# Session 6-5 Single-Row Functions

Kyunghee Lee, PhD

# Terminology

- Function – predefined block of code that accepts arguments

- Single-row function – returns one row of results for each record processed

- Multiple-row function – returns one result per group of data processed (covered in the next chapter)

- Oracle SQL Functions (https://docs.oracle.com/cd/B19306_01/server.102/b14200/functions001.htm)

**WAYNE STATE**
Mike Ilitch School of Business

# Types of Functions

| TYPE OF FUNCTIONS | FUNCTIONS |
|---|---|
| Case conversion functions | UPPER, LOWER, INITCAP |
| Character manipulation functions | SUBSTR, INSTR, LENGTH, LPAD/RPAD, RTRIM/LTRIM, REPLACE, TRANSLATE, CONCAT |
| Numeric functions | ROUND, TRUNC, MOD, ABS |
| Date functions | MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY, TO_DATE, ROUND, CURRENT_DATE |
| Regular expressions | REGEXP_LIKE, REGEXP_SUBSTR |
| Other functions | NVL, NVL2, TO_CHAR, DECODE, SOUNDEX, TO_NUMBER |

**FIGURE 10-1**   Functions covered in this chapter

**WAYNE STATE**
Mike Ilitch School of Business

# Case Conversion Functions

- Case conversion functions alter the case of data stored in a column or character string:
  - Used in a SELECT clause they alter the appearance of the data in the results
  - Used in a WHERE clause they alter the value for comparison

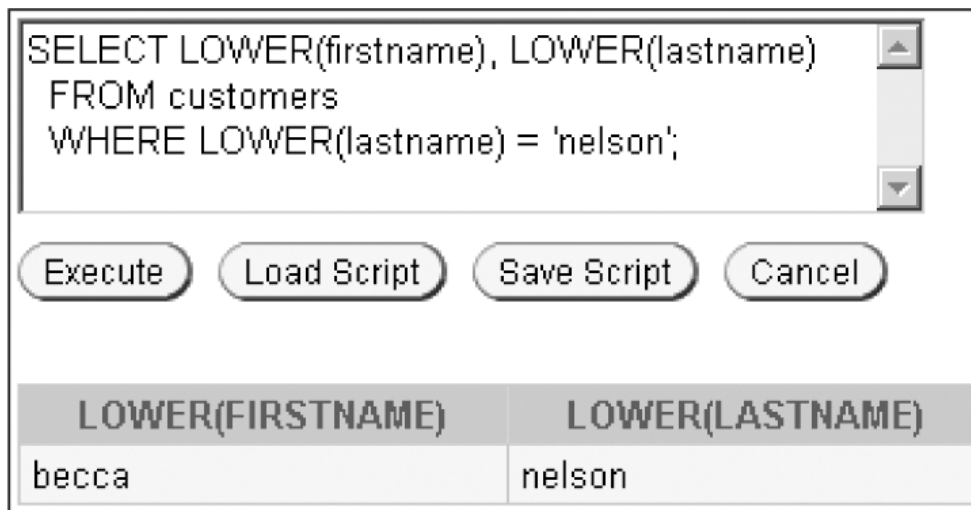| Function | Description | Syntax |
|----------|-------------|--------|
| LOWER | Convert characters to lowercase letters | LOWER(*string*) |
| UPPER | Convert characters to UPPERCASE letters | UPPER(*string*) |
| INITCAP | Convert characters to Mixed Case | INITCAP(*string*) |

**WAYNE STATE**
Mike Ilitch School of Business

# LOWER Function

- Used to convert characters to lowercase letters



```
SELECT LOWER(firstname), LOWER(lastname)
  FROM customers
  WHERE LOWER(lastname) = 'nelson';
```

| Execute | Load Script | Save Script | Cancel |
| --- | --- | --- | --- |

| LOWER(FIRSTNAME) | LOWER(LASTNAME) |
| --- | --- |
| becca | nelson |

**FIGURE 10-3** LOWER function in SELECT and WHERE clauses

WAYNE STATE
Mike Ilitch School of Business

»5

# Character Manipulation Functions

- Character manipulation functions manipulate data by extracting substrings, counting the number of characters, replacing strings, etc.

