a table. The basic form of this statement is
INSERT
INTO tableName (columnName1, columnName2, ..., columnNameN)
VALUES (value1, value2, ..., valueN)
- **Example:**
INSERT INTO authors (firstName, lastName)
VALUES ('Sue', 'Smith')

INSERT Statement

- For every row added to this table, MS-Access assigns a unique **authorID** value that is the next value in the autoincremented sequence (i.e., 1, 2, 3 and so on).
- Sample data from table Authors after an **INSERT** operation

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

INSERT Statement

- 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.

UPDATE Statement



UPDATE Statement

- An **UPDATE** statement modifies data in a table.
- The basic form of the **UPDATE** statement is

UPDATE tableName

SET columnName1 = value1, columnName2 = value2,
..., columnNameN = valueN

WHERE criteria

UPDATE Statement

- Example:

```
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 Statement



DELETE Statement

- A SQL **DELETE** statement removes rows from a table.
- The basic form of a **DELETE** statement is

DELETE FROM tableName **WHERE** criteria

DELETE Statement

- Example:

```
DELETE FROM authors
```

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

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



References



References

- H. M. Deitel and P. J. Deitel, **Java™ How to Program**, Sixth Edition, Prentice Hall, 2005.
(Chapter 25)



The End