

10.5

Structured Query Language (SQL)

SQL Is a Standard Language for Working With Databases



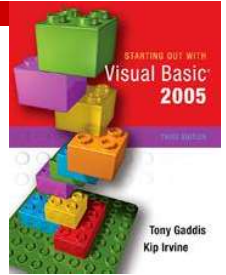
The Select Statement

- *Select* retrieves rows from one or more tables in a database
 - Basic form of Select for a single table is
- The following Select statement retrieves the ID and Salary fields from the SalesStaff table

```
Select column-list  
From table
```

- `column-list` contains column names to select from `table`, each separated by a comma

```
Select ID, Salary  
From SalesStaff
```



Column Names

- Use asterisk to select all columns in a table

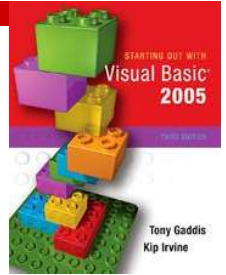
```
Select *  
From SalesStaff
```

- Unlike VB names, SQL columns can have embedded spaces
 - If so, use square brackets around column names

```
Select [Last Name], [First Name]  
From SalesStaff
```

- Better to avoid embedded spaces for this reason
- **As** operator can be used to rename columns

```
Select Last_Name, Hire_Date As Date_Hired  
From SalesStaff
```

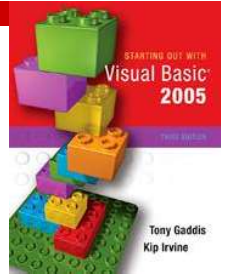


Creating New Columns

- Sometimes useful to create a new column by appending existing columns together
 - Create a Full_Name field from first and last name

```
Select Last_Name + ', ' + First_Name as Full_Name
From SalesStaff
```
 - Creates a Full_Name field in the format *last, first*
- Can also be useful to create a new column by performing arithmetic operations
 - Columns involved must be numeric

```
Select ID, hrsWorked * hourlyRate As payAmount
From Payroll
```
 - Creates a payAmount column with gross pay



Sorting Rows with Order By Clause

- SQL Select has an optional *Order By* clause that affects the order in which rows appear

```
Order by Last_Name, First_Name
```

- Displays rows in order by last name, then first
- Sort in reverse order (high to low) using *Desc*

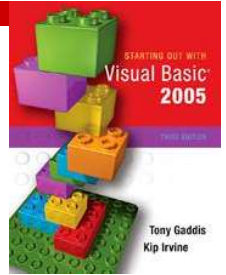
```
Order by Last_Name DESC
```

- *Order By* clause appears after *From* clause

```
Select First_Name, Last_Name, Date_Joined  
From Members
```

```
Order By Last_Name, First_Name
```

- Lists all members by last name, then first



Selecting Rows with Where Clause

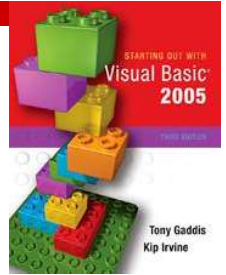
- SQL Select has an optional *Where* clause that can be used to select (or filter) certain rows

```
Where Last_Name = 'Gomez'
```

- Displays only rows where last name is Gomez
 - Must be a defined column (in table or created)
- This example selects based on a created field

```
Select Last_Name, hrsWorked * Rate As payAmount  
From Payroll  
Where payAmount > 1000  
Order by Last_Name
```

- Selects those being paid more than \$1,000

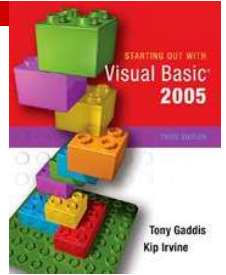


SQL Relational Operators

- SQL *Where* uses relational operators just like a VB *If*

<u>Operator</u>	<u>Meaning</u>
=	equal to
<>	not equal to
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
Between	between two values (inclusive)
Like	similar to (match using wildcard)

- Example of Between operator:
Where Hire_Date Between #1/1/1992# and #12/31/1999#
- Example of Like operator with % sign as wildcard:
Where Last_Name Like 'A%'



Compound Expressions

- SQL uses *And*, *Or*, and *Not* to create compound expressions
- Select all employees hired after 1/1/1990 *and* with a salary is greater than \$40,000
`Where (Hire_Date > #1/1/1990#) and (Salary > 40000)`
- Select all employees hired after 1/1/1990 *or* with a salary is greater than \$40,000
`Where (Hire_Date > #1/1/1990#) or (Salary > 40000)`
- Select employee names not beginning with A
`Where Last_Name Not Like 'A%'`