Chapter 3
The Relational Model 2: SQL
Table Creation

• **SQL CREATE TABLE** command
  – Creates a table by describing its layout

• Typical restrictions placed on table and column names by DBMS
  – Names cannot exceed 18 characters
  – Names must start with a letter
  – Names can contain only letters, numbers, and underscores (_)
  – Names cannot contain spaces
Table Creation (continued)

- **INTEGER**
  - Number without a decimal point
- **SMALLINT**
  - Uses less space than INTEGER
- **DECIMAL(p,q)**
  - P number of digits; q number of decimal places
- **CHAR(n)**
  - Character string n places long
- **DATE**
  - Dates in DD-MON-YYYY or MM/DD/YYYY form
Simple Retrieval

- **SELECT-FROM-WHERE**: SQL retrieval command
  - **SELECT clause**: lists fields to display
  - **FROM clause**: lists table or tables that contain data to display in query results
  - **WHERE clause** (optional): lists any conditions to be applied to the data to retrieve

- **Simple condition**: field name, a comparison operator, and either another field name or a value
Simple Retrieval (continued)

FIGURE 3-6: SQL query with WHERE condition
Simple Retrieval (continued)

FIGURE 3-7: Query results

Customers with credit limits of $10,000
Simple Retrieval (continued)

<table>
<thead>
<tr>
<th>Comparison Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equal to</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Not equal to (used by most implementations of SQL)</td>
</tr>
<tr>
<td>!=</td>
<td>Not equal to (used by some implementations of SQL)</td>
</tr>
</tbody>
</table>

FIGURE 3-8: Comparison operators used in SQL commands
Compound Conditions

• **Compound condition**
  – Connecting two or more simple conditions using one or both of the following operators: AND and OR
  – Preceding a single condition with the NOT operator
• Connecting simple conditions using AND operator
  – *All* of the simple conditions must be true for the compound condition to be true
• Connecting simple conditions using OR operator
  – *Any* of the simple conditions must be true for the compound condition to be true
Compound Conditions (continued)

FIGURE 3-15: Compound condition that uses the AND operator

FIGURE 3-16: Query results
Compound Conditions (continued)

FIGURE 3-17: Compound condition that uses the OR operator

Because the condition contains the OR operator, either or both conditions must be true for a record to appear in the query results.

FIGURE 3-18: Query results

Parts in warehouse 3 or with more than 20 units on hand or both.
Compound Conditions (continued)

• Preceding a condition by NOT operator
  – Reverses the truth or falsity of the original condition

• BETWEEN operator
  – Value must be between the listed numbers
Computed Fields

• Computed field or calculated field
  – Field whose values you derive from existing fields
  – Can involve:
    • Addition (+)
    • Subtraction (-)
    • Multiplication (*)
    • Division (/)
Computed Fields (continued)

FIGURE 3-25: SQL query with a computed field and condition

FIGURE 3-26: Query results
Using Special Operators (LIKE and IN)

- Wildcards in Access SQL
  - Asterisk (*): collection of characters
  - Question mark (?): any individual character
- Wildcards in MySQL
  - Percent sign (%): any collection of characters
  - Underscore (_): any individual character
- To use a wildcard, include the LIKE operator in the WHERE clause
- IN operator provides a concise way of phrasing certain conditions
Using Special Operators (LIKE and IN) (continued)

FIGURE 3-27: SQL query with a LIKE operator

FIGURE 3-28: Query results
Using Special Operators (LIKE and IN) (continued)

**FIGURE 3-28:** SQL query with an IN operator

**FIGURE 3-29:** Query results
Sorting

- Sort data using the **ORDER BY clause**
- Sort key: field on which to sort data
- When sorting data on two fields:
  - Major sort key (or primary sort key): more important sort key
  - Minor sort key (or secondary sort key): less important sort key
Sorting (continued)

FIGURE 3-33: SQL query to sort data on multiple fields

FIGURE 3-34: Query results
Built-in Functions

• Built-in functions (aggregate functions) in SQL
  – COUNT: calculates number of entries
  – SUM or AVG: calculates sum or average of all entries in a given column
  – MAX or MIN: calculates largest or smallest values respectively
Built-in Functions (continued)

**FIGURE 3-35:** SQL query to count records

```sql
SELECT COUNT(*)
FROM Part
WHERE Class='HW';
```

**FIGURE 3-36:** Query results

- **COUNT function**
- **Condition to select records in the HW item class**
- **Column heading created by Access for a field containing an expression**
- **Number of records in item class HW**
Subqueries

- **Subquery**: inner query
- Subquery is evaluated first
- Outer query is evaluated after the subquery
Subqueries (continued)

FIGURE 3-41: SQL query with a subquery

FIGURE 3-42: Query results
Grouping

• Create groups of records that share a common characteristic

• **GROUP BY clause** indicates grouping in SQL

• **HAVING clause** is to groups what the WHERE clause is to rows
Grouping (continued)

**FIGURE 3-45:** SQL query to restrict the groups that are included

```
SELECT RepNum, COUNT(*) AS NumCustomers, AVERAGE(Balance) AS AverageBalance
FROM Customer
GROUP BY RepNum
HAVING COUNT(*) < 4
ORDER BY RepNum
```

Only groups with fewer than four records will be included.

**FIGURE 3-46:** Query results
Joining Tables

• Queries can locate data from more than one table
• Enter appropriate conditions in the WHERE clause
• To join tables, construct the SQL command as:
  1. SELECT clause: list all fields you want to display
  2. FROM clause: list all tables involved in the query
  3. WHERE clause: give the condition that will restrict
     the data to be retrieved to only those rows from the
     two tables that match
Joining Tables (continued)

FIGURE 3-49: SQL query to join tables
Joining Tables (continued)

FIGURE 3-50: Query results
Union

• Union of two tables is a table containing all rows in the first table, the second table, or both tables
• Two tables involved must be union compatible
  – Same number of fields
  – Corresponding fields must have same data types
Union (continued)

FIGURE 3-55: SQL query to perform a union

FIGURE 3-56: Query results
Updating Tables

• **UPDATE** command makes changes to existing data
• **INSERT** command adds new data to a table
• **DELETE** command deletes data from the database
Updating Tables (continued)

**FIGURE 3-57:** SQL query to update data

**FIGURE 3-58:** SQL query to insert a row
FIGURE 3-59: SQL query to delete rows

- **Table from which to delete rows**
- **Condition to select part number BV06**
Creating a Table from a Query

• **INTO clause**
  – Saves the results of a query as a table
  – Specified before FROM and WHERE clauses

• **MySQL**
  – Create the new table using a CREATE TABLE command
  – Use an INSERT command to insert the appropriate data into the new table
Creating a Table from a Query (continued)

FIGURE 3-60a: Query to create a new table (Access)