Database Principles: Fundamentals of Design, Implementation, and Management

Tenth Edition

Chapter 6

Procedural Language SQL and Advanced SQL

Objectives

- In this chapter, you will learn:
  - How to use the advanced SQL JOIN operator syntax
  - About the different types of subqueries and correlated queries
  - How to use SQL functions to manipulate dates, strings, and other data
  - About the relational set operators UNION, UNION ALL, INTERSECT, and MINUS

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Objectives (cont'd.)

- – How to create and use views and updatable views
- – How to create and use triggers and stored procedures
- – How to create embedded SQL

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

SQL Join Operators

- Join operation merges rows from two tables and returns the rows with one of the following:
  - Have common values in common columns
    - Natural join
  - Meet a given join condition
    - Equality or inequality
  - Have common values in common columns or have no matching values
    - Outer join
- Inner join: only returns rows meeting criteria

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Cross Join

- Performs relational product of two tables
  - Also called Cartesian product
- Syntax:

  SELECT column-list FROM table1 CROSS JOIN table2

- Perform a cross join that yields specified attributes

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Natural Join

- Returns all rows with matching values in the matching columns
  - Eliminates duplicate columns
- Used when tables share one or more common attributes with common names
- Syntax:

  SELECT column-list FROM table1 NATURAL JOIN table2

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

JOIN USING Clause

- Returns only rows with matching values in the column indicated in the USING clause
- Syntax:

  SELECT column-list FROM table1 JOIN table2 USING (common-column)

- JOIN USING operand does not require table qualifiers
  - Oracle returns error if table name is specified

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

JOIN ON Clause

- Used when tables have no common attributes
- Returns only rows that meet the join condition
  - Typically includes equality comparison expression of two columns
- Syntax:

  SELECT column-list FROM table1 JOIN table2 ON join-condition

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Outer Joins

- Returns rows matching the join condition
- Also returns rows with unmatched attribute values for tables to be joined
- Three types
  - Left
  - Right
  - Full
- Left and right designate order in which tables are processed

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Outer Joins (cont'd.)

- Left outer join
  - Returns rows matching the join condition
  - Returns rows in left side table with unmatched values
  - Syntax: SELECT column-list FROM table1 LEFT [OUTER] JOIN table2 ON join-condition
- Right outer join
  - Returns rows matching join condition
  - Returns rows in right side table with unmatched values

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

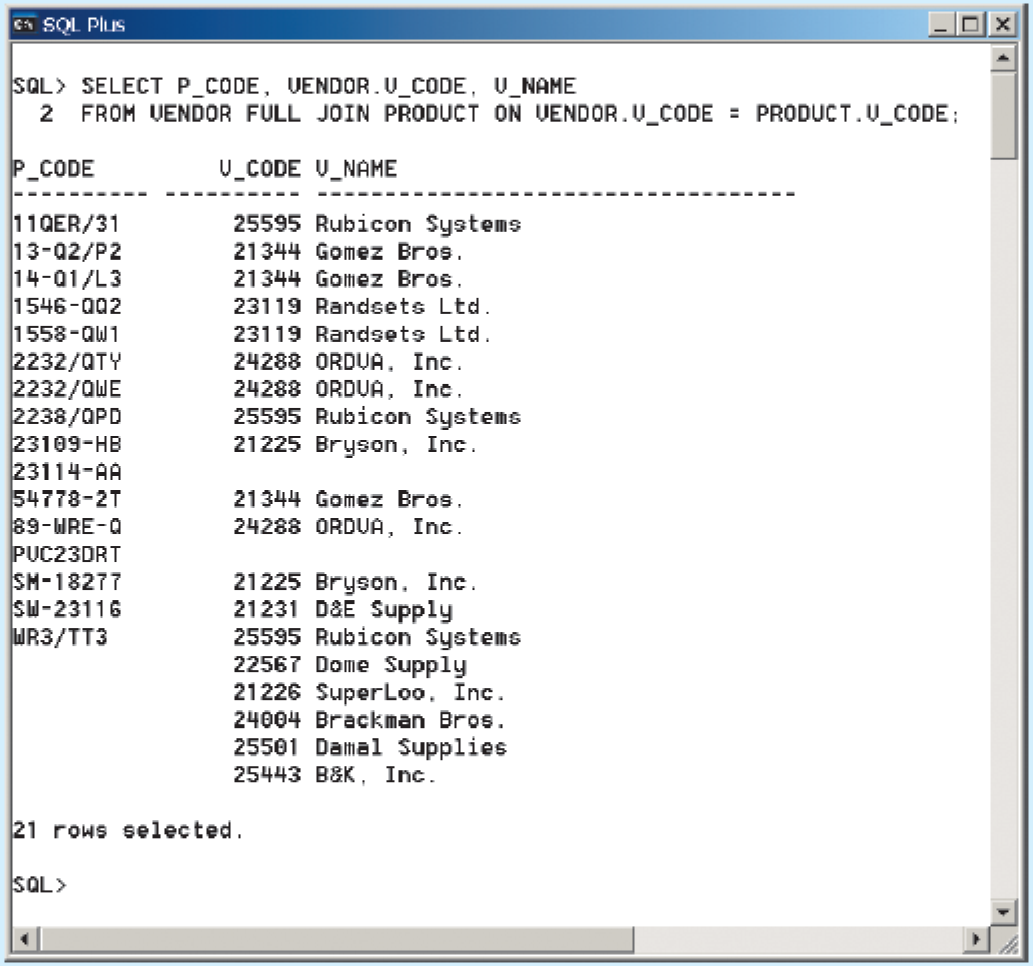
Outer Joins (cont'd.)