# SUBSTR Function

- Used to return a substring, or portion of a string
- **SUBSTR(*string, start*, length)**

```
SELECT DISTINCT zip, SUBSTR(zip, 1, 3), SUBSTR(zip, -3, 2)
FROM customers;
```

Execute    Load Script    Save Script    Cancel

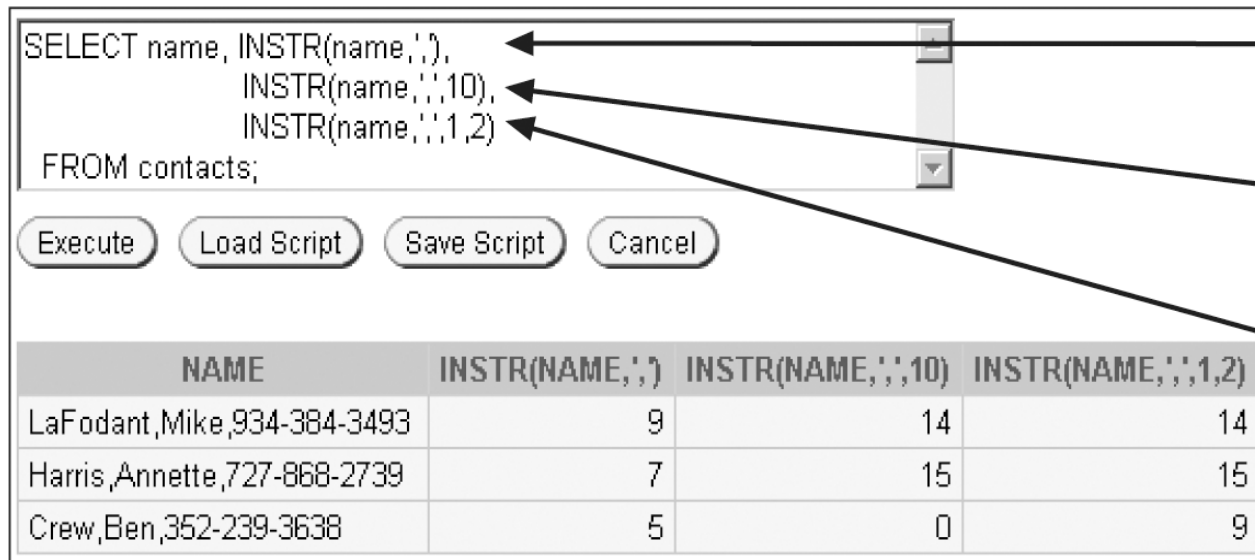| ZIP | SUBSTR(ZIP,1 | SUBSTR(Z |
|-----|--------------|----------|
| 83707 | 837 | 70 |
| 98115 | 981 | 11 |
| 12211 | 122 | 21 |
| 91508 | 915 | 50 |

**WAYNE STATE**
Mike Ilitch School of Business

# INSTR Function

- Locate a substring within a character string
- **INSTR(*string, lookup_char, start, occurrence)***



SELECT name, INSTR(name,','),
          INSTR(name,',',10),
          INSTR(name,',',1,2)
FROM contacts;

