Murach Chapter 3

How to Retrieve Data From a Single Table

Week 2, Lecture 4

Knowledge Points in this lecture

- Special note about DATE type
- ROWNUM pseudo column
- Comparison and logical operators, parentheses in logical expressions
- Expanded Syntax of SELECT clause
- More operators used in WHERE clause:
  - IN, BETWEEN AND, LIKE, IS NULL, IS NOT NULL
- Expanded Syntax of ORDER BY clause
- Nested sort
- Sorting query results by using the following in ORDER BY clause
  - Column alias
  - Expression

Warning about date comparisons

- All DATE data types include both a date and time.
- The value returned by the SYSDATE function includes a date and a time.
- When you code a date literal like '31-May-2014', the time defaults to 00:00:00 (12AM, midnight) on a 24-hour clock.
- If you ignore the times, a date comparison may not yield the results you expect.

Notes

- Chapter 8 provides solutions for this problem.

A SELECT statement that uses the ROWNUM pseudo column to limit the number of rows

```
SELECT vendor_id, invoice_total
FROM invoices
WHERE ROWNUM <= 5
```

	VENDOR_ID	INVOICE_TOTAL
1	34	116.54
2	34	1083.58
3	110	20551.18
4	110	26881.4
5	81	936.93

A SELECT statement that sorts the result set after the WHERE clause

```
SELECT vendor_id, invoice_total
FROM invoices
WHERE ROWNUM <= 5
ORDER BY invoice_total DESC
```

	VENDOR_ID	INVOICE_TOTAL
1	110	26881.4
2	110	20551.18
3	34	1083.58
4	81	936.93
5	34	116.54

- Use ROWNUM together with sorting (ORDER BY)
- Not Top-N type of query

A SELECT statement that sorts the result set before the WHERE clause

```
SELECT vendor_id, invoice_total
FROM (SELECT * FROM invoices
      ORDER BY invoice_total DESC)
WHERE ROWNUM <= 5
```

	VENDOR_ID	INVOICE_TOTAL
1	110	37966.19
2	110	26881.4
3	110	23517.58
4	72	21842
5	110	20551.18

- More practical use of ROWNUM together with sorting (ORDER BY)
- Top-N type of query
- To be covered in more detail in later chapters

The syntax of the WHERE clause with logical operators

```
WHERE [NOT] search_condition_1
    {AND|OR} [NOT] search_condition_2 ...
```

Examples of queries using logical operators

A search condition that uses the AND operator

```
WHERE vendor_state = 'NJ' AND vendor_city = 'Springfield'
```

A search condition that uses the OR operator

```
WHERE vendor_state = 'NJ' OR vendor_city = 'Pittsburgh'
```

A search condition that uses the NOT operator

```
WHERE NOT (invoice_total >= 5000
      OR NOT invoice_date <= '01-JUL-2014')
```

The same condition rephrased to eliminate NOT

```
WHERE invoice_total < 5000
      AND invoice_date <= '01-JUL-2008'
```

A compound condition without parentheses

```
SELECT invoice_number, invoice_date, invoice_total
FROM invoices
WHERE invoice_date > '01-MAY-2014' OR invoice_total > 500
    AND invoice_total - payment_total - credit_total > 0
ORDER BY invoice_number
```

	INVOICE_NUMBER	INVOICE_DATE	INVOICE_TOTAL
1	0-2058	08-MAY-14	37966.19
2	0-2060	08-MAY-14	23517.58
3	0-2436	07-MAY-14	10976.06

```
(91 rows selected)
```

The order of precedence for compound conditions

- NOT
- AND
- OR

The same compound condition with parentheses

```
WHERE (invoice_date > '01-MAY-2014'
       OR invoice_total > 500)
    AND invoice_total - payment_total - credit_total > 0
ORDER BY invoice_number
```

	INVOICE_NUMBER	INVOICE_DATE	INVOICE_TOTAL
1	0-2436	07-MAY-14	10976.06
2	109596	14-JUN-14	41.8
3	111-92R-10092	04-JUN-14	46.21

(39 rows selected)

The expanded syntax of the SELECT clause

```
SELECT [ALL|DISTINCT]
    column_specification [[AS] result_column]
    [, column_specification [[AS] result_column]] ...
```

Five ways to code column specifications

- All columns in base table
- Column name in base table
- Concatenation
- Calculation
- Scalar function

- [ ] : Optional; | : Or ; underlined : default option; comma: separator in a list
- ALL | DISTINCT:
  - keep all rows including duplicates OR only distinct rows in the query result
- Base table – table stored in the database

The syntax of the WHERE clause with the IN operator

```
WHERE test_expression
  [NOT] IN ({subquery|expression_1 [, expression_2]...})
```

Examples of the IN operator

The IN operator with a list of numeric literals

```
WHERE terms_id IN (1, 3, 4)
```

The IN operator preceded by NOT

```
WHERE vendor_state NOT IN ('CA', 'NV', 'OR')
```

The IN operator with a subquery

```
WHERE vendor_id IN
    (SELECT vendor_id
     FROM invoices
     WHERE invoice_date = '01-MAY-2014')
```

The syntax of the WHERE clause with the BETWEEN operator

```
WHERE test_expression
    [NOT] BETWEEN begin_expression AND end_expression
```