- Full outer join
  - Returns rows matching join condition
  - Returns all rows with unmatched values in either side table
  - Syntax:

```
SELECT    column-list
FROM      table1 FULL [OUTER] JOIN table2
          ON join-condition
```

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

FIGURE 6.6 FULL JOIN results



```
SQL> SELECT P_CODE, VENDOR.V_CODE, V_NAME
  2  FROM VENDOR FULL JOIN PRODUCT ON VENDOR.V_CODE = PRODUCT.V_CODE;

P_CODE         V_CODE V_NAME
---------- ---------- -----------------------------------
11QER/31        25595 Rubicon Systems
13-Q2/P2        21344 Gomez Bros.
14-Q1/L3        21344 Gomez Bros.
1546-QQ2        23119 Randsets Ltd.
1558-QW1        23119 Randsets Ltd.
2232/QTY        24288 ORDVA, Inc.
2232/QWE        24288 ORDVA, Inc.
2238/QPD        25595 Rubicon Systems
23109-HB        21225 Bryson, Inc.
23114-AA
54778-2T        21344 Gomez Bros.
89-WRE-Q        24288 ORDVA, Inc.
PVC23DRT
SM-18277        21225 Bryson, Inc.
SW-23116        21231 D&E Supply
WR3/TT3         25595 Rubicon Systems
                22567 Dome Supply
                21226 SuperLoo, Inc.
                24004 Brackman Bros.
                25501 Damal Supplies
                25443 B&K, Inc.

21 rows selected.

SQL>
```

SOURCE: Course Technology/Cengage Learning

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Subqueries and Correlated Queries

- Often necessary to process data based on other processed data
- Subquery is a query inside a query, normally inside parentheses
- First query is the outer query
  - Inside query is the inner query
- Inner query is executed first
- Output of inner query is used as input for outer query
- Sometimes referred to as a nested query

