11e

Chapter 7

Introduction to Structured Query Language (SQL)

Learning Objectives

- In this chapter, you will learn:
  - The basic commands and functions of SQL
  - How to use SQL for data administration (to create tables and indexes)
  - How to use SQL for data manipulation (to add, modify, delete, and retrieve data)
  - How to use SQL to query a database for useful information

Structured Query Language (SQL)

- Categories of SQL function
  - Data definition language (DDL)
  - Data manipulation language (DML)
- Nonprocedural language with basic command vocabulary set of less than 100 words
- Differences in SQL dialects are minor

Table 7.1 - SQL Data Definition Command

COMMAND OR OPTION	DESCRIPTION
CREATE SCHEMA AUTHORIZATION	Creates a database schema
CREATE TABLE	Creates a new table in the user's database schema
NOT NULL	Ensures that a column will not have null values
UNIQUE	Ensures that a column will not have duplicate values
PRIMARY KEY	Defines a primary key for a table
FOREIGN KEY	Defines a foreign key for a table
DEFAULT	Defines a default value for a column (when no value is given)
CHECK	Validates data in an attribute
CREATE INDEX	Creates an index for a table
CREATE VIEW	Creates a dynamic subset of rows and columns from one or more tables (see Chapter 8, Advanced SQL)
ALTER TABLE	Modifies a table's definition (adds, modifies, or deletes attributes or constraints)
CREATE TABLE AS	Creates a new table based on a query in the user's database schema
DROP TABLE	Permanently deletes a table (and its data)
DROP INDEX	Permanently deletes an index
DROP VIEW	Permanently deletes a view