Locate the position of the first comma

Locate the position of the first comma starting with position 10
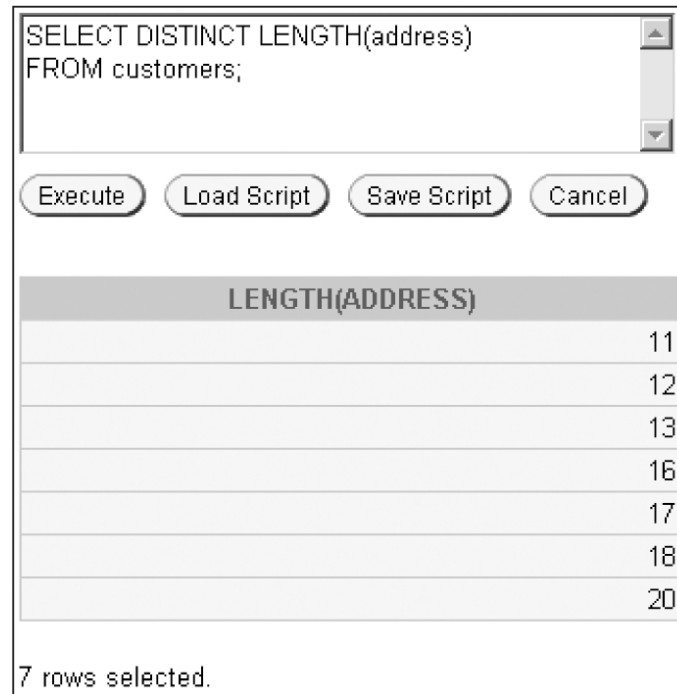
Locate the position of the second comma

| NAME | INSTR(NAME,',') | INSTR(NAME,',',10) | INSTR(NAME,',',1,2) |
|---|---|---|---|
| LaFodant,Mike,934-384-3493 | 9 | 14 | 14 |
| Harris,Annette,727-868-2739 | 7 | 15 | 15 |
| Crew,Ben,352-239-3638 | 5 | 0 | 9 |

# LENGTH Function

- Used to determine the number of characters in a string



```
SELECT DISTINCT LENGTH(address)
FROM customers;
```

Execute | Load Script | Save Script | Cancel

| LENGTH(ADDRESS) |
|---|
| 11 |
| 12 |
| 13 |
| 16 |
| 17 |
| 18 |
| 20 |

7 rows selected.

# LPAD and RPAD Functions

- Used to pad, or fill in, a character string to a fixed width
- **LPAD(*string, # of characters to add, char*)**

```
SELECT firstname, LPAD(firstname,12,' '), LPAD(firstname,12,'*')
  FROM customers
  WHERE firstname LIKE 'J%';
```

Execute    Load Script    Save Script    Cancel

```
FIRSTNAME                      LPAD(FIRSTNAME,12,'')              LPAD(FIRSTNAME,12,'*')
------------------------------ ----------------------------------- -----------------------------
JORGE                                 JORGE                        *******JORGE
JAKE                                   JAKE                        ********JAKE
JASMINE                             JASMINE                        *****JASMINE
JENNIFER                           JENNIFER                        ****JENNIFER
```

WAYNE STATE
**Mike Ilitch School of Business**

# LTRIM and RTRIM Functions

- Used to remove a specific string of characters

```
SELECT firstname, lastname, address, LTRIM(address, 'P.O. BOX')
FROM customers
WHERE address LIKE 'P.O. BOX%';
```

[Execute] [Load Script] [Save Script] [Cancel]

| FIRSTNAME | LASTNAME | ADDRESS | LTRIM(ADDRESS,'P.O.B |
|-----------|----------|---------|----------------------|
| BONITA | MORALES | P.O. BOX 651 | 651 |
| RYAN | THOMPSON | P.O. BOX 9835 | 9835 |

WAYNE STATE
Mike Ilitch School of Business

# REPLACE Function

- Substitutes a string with another specified string
- **REPLACE(*string, search string, replacement_string*)**



```
SELECT address, REPLACE(address, 'P.O. ', 'POST OFFICE ')
FROM customers;
```

Execute   Load Script   Save Script   Cancel

| ADDRESS | REPLACE(ADDRESS,'P.O.','POSTOFFI |
|---------|----------------------------------|
| P.O. BOX 651 | POST OFFICE BOX 651 |
| P.O. BOX 9835 | POST OFFICE BOX 9835 |