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

WHERE Subqueries

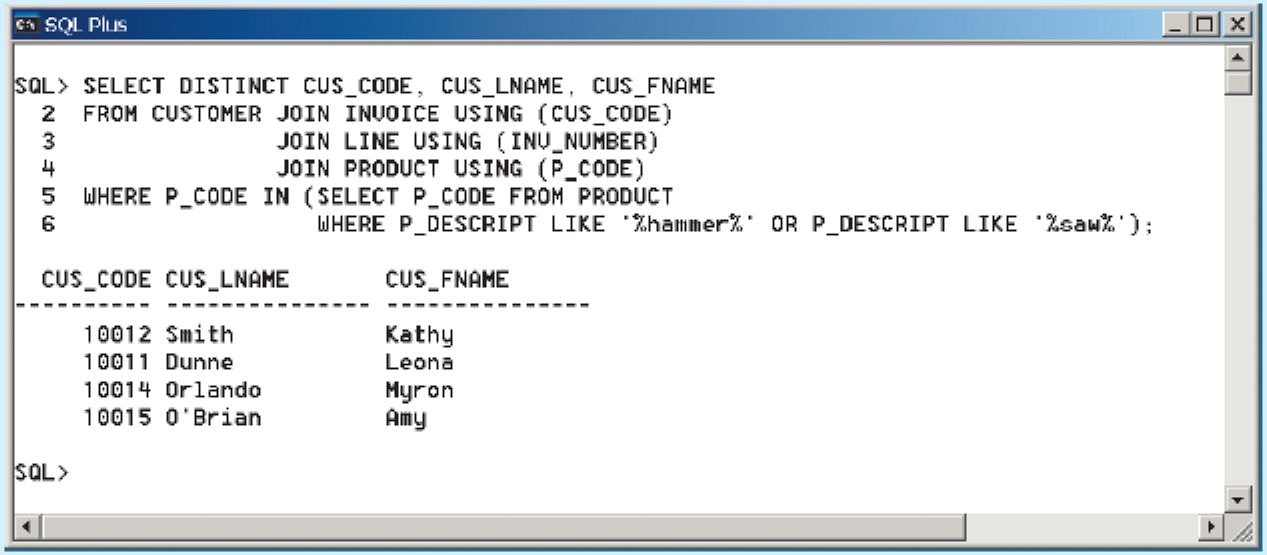
- Most common type uses inner SELECT subquery on right side of WHERE comparison
  - Requires a subquery that returns only one single value
- Value generated by subquery must be of comparable data type
- Can be used in combination with joins

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

IN Subqueries

- Used when comparing a single attribute to a list of values

FIGURE 6.8 IN subquery example



```
SQL> SELECT DISTINCT CUS_CODE, CUS_LNAME, CUS_FNAME
  2  FROM CUSTOMER JOIN INVOICE USING (CUS_CODE)
  3                 JOIN LINE USING (INV_NUMBER)
  4                 JOIN PRODUCT USING (P_CODE)
  5  WHERE P_CODE IN (SELECT P_CODE FROM PRODUCT
  6                    WHERE P_DESCRIPT LIKE '%hammer%' OR P_DESCRIPT LIKE '%saw%');

  CUS_CODE CUS_LNAME       CUS_FNAME
---------- --------------- ---------------
     10012 Smith           Kathy
     10011 Dunne           Leona
     10014 Orlando         Myron
     10015 O'Brian         Amy

SQL>
```

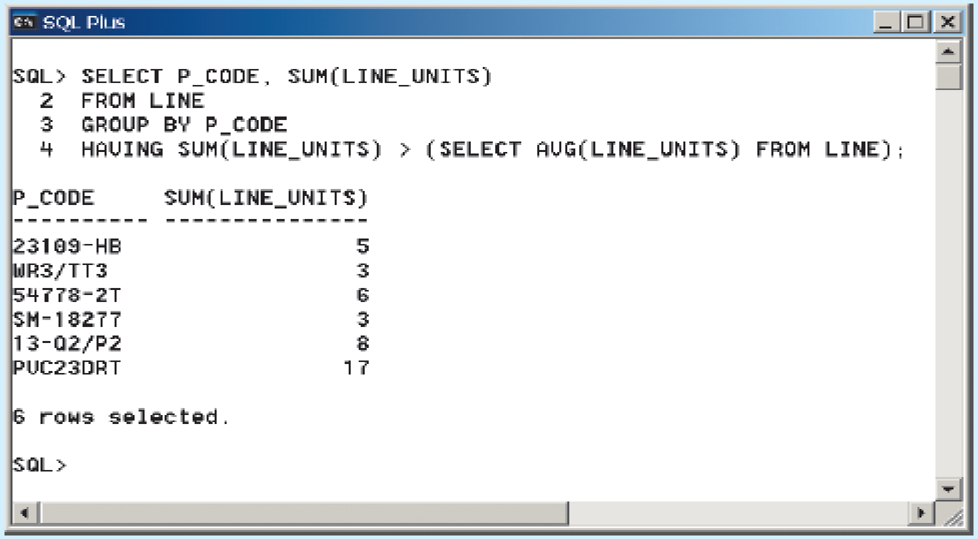
SOURCE: Course Technology/Cengage Learning

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

HAVING Subqueries

- HAVING clause restricts the output of a GROUP BY query
  – Applies conditional criterion to the grouped rows