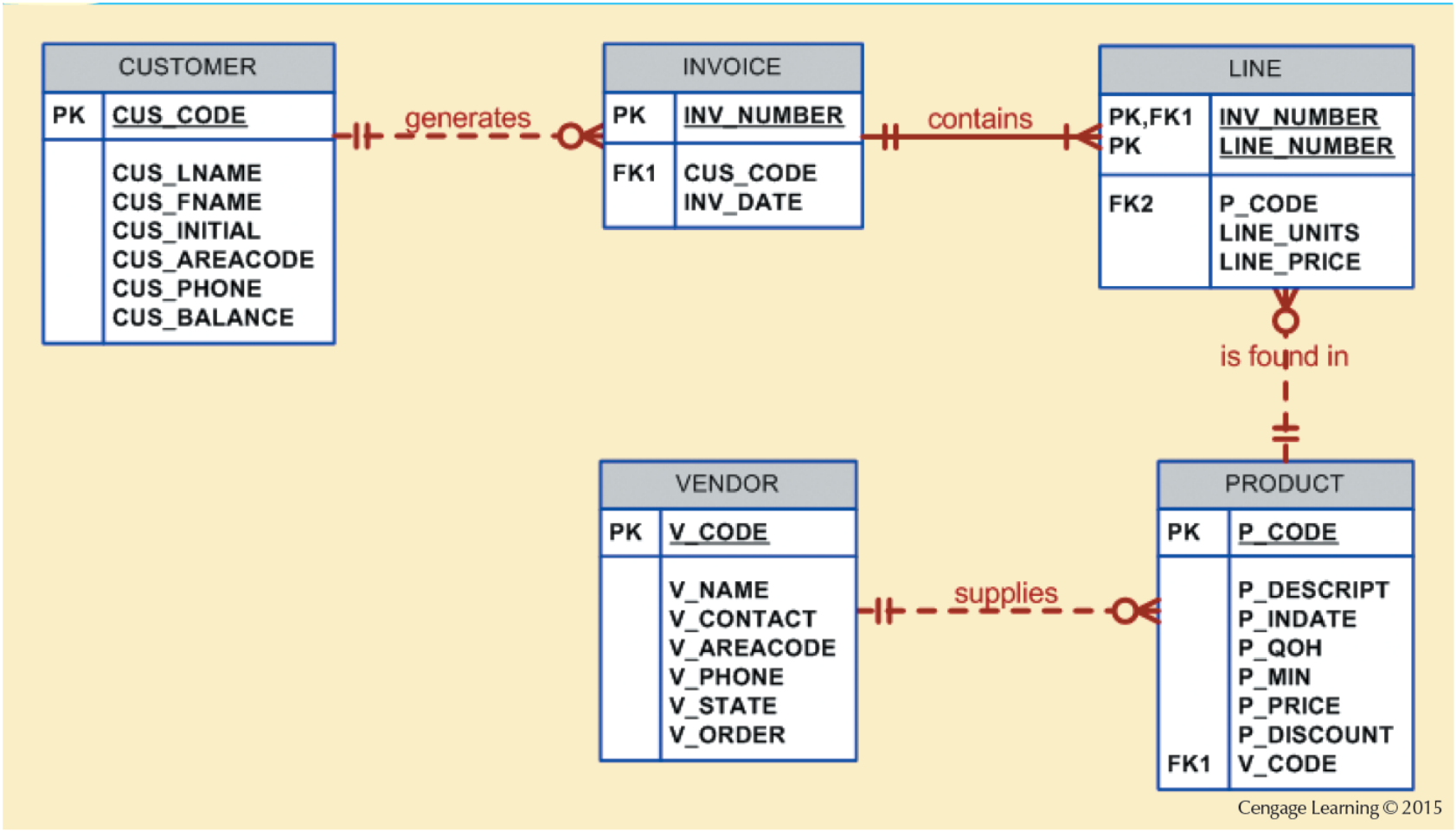
Cengage Learning © 2015

Table 7.2 - SQL Data Manipulation Commands

COMMAND OR OPTION	DESCRIPTION
INSERT	Inserts row(s) into a table
SELECT	Selects attributes from rows in one or more tables or views
WHERE	Restricts the selection of rows based on a conditional expression
GROUP BY	Groups the selected rows based on one or more attributes
HAVING	Restricts the selection of grouped rows based on a condition
ORDER BY	Orders the selected rows based on one or more attributes
UPDATE	Modifies an attribute's values in one or more table's rows
DELETE	Deletes one or more rows from a table
COMMIT	Permanently saves data changes
ROLLBACK	Restores data to their original values
Comparison operators	
=, <, >, <=, >=, <>	Used in conditional expressions
Logical operators	
AND/OR/NOT	Used in conditional expressions
Special operators	Used in conditional expressions
BETWEEN	Checks whether an attribute value is within a range
IS NULL	Checks whether an attribute value is null
LIKE	Checks whether an attribute value matches a given string pattern
IN	Checks whether an attribute value matches any value within a value list
EXISTS	Checks whether a subquery returns any rows
DISTINCT	Limits values to unique values
Aggregate functions	Used with SELECT to return mathematical summaries on columns
COUNT	Returns the number of rows with non-null values for a given column
MIN	Returns the minimum attribute value found in a given column
MAX	Returns the maximum attribute value found in a given column
SUM	Returns the sum of all values for a given column
AVG	Returns the average of all values for a given column

Cengage Learning © 2015

Figure 7.1 - The Database Model



Tasks to be Completed Before Using a New RDBMS

- Create database structure
  - RDBMS creates physical files that will hold database
  - Differs from one RDBMS to another
- **Authentication**: Process DBMS uses to verify that only registered users access the data
  - Required for the creation tables
  - User should log on to RDBMS using user ID and password created by database administrator

The Database Schema

- Logical group of database objects related to each other
- Command
  - CREATE SCHEMA AUTHORIZATION {creator};
    - Seldom used directly

Common SQL Data Types

Numeric

- NUMBER(L,D) or NUMERIC(L,D)

Character

- CHAR(L)
- VARCHAR(L) or VARCHAR2(L)

Date

- DATE

Creating Table Structures

- Use one line per column (attribute) definition
- Use spaces to line up attribute characteristics and constraints
- Table and attribute names are capitalized
- Features of table creating command sequence
  - NOT NULL specification
  - UNIQUE specification
- Syntax to create table
  - CREATE TABLE tablename();

Primary Key and Foreign Key

- Primary key attributes contain both a NOT NULL and a UNIQUE specification
- RDBMS will automatically enforce referential integrity for foreign keys
- Command sequence ends with semicolon
- ANSI SQL allows use of following clauses to cover CASCADE, SET NULL, or SET DEFAULT
  - ON DELETE and ON UPDATE

SQL Constraints

NOT NULL

- Ensures that column does not accept nulls

UNIQUE

- Ensures that all values in column are unique

DEFAULT

- Assigns value to attribute when a new row is added to table

CHECK

- Validates data when attribute value is entered

SQL Indexes

- When primary key is declared, DBMS automatically creates unique index
- Composite index:
  - Is based on two or more attributes
  - Prevents data duplication
- Syntax to create SQL indexes
  - **CREATE INDEX** indexname ON tablename();
- Syntax to delete an index
  - **DROP INDEX** indexname;

Data Manipulation Commands

INSERT: Command to insert data into table

- Syntax - INSERT INTO tablename VALUES();
- Used to add table rows with NULL and NOT NULL attributes

COMMIT: Command to save changes

- Syntax - COMMIT [WORK];
- Ensures database update integrity

Data Manipulation Commands

SELECT: Command to list the contents

- Syntax - SELECT *columnlist* FROM *tablename*;
- **Wildcard character**(*): Substitute for other characters/command