# Number Functions
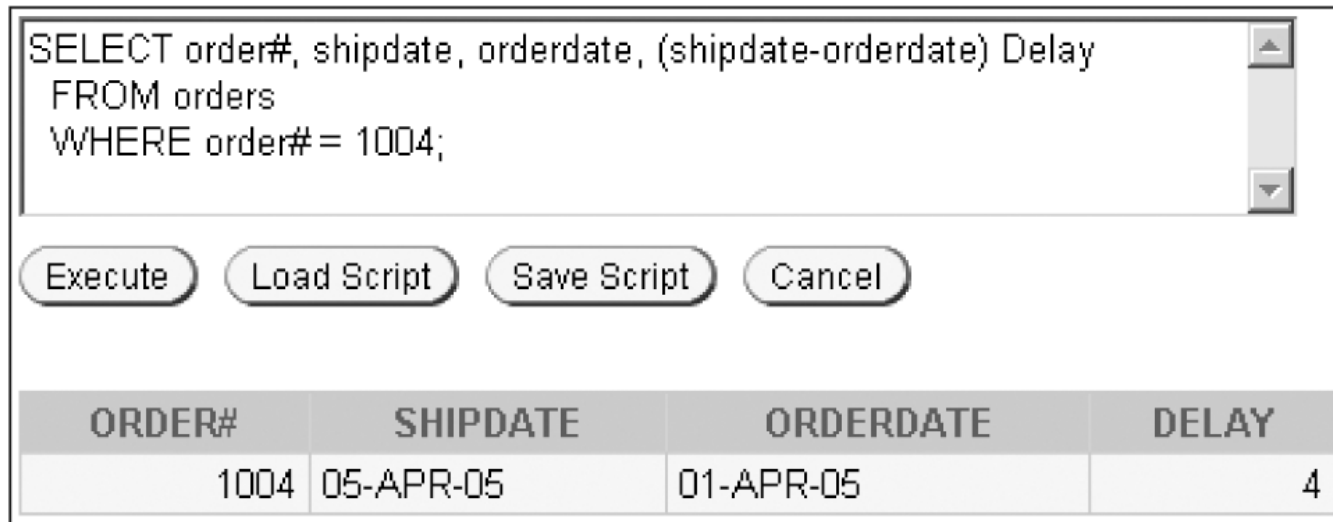
- Allow for manipulation of numeric data:

| Function | Description | Syntax |
|---|---|---|
| ROUND | Round numeric columns to a stated precision | ROUND(n, *decimal place*) |
| TRUNC | Truncate a numeric value to a specific position | TRUNC(n, *decimal place*) |
| MOD | Return the remainder of n divided by m | MOD(n,m) |
| ABS | Return the absolute value of n | ABS(n) |

WAYNE STATE
Mike Ilitch School of Business

# Date Functions

- Used to perform date calculations or format date values

- Subtract date for number of days difference



```
SELECT order#, shipdate, orderdate, (shipdate-orderdate) Delay
  FROM orders
  WHERE order# = 1004;
```

| ORDER# | SHIPDATE | ORDERDATE | DELAY |
|---|---|---|---|
| 1004 | 05-APR-05 | 01-APR-05 | 4 |

# MONTHS_BETWEEN Function

- Determines the number of months between two dates



```
SELECT title, MONTHS_BETWEEN(orderdate,pubdate) Mths
 FROM books JOIN orderitems USING (isbn)
        JOIN orders USING (order#)
 WHERE order# = 1004;
```

[ Execute ]   [ Load Script ]   [ Save Script ]   [ Cancel ]

| TITLE | MTHS |
|---|---|
| PAINLESS CHILD-REARING | 56.483871 |

# ADD_MONTHS Function

- Adds a specified number of months to a date

```
SELECT title, pubdate, ADD_MONTHS(pubdate, 60) "Renegotiate Date",
    ADD_MONTHS(pubdate,84) "Drop Date"
 FROM books
 ORDER BY "Drop Date";
```

( Execute )  ( Load Script )  ( Save Script )  ( Cancel )

| TITLE | PUBDATE | Renegotiate Date | Drop Date |
|---|---|---|---|
| PAINLESS CHILD-REARING | 17-JUL-00 | 17-JUL-05 | 17-JUL-07 |
| THE WOK WAY TO COOK | 11-SEP-00 | 11-SEP-05 | 11-SEP-07 |