FIGURE 6.9 HAVING subquery example



```
SQL> SELECT P_CODE, SUM(LINE_UNITS)
  2  FROM LINE
  3  GROUP BY P_CODE
  4  HAVING SUM(LINE_UNITS) > (SELECT AVG(LINE_UNITS) FROM LINE);

P_CODE     SUM(LINE_UNITS)
---------- ---------------
23109-HB                 5
WR3/TT3                  3
54778-2T                 6
SM-18277                 3
13-Q2/P2                 8
PVC23DRT                17

6 rows selected.

SQL>
```

SOURCE: Course Technology/Cengage Learning

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Multirow Subquery Operators: ANY and ALL

- Allows comparison of single value with a list of values using inequality comparison
- “Greater than ALL” equivalent to “greater than the highest in list”
- “Less than ALL” equivalent to “less than lowest”
- Using equal to ANY operator equivalent to IN operator

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

FROM Subqueries

- Specifies the tables from which the data will be drawn
- Can use SELECT subquery in the FROM clause
  - View name can be used anywhere a table is expected

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Attribute List Subqueries

- SELECT statement uses attribute list to indicate columns to project resulting set
  - Columns can be attributes of base tables
  - Result of aggregate function
- Attribute list can also include subquery expression: inline subquery
  - Must return one single value
- Cannot use an alias in the attribute list

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Correlated Subqueries

- Subquery that executes once for each row in the outer query
- Correlated because inner query is related to the outer query
  - Inner query references column of outer subquery
- Can also be used with the EXISTS special operator

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

SQL Functions

- Generating information from data often requires many data manipulations
- SQL functions are similar to functions in programming languages
- Functions always use numerical, date, or string value
- Value may be part of a command or attribute in a table
- Function may appear anywhere in an SQL statement

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Date and Time Functions

- All SQL-standard DBMSs support date and time functions
- Date functions take one parameter and return a value
- Date/time data types are implemented differently by different DBMS vendors
- ANSI SQL standard defines date data types, but not how data types are stored

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Numeric Functions

- Grouped in different ways
  - Algebraic, trigonometric, logarithmic, etc.
- Do not confuse with aggregate functions
  - Aggregate functions operate over sets
  - Numeric functions operate over single row
- Numeric functions take one numeric parameter and return one value

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

TABLE 6.5 Selected Numeric Functions

FUNCTION	EXAMPLE(S)

ABS
Returns the absolute value of a number
Syntax:
ABS(numeric_value)
	In Oracle, use the following:
	SELECT 1.95, -1.93, ABS(1.95), ABS(-1.93)
	FROM DUAL;
	In MS Access and SQL Server, use the following:
	SELECT 1.95, -1.93, ABS(1.95), ABS(-1.93);

ROUND Rounds a value to a specified precision (number of digits) Syntax: ROUND(numeric_value, p) p = precision	Lists the product prices rounded to one and zero decimal places: SELECT P_CODE, P_PRICE, ROUND(P_PRICE,1) AS PRICE1, ROUND(P_PRICE,0) AS PRICE0 FROM PRODUCT;

CEIL/CEILING/FLOOR
Returns the smallest integer greater than or equal to a number or returns the largest integer equal to or less than a number, respectively
Syntax:
CEIL(numeric_value) Oracle
CEILING(numeric_value) SQL Server
FLOOR(numeric_value)
	Lists the product price, the smallest integer greater than or equal to the product price, and the largest integer equal to or less than the product price.
	In Oracle, use the following:
	SELECT P_PRICE, CEIL(P_PRICE), FLOOR(P_PRICE)
	FROM PRODUCT;
	In SQL Server, use the following:
	SELECT P_PRICE, CEILING(P_PRICE), FLOOR(P_PRICE)
	FROM PRODUCT;
	MS Access does not support these functions.

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

String Functions

- String manipulations are the most used functions in programming
- String manipulation function examples:
  - Concatenation
  - Printing in uppercase
  - Finding length of an attribute

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Conversion Functions

- Take a value of given data type and convert it to the equivalent value in another data type
- Oracle conversion functions:
  - TO_CHAR: takes a date value, converts to character string
  - TO_DATE: takes character string representing a date, converts it to actual date in Oracle format
- SQL Server uses CAST and CONVERT functions

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Relational Set Operators

- UNION
- INTERSECT
- MINUS
- Work properly if relations are union-compatible
  - Names of relation attributes must be the same and their data types must be identical

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

UNION

- Combines rows from two or more queries without including duplicate rows
  - Example:

```
SELECT      CUS_LNAME, CUS_FNAME,
            CUS_INITIAL, CUS_AREACODE,
FROM        CUSTOMER
UNION
SELECT      CUS_LNAME, CUS_FNAME,
            CUS_INITIAL, CUS_AREACODE,
FROM        CUSTOMER_2
```