Examples of the BETWEEN operator

The BETWEEN operator with literal values

```
WHERE invoice_date
    BETWEEN '01-MAY-2014' AND '31-MAY-2014'
```

The BETWEEN operator preceded by NOT

```
WHERE vendor_zip_code NOT BETWEEN 93600 AND 93799
```

The BETWEEN operator with a calculated value

```
WHERE invoice_total - payment_total - credit_total
    BETWEEN 200 AND 500
```

The BETWEEN operator with upper and lower limits

```
WHERE invoice_due_date BETWEEN SYSDATE AND (SYSDATE + 30)
```

The syntax of the WHERE clause with the LIKE operator

```
WHERE match_expression [NOT] LIKE pattern
```

Wildcard symbols

% - any string of 0 or more characters
_ : any single character

WHERE clauses that use the LIKE operator

Example 1

```
WHERE vendor_city LIKE 'SAN%'
```

Cities that will be retrieved

"San Diego" and "Santa Ana"

Example 2

```
WHERE vendor_name LIKE 'COMPU_ER%'
```

Vendors that will be retrieved

"Compuserve" and "Computerworld"

Note: Characters in the pattern are case sensitive

The syntax of the WHERE clause with the Is null condition

```
WHERE expression IS [NOT] NULL
```

The contents of the Null_Sample table

```
SELECT *
FROM null_sample
```

	INVOICE_ID	INVOICE_TOTAL
1	1	125
2	2	0
3	3	(null)
4	4	2199.99
5	5	0

NULL – not applicable, not available, unknown

A SELECT statement that retrieves rows with zero values

```
SELECT *
FROM null_sample
WHERE invoice_total = 0
```

	INVOICE_ID	INVOICE_TOTAL
1	2	0
2	5	0

A SELECT statement that retrieves rows with non-zero values

```
SELECT *
FROM null_sample
WHERE invoice_total <> 0
```

	INVOICE_ID	INVOICE_TOTAL
1	1	125
2	4	2199.99

A SELECT statement that retrieves rows with null values

```
SELECT *
FROM null_sample
WHERE invoice_total IS NULL
```

	INVOICE_ID	INVOICE_TOTAL
1	3	(null)

A SELECT statement that retrieves rows without null values

```
SELECT *
FROM null_sample
WHERE invoice_total IS NOT NULL
```

	INVOICE_ID	INVOICE_TOTAL
1	1	125
2	2	0
3	4	2199.99
4	5	0

The expanded syntax of the ORDER BY clause

```
ORDER BY expression [ASC|DESC]
     [, expression [ASC|DESC]] ...
```

[A]: A is optional
A|B: A or B; A: A is default

An ORDER BY clause that sorts by one column

```
SELECT vendor_name,
    vendor_city || ', ' || vendor_state || ' ' ||
vendor_zip_code AS address
FROM vendors
ORDER BY vendor_name
```

	VENDOR_NAME	ADDRESS
1	ASC Signs	Fresno, CA 93703
2	AT&T	Phoenix, AZ 85062
3	Abbey Office Furnishings	Fresno, CA 93722

Default: ascending (ASC)

An ORDER BY clause that sorts by one column in descending sequence

```
SELECT vendor_name,
    vendor_city || ', ' || vendor_state || ' ' ||
vendor_zip_code AS address
FROM vendors
ORDER BY vendor_name DESC
```

	VENDOR_NAME	ADDRESS
1	Zylka Design	Fresno, CA 93711
2	Zip Print & Copy Center	Fresno, CA 93777
3	Zee Medical Service Co	Washington, IA 52353

An ORDER BY clause that sorts by three columns

```
SELECT vendor_name,
    vendor_city || ', ' || vendor_state || ' ' ||
        vendor_zip_code AS address
FROM vendors
ORDER BY vendor_state, vendor_city, vendor_name
```

	VENDOR_NAME	ADDRESS
1	AT&T	Phoenix, AZ 85062
2	Computer Library	Phoenix, AZ 85023
3	Wells Fargo Bank	Phoenix, AZ 85038
4	Aztek Label	Anaheim, CA 92807
5	Blue Shield of California	Anaheim, CA 92850
6	Diversified Printing & Pub	Brea, CA 92621
7	ASC Signs	Fresno, CA 93703

Nested Sort example

- ORDER BY vendor_state, vendor_city, vendor_name

An ORDER BY clause that uses a column alias

```
SELECT vendor_name,
    vendor_city || ', ' || vendor_state || ' ' ||
        vendor_zip_code AS address
FROM vendors
ORDER BY address, vendor_name
```

	VENDOR_NAME	ADDRESS
1	Aztek Label	Anaheim, CA 92807
2	Blue Shield of California	Anaheim, CA 92850
3	Malloy Lithographing Inc	Ann Arbor, MI 48106
4	Data Reproductions Corp	Auburn Hills, MI 48326

An ORDER BY clause that uses an expression

```
SELECT vendor_name,
    vendor_city || ', ' || vendor_state || ' ' ||
        vendor_zip_code AS address
FROM vendors
ORDER BY vendor_contact_last_name
    || vendor_contact_first_name
```

	VENDOR_NAME	ADDRESS
1	Dristas Groom & McCormick	Fresno, CA 93720
2	Internal Revenue Service	Fresno, CA 93888
3	US Postal Service	Madison, WI 53707
4	Yale Industrial Trucks-Fresno	Fresno, CA 93706