**WAYNE STATE**
Mike Ilitch School of Business

# Datatype Conversion

- TO_DATE
  - From string to date
  - TO_DATE(string, date format)

- TO_CHAR
  - From date or number to string
  - TO_CHAR(date or number, string format)

- TO_NUMBER
  - From string to number
  - TO_NUMBER(string)

**WAYNE STATE**
Mike Ilitch School of Business

# TO_DATE Function

- Converts various date formats to the internal format (DD-MON-YY) used by Oracle 10*g*

```
SELECT order#, orderdate, shipdate
  FROM orders
  WHERE orderdate = TO_DATE('March 31, 2005', 'Month DD, YYYY');
```

Execute    Load Script    Save Script    Cancel

| ORDER# | ORDERDATE | SHIPDATE |
|--------|-----------|----------|
| 1000 | 31-MAR-05 | 02-APR-05 |
| 1001 | 31-MAR-05 | 01-APR-05 |
| 1002 | 31-MAR-05 | 01-APR-05 |

# Format Model Elements - Dates

| DATE FORMATS | | |
|---|---|---|
| Element | Description | Example |
| MONTH | Name of the month spelled out—padded with blank spaces to a total width of nine spaces | APRIL |
| MON | Three-letter abbreviation for the name of the month | APR |
| MM | Two-digit numeric value of the month | 04 |
| RM | Roman numeral month | IV |
| D | Numeric value for the day of the week | Wednesday = 4 |
| DD | Numeric value for the day of the month | 28 |
| DDD | Numeric value for the day of the year | December 31 = 365 |
| DAY | Name of the day of the week—padded with blank spaces to a length of nine characters | Wednesday |
| DY | Three-letter abbreviation for the day of the week | WED |
| YYYY | Displays the Four-digit numeric value of the year | 2004 |
| YYY or YY or Y | The last three, two, or single digit(s) of the year | 2004 = 004; 2004 = 04; 2004 = 4 |
| YEAR | Spelled-out version of the year | TWO THOUSAND FOUR |
| B.C. or A.D. | Value indicating B.C. or A.D. | 2004 A.D. |

**FIGURE 10-25** Data format elements

# TO_DATE Function

- March 31, 2005 → 03-31-05

```
SELECT order#, orderdate, shipdate
  FROM orders
  WHERE orderdate = TO_DATE('03-31-05', 'MM-DD-YY')
```

Execute     Load Script     Save Script     Cancel

| ORDER# | ORDERDATE | SHIPDATE |
|--------|-----------|----------|
| 1000 | 31-MAR-05 | 02-APR-05 |
| 1001 | 31-MAR-05 | 01-APR-05 |
| 1002 | 31-MAR-05 | 01-APR-05 |
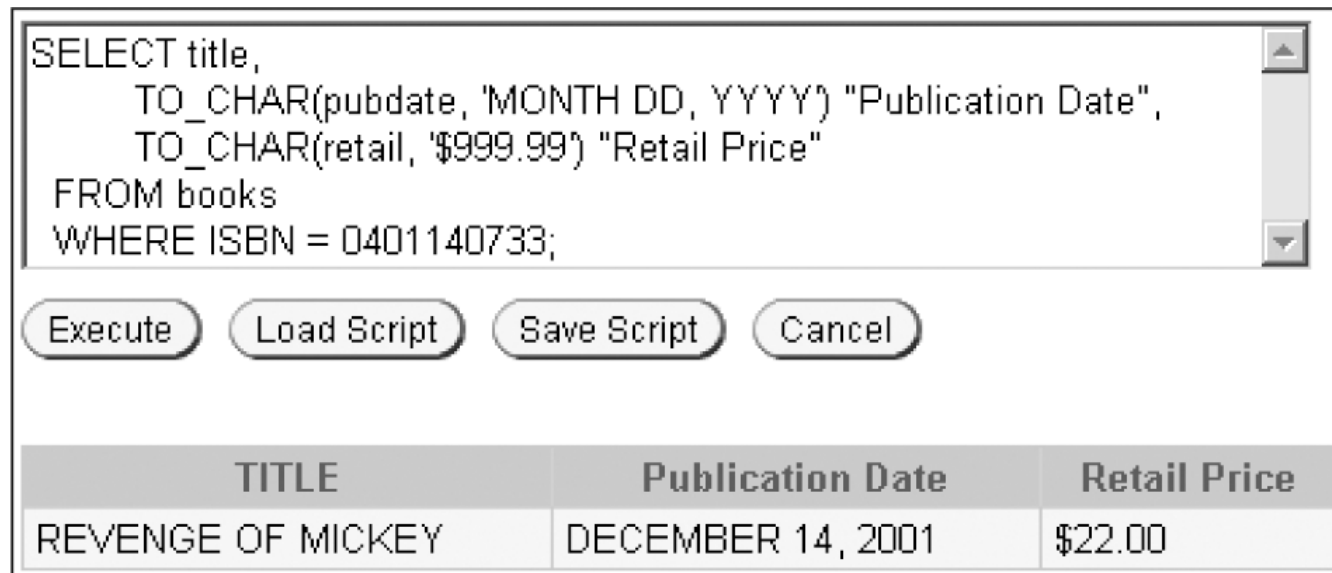
**WAYNE STATE**
Mike Ilitch School of Business