- Can be used to unite more than two queries

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

UNION ALL

- Produces a relation that retains duplicate rows
  - Example query:

```
SELECT     CUS_LNAME, CUS_FNAME,
           CUS_INITIAL, CUS_AREACODE,
FROM       CUSTOMER
UNION ALL
SELECT     CUS_LNAME, CUS_FNAME,
           CUS_INITIAL, CUS_AREACODE,
FROM       CUSTOMER_2;
```

- Can be used to unite more than two queries

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

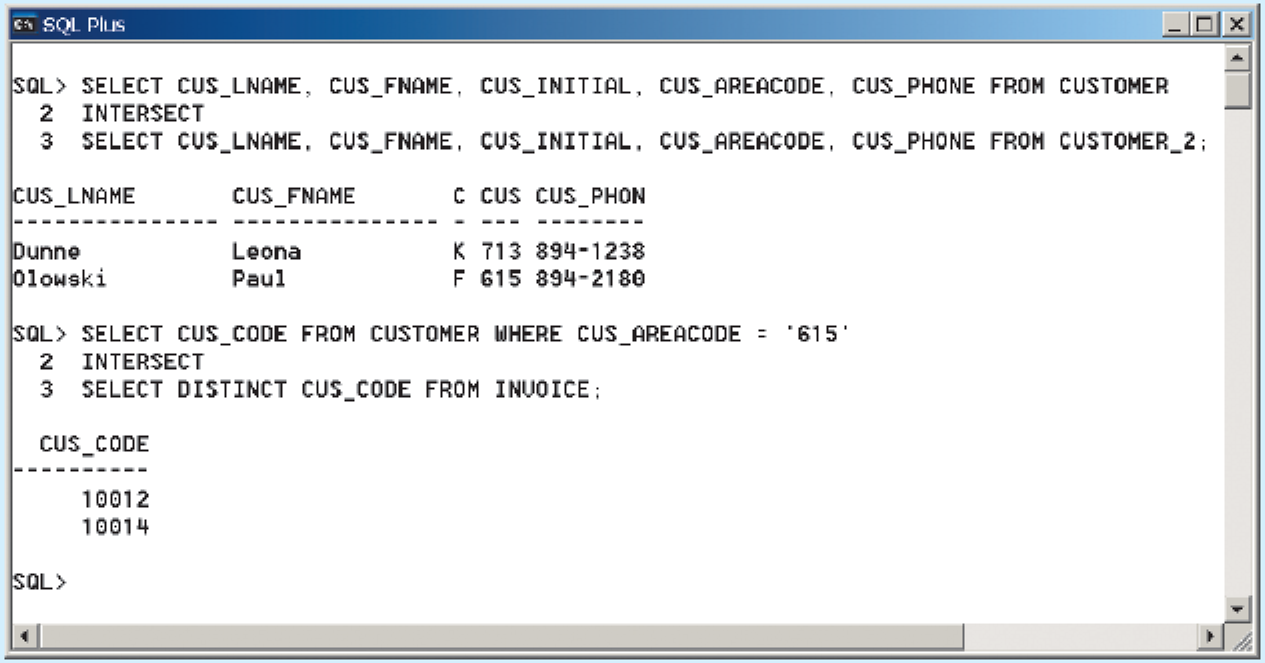
INTERSECT

- Combines rows from two queries, returning only the rows that appear in both sets
- Syntax: query INTERSECT query
  - Example query:

```
SELECT     CUS_LNAME, CUS_FNAME,
           CUS_INITIAL, CUS_AREACODE,
FROM       CUSTOMER
INTERSECT
SELECT     CUS_LNAME, CUS_FNAME,
           CUS_INITIAL, CUS_AREACODE,
FROM       CUSTOMER_2
```

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

FIGURE 6.18 INTERSECT query results



```
SQL> SELECT CUS_LNAME, CUS_FNAME, CUS_INITIAL, CUS_AREACODE, CUS_PHONE FROM CUSTOMER
  2  INTERSECT
  3  SELECT CUS_LNAME, CUS_FNAME, CUS_INITIAL, CUS_AREACODE, CUS_PHONE FROM CUSTOMER_2;

CUS_LNAME       CUS_FNAME       C CUS CUS_PHON
--------------- --------------- - --- --------
Dunne           Leona           K 713 894-1238
Olowski         Paul            F 615 894-2180

SQL> SELECT CUS_CODE FROM CUSTOMER WHERE CUS_AREACODE = '615'
  2  INTERSECT
  3  SELECT DISTINCT CUS_CODE FROM INVOICE;

  CUS_CODE
----------
     10012
     10014

SQL>
```

SOURCE: Course Technology/Cengage Learning

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

MINUS

- Combines rows from two queries
  - Returns only the rows that appear in the first set but not in the second
- Syntax: query MINUS query
  - Example:

```
SELECT      CUS_LNAME, CUS_FNAME,
            CUS_INITIAL, CUS_AREACODE,
FROM        CUSTOMER
MINUS
SELECT      CUS_LNAME, CUS_FNAME,
            CUS_INITIAL, CUS_AREACODE,
FROM        CUSTOMER_2
```

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Syntax Alternatives

- IN and NOT IN subqueries can be used in place of INTERSECT
- Example:

```
SELECT          CUS_CODE FROM CUSTOMER
WHERE           CUS_AREACODE = ‘615’ AND
                CUS_CODE IN (SELECT
                DISTINCT CUS_CODE
                FROM
                INVOICE);
```

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Virtual Tables: Creating a View

- View
  - Virtual table based on a SELECT query
- Base tables
- Tables on which the view is based
- CREATE VIEW viewname AS SELECT query
- Relational view special characteristics

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Updatable Views

- Batch update routine pools multiple transactions into a single batch
  - Update master table field in a single operation
- Updatable view is a view that can be used to update attributes in the base tables
- Not all views are updatable
  - GROUP BY expressions or aggregate functions cannot be used
  - Cannot use set operators
  - Most restrictions are based on use of JOINs

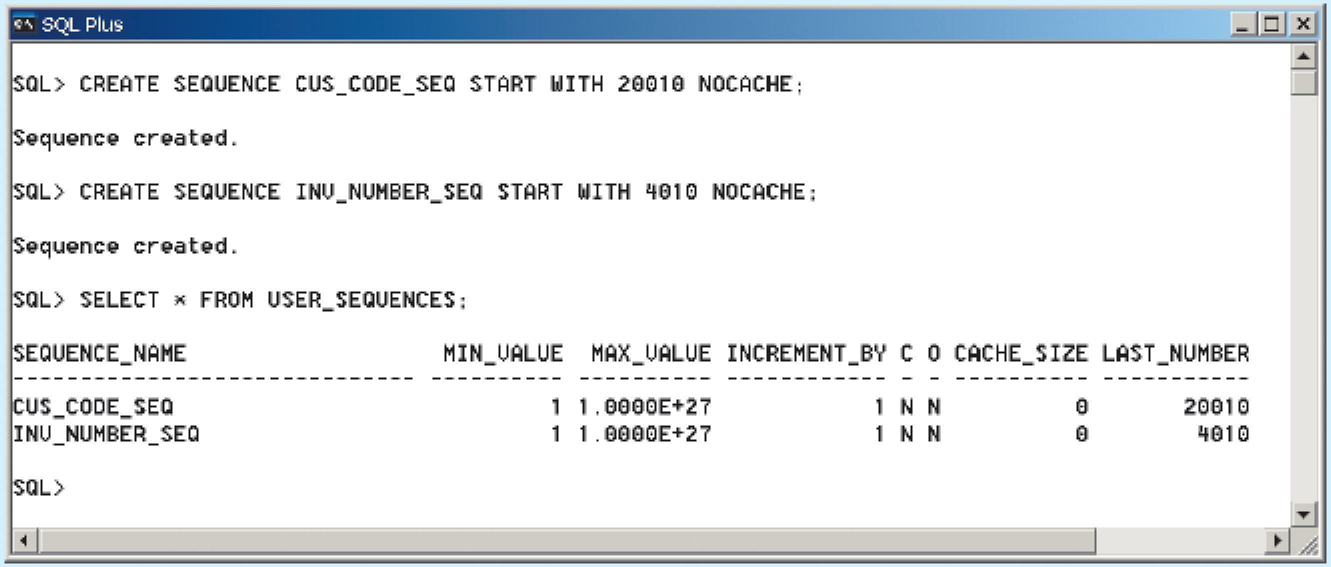
© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Oracle Sequences