# TO_CHAR Function

- Converts dates and numbers to a formatted character string

# Format Model Elements – Time and Number
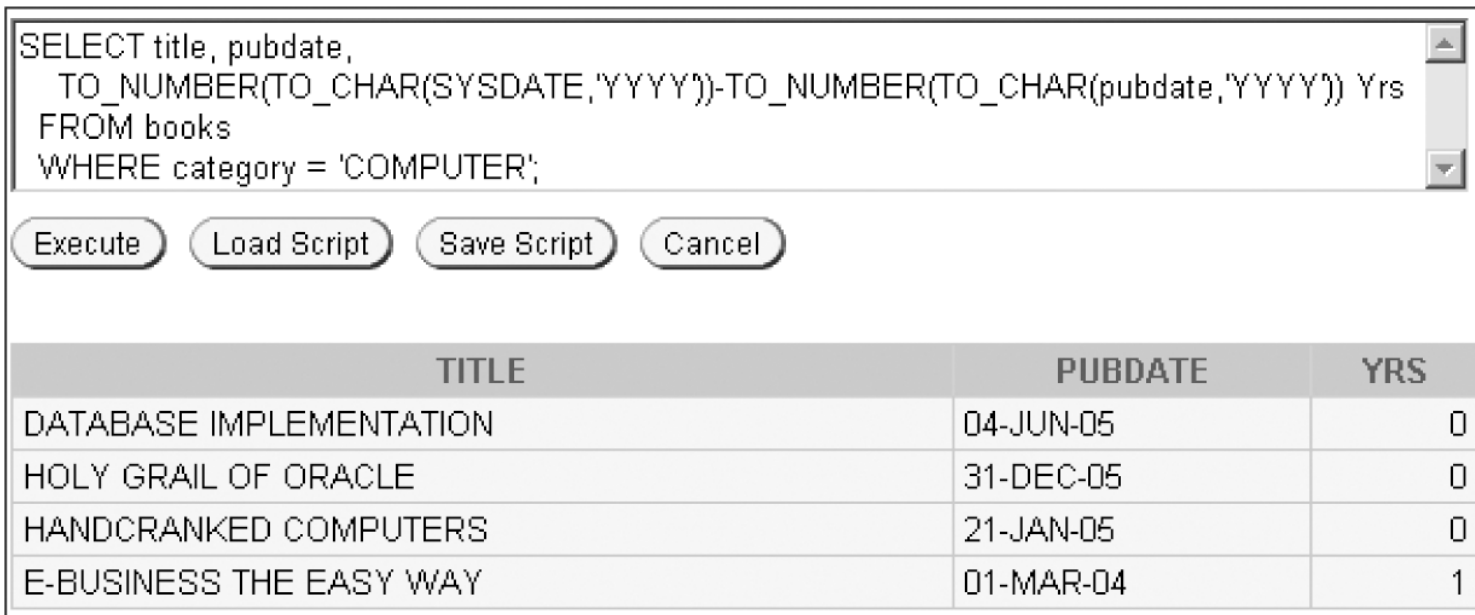
| Time Elements | | |
|---|---|---|
| SS | Seconds | Value between 0–59 |
| SSSS | Seconds past midnight | Value between 0–86399 |
| MI | Minutes | Value between 0–59 |
| HH or HH12 | Hours | Value between 1–12 |
| HH24 | Hours | Value between 0–23 |
| A.M. or P.M. | Value indicating morning or evening hours | A.M. (before noon) or P.M. (after noon) |
| Number Elements | | |
| 9 | Indicates width of display with a series of 9s, but insignificant leading zeros are not displayed | 99999 |
| 0 | Displays insignificant leading zeros | 0009999 |
| $ | Displays a floating dollar sign | $99999 |
| . | Indicates number of decimals to display | 999.99 |
| , | Displays a comma in the position indicated | 9,999 |