- MS Access AutoNumber data type fills a column with unique numeric values
- Oracle sequences
  - Independent object in the database
  - Named, used anywhere a value expected
  - Not tied to a table or column
  - Generate numeric values that can be assigned to any column in any table
  - Created and deleted at any time

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

FIGURE 6.27 Oracle sequence



```
SQL> CREATE SEQUENCE CUS_CODE_SEQ START WITH 20010 NOCACHE;

Sequence created.

SQL> CREATE SEQUENCE INV_NUMBER_SEQ START WITH 4010 NOCACHE;

Sequence created.

SQL> SELECT * FROM USER_SEQUENCES;

SEQUENCE_NAME                   MIN_VALUE  MAX_VALUE INCREMENT_BY C O CACHE_SIZE LAST_NUMBER
------------------------------ ---------- ---------- ------------ - - ---------- -----------
CUS_CODE_SEQ                            1 1.0000E+27            1 N N          0       20010
INV_NUMBER_SEQ                          1 1.0000E+27            1 N N          0        4010

SQL>
```

SOURCE: Course Technology/Cengage Learning

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Procedural SQL

- SQL does not support conditional execution
- Isolate critical code
  - All applications access shared code
  - Better maintenance and logic control
- Persistent stored module (PSM) is a block of code containing:
  - Standard SQL statements
  - Procedural extensions
  - Stored and executed at the DBMS server

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Procedural SQL (cont'd.)

- Procedural SQL (PL/SQL) enables you to:
  - Store procedural code and SQL statements in database
  - Merge SQL and traditional programming constructs
- Procedural code executed by DBMS when invoked by end user
  - Anonymous PL/SQL blocks and triggers
  - Stored procedures and PL/SQL functions

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

TABLE 6.8 PL/SQL Basic Data Types

DATA TYPE	DESCRIPTION
CHAR	Character values of a fixed length; for example: W_ZIP CHAR(5)
VARCHAR2	Variable-length character values; for example: W_FNAME VARCHAR2(15)
NUMBER	Numeric values; for example: W_PRICE NUMBER(6,2)
DATE	Date values; for example: W_EMP_DOB DATE
%TYPE	Inherits the data type from a variable that you declared previously or from an attribute of a database table; for example: W_PRICE PRODUCT.P_PRICE%TYPE Assigns W_PRICE the same data type as the P_PRICE column in the PRODUCT table

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Triggers

- Procedural SQL code automatically invoked by RDBMS on data manipulation event
- Trigger definition:
  - Triggering timing: BEFORE or AFTER
  - Triggering event: INSERT, UPDATE, DELETE
  - Triggering level:
    - Statement-level trigger
    - Row-level trigger
  - Triggering action
- DROP TRIGGER trigger_name

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Stored Procedures

- Named collection of procedural and SQL statements
- Advantages
  - Substantially reduce network traffic and increase performance
    - No transmission of individual SQL statements over network
  - Reduce code duplication by means of code isolation and code sharing
    - Minimize chance of errors and cost of application development and maintenance

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

PL/SQL Processing with Cursors

- Cursor: special construct in procedural SQL to hold data rows returned by SQL query
- Implicit cursor: automatically created when SQL returns only one value
- Explicit cursor: holds the output of an SQL statement that may return two or more rows
- Cursor-style processor retrieves data from cursor one row at a time
  - – Current row is copied to PL/SQL variables

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

PL/SQL Stored Functions

- Named group of procedural and SQL statements that returns a value
- Syntax:

```
CREATE FUNCTION function_name(argument
IN data-type, …) RETURN data-type [IS]
BEGIN
        PL/SQL statements;
        …
        RETURN (value or expression);
END;
```

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Embedded SQL

- Key differences between SQL and procedural languages:
  - Run-time mismatch
    - SQL is executed one instruction at a time
    - Host language typically runs at client side in its own memory space
  - Processing mismatch
    - Host language processes one data element at a time
  - Data type mismatch

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Embedded SQL (cont'd.)