**WAYNE STATE**
Mike Ilitch School of Business

# TO_NUMBER Function

```
SELECT title, pubdate,
   TO_NUMBER(TO_CHAR(SYSDATE,'YYYY'))-TO_NUMBER(TO_CHAR(pubdate,'YYYY')) Yrs
FROM books
WHERE category = 'COMPUTER';
```

Execute    Load Script    Save Script    Cancel

| TITLE | PUBDATE | YRS |
|-------|---------|-----|
| DATABASE IMPLEMENTATION | 04-JUN-05 | 0 |
| HOLY GRAIL OF ORACLE | 31-DEC-05 | 0 |
| HANDCRANKED COMPUTERS | 21-JAN-05 | 0 |
| E-BUSINESS THE EASY WAY | 01-MAR-04 | 1 |

**FIGURE 10-41**   Using the TO_NUMBER function to convert a string to a numeric datatype

WAYNE STATE
**Mike Ilitch School of Business**

# NVL Function

- Substitutes a value for a NULL value



```
SELECT order#, orderdate, NVL(shipdate, '07-APR-05'),
       NVL(shipdate, '07-APR-05')-orderdate "Delay"
FROM orders
WHERE order#=1018;
```

Execute   Load Script   Save Script   Cancel

| ORDER# | ORDERDATE | NVL(SHIPDATE,'07-A | Delay |
|--------|-----------|--------------------|-------|
| 1018 | 05-APR-05 | 07-APR-05 | 2 |

**WAYNE STATE**
Mike Ilitch School of Business

# NVL2 Function

- Allows different actions based on whether a value is NULL

```
SELECT order#, orderdate,
        NVL2(shipdate, 'Shipped', 'Not Shipped') "Status"
FROM orders;
```

( Execute )   ( Load Script )   ( Save Script )   ( Cancel )

| ORDER# | ORDERDATE | Status |
|--------|-----------|--------|
| 1000 | 31-MAR-05 | Shipped |
| 1001 | 31-MAR-05 | Shipped |
| 1002 | 31-MAR-05 | Shipped |

**WAYNE STATE**
Mike Ilitch School of Business

# DECODE Function

- Determines action based upon values in a list

```
SELECT customer#, state,
       DECODE(state, 'CA', .08,
                     'FL', .07,
                     0) "Sales Tax Rate"
FROM customers;
```

Execute     Load Script     Save Script     Cancel

| CUSTOMER# | STATE | Sales Tax Rate |
|-----------|-------|----------------|
| 1001 | FL | .07 |
| 1002 | CA | .08 |
| 1003 | FL | .07 |

WAYNE STATE
Mike Ilitch School of Business

# Wrap-up

- Case conversion functions
- Character manipulation functions
- Numeric functions
- Date functions
- Regular expressions
- Other functions

WAYNE STATE
**Mike Ilitch School of Business**