- Embedded SQL framework defines:
  - Standard syntax to identify embedded SQL code within host language
  - Standard syntax to identify host variables
  - Communication area exchanges status and error information between SQL and host language

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

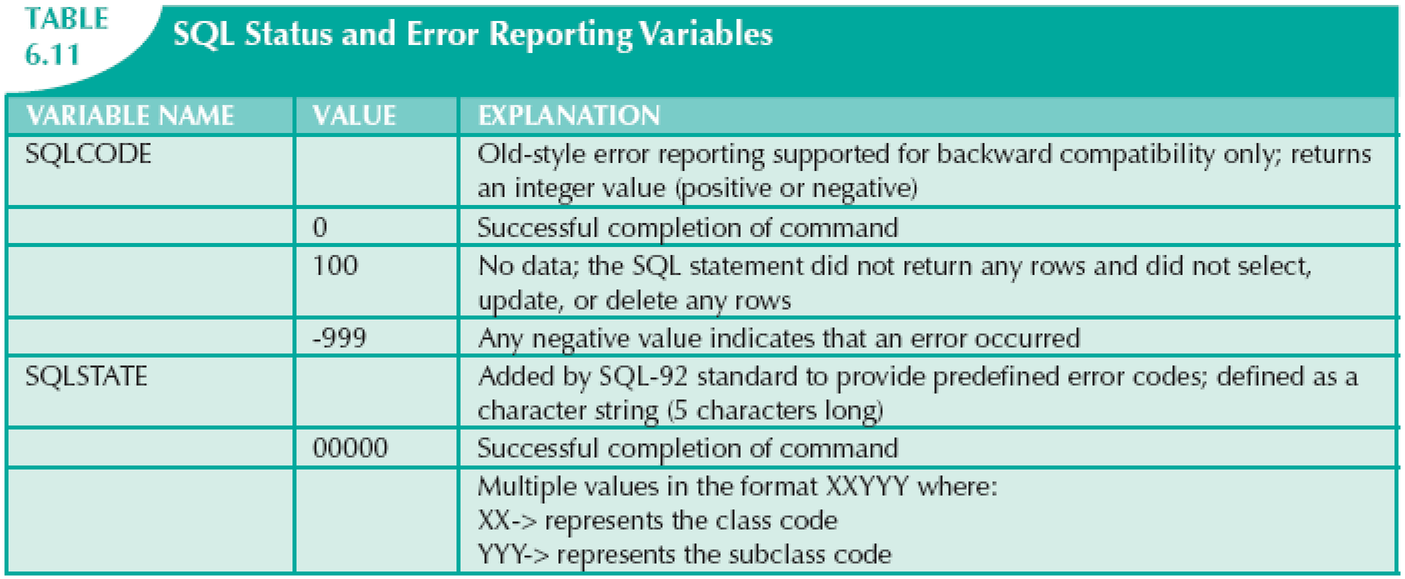


TABLE 6.11 SQL Status and Error Reporting Variables

VARIABLE NAME	VALUE	EXPLANATION
SQLCODE		Old-style error reporting supported for backward compatibility only; returns an integer value (positive or negative)
	0	Successful completion of command
	100	No data; the SQL statement did not return any rows and did not select, update, or delete any rows
	-999	Any negative value indicates that an error occurred
SQLSTATE		Added by SQL-92 standard to provide predefined error codes; defined as a character string (5 characters long)
	00000	Successful completion of command
		Multiple values in the format XXYYY where: XX-> represents the class code YYY-> represents the subclass code

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Embedded SQL (cont'd.)

- Static SQL
  - Embedded SQL in which programmer uses predefined SQL statements and parameters
    - End users of programs are limited to actions that were specified in application programs
  - SQL statements will not change while application is running

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Embedded SQL (cont'd.)

- Dynamic SQL
  - SQL statement is not known in advance, but instead is generated at run time
  - Program can generate SQL statements at run-time that are required to respond to ad hoc queries
  - Attribute list and condition are not known until end user specifies them
  - Tends to be much slower than static SQL
  - Requires more computer resources

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Summary

- Operations that join tables are classified as inner joins and outer joins
- Natural join returns all rows with matching values in the matching columns
  - Eliminates duplicate columns
- Subqueries and correlated queries process data based on other processed data
- Most subqueries are executed in serial fashion
- SQL functions are used to extract or transform data

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Summary (cont'd.)

- Relational set operators combine output of two queries to generate new relation
- Oracle sequences may be used to generate values to be assigned to a record
- PL/SQL can be used to create triggers, stored procedures, and PL/SQL functions
- A stored procedure is a named collection of SQL statements

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Summary (cont'd.)

- When SQL statements return more than one value inside the PL/SQL code, cursor is needed
- Embedded SQL uses SQL statements within an application programming language

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.