UPDATE: Command to modify data

- Syntax - UPDATE *tablename* SET *columnname* = expression [, *columnname* = *expression*] [WHERE *conditionlist*];

Data Manipulation Commands

WHERE condition

- Specifies the rows to be selected

ROLLBACK: Command to restore the database

- Syntax - ROLLBACK;
- Undoes the changes since last COMMIT command

DELETE: Command to delete

- Syntax - DELETE FROM *tablename*
  - [WHERE *conditionlist*];

Inserting Table Rows with a SELECT Subquery

- Syntax
  - INSERT INTO *tablename* SELECT *columnlist* FROM *tablename*
- Used to add multiple rows using another table as source
- SELECT command - Acts as a subquery and is executed first
  - **Subquery**: Query embedded/nested inside another query

Selecting Rows Using Conditional Restrictions

- Following syntax enables to specify which rows to select
  - SELECT *columnlist*
  - FROM *tablelist*
  - [WHERE *conditionlist*];
- Used to select partial table contents by placing restrictions on the rows
- Optional WHERE clause
  - Adds conditional restrictions to the SELECT statement

Comparison Operators

- Add conditional restrictions on selected table contents
- Used on:
  - Character attributes
  - Dates

Table 7.6 - Comparison Operators

SYMBOL	MEANING
=	Equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
<> or !=	Not equal to

Cengage Learning © 2015

Comparison Operators: Computed Columns and Column Aliases

- SQL accepts any valid expressions/formulas in the computed columns
- **Alias**: Alternate name given to a column or table in any SQL statement to improve the readability
- Computed column, an alias, and date arithmetic can be used in a single query

Arithmetic operators

- **The Rule of Precedence**: Establish the order in which computations are completed
- Perform:
  - Operations within parentheses
  - Power operations
  - Multiplications and divisions
  - Additions and subtractions

Table 7.7 - The Arithmetic Operators

ARITHMETIC OPERATOR	DESCRIPTION
+	Add
-	Subtract
*	Multiply
/	Divide
^	Raise to the power of (some applications use ** instead of ^)

Cengage Learning © 2015

Figure 7.12 - Selected PRODUCT Table Attributes: The logical OR

P_DESCRIPT	P_INDATE	P_PRICE	V_CODE
7.25-in. pwr. saw blade	13-Dec-13	14.99	21344
9.00-in. pwr. saw blade	13-Nov-13	17.49	21344
B&D jigsaw, 12-in. blade	30-Dec-13	109.92	24288
B&D jigsaw, 8-in. blade	24-Dec-13	99.87	24288
Rat-tail file, 1/8-in. fine	15-Dec-13	4.99	21344
Hicut chain saw, 16 in.	07-Feb-14	256.99	24288

Cengage Learning © 2015

Figure 7.13 - Selected PRODUCT Table Attributes: The Logical AND

P_DESCRIPT	P_INDATE	P_PRICE	V_CODE
B&D cordless drill, 1/2-in.	20-Jan-14	38.95	25595
Claw hammer	20-Jan-14	9.95	21225
PVC pipe, 3.5-in., 8-ft	20-Feb-14	5.87	
1.25-in. metal screw, 25	01-Mar-14	6.99	21225
2.5-in. wd. screw, 50	24-Feb-14	8.45	21231

Cengage Learning © 2015

Figure 7.14 - Selected PRODUCT Table Attributes: The Logical AND and OR

P_DESCRIPT	P_INDATE	P_PRICE	V_CODE
B&D jigsaw, 12-in. blade	30-Dec-13	109.92	24288
B&D jigsaw, 8-in. blade	24-Dec-13	99.87	24288
B&D cordless drill, 1/2-in.	20-Jan-14	38.95	25595
Claw hammer	20-Jan-14	9.95	21225
Hicut chain saw, 16 in.	07-Feb-14	256.99	24288
PVC pipe, 3.5-in., 8-ft	20-Feb-14	5.87	
1.25-in. metal screw, 25	01-Mar-14	6.99	21225
2.5-in. wd. screw, 50	24-Feb-14	8.45	21231

Cengage Learning © 2015

Special Operators

BETWEEN

- Checks whether attribute value is within a range

IS NULL

- Checks whether attribute value is null

LIKE

- Checks whether attribute value matches given string pattern

IN

- Checks whether attribute value matches any value within a value list

EXISTS

- Checks if subquery returns any rows

Advanced Data Definition Commands

- **ALTER TABLE** command: To make changes in the table structure
- Keywords use with the command
  - ADD - Adds a column
  - MODIFY - Changes column characteristics
  - DROP - Deletes a column
- Used to:
  - Add table constraints
  - Remove table constraints

Changing Column's Data Type

- ALTER can be used to change data type
- Some RDBMSs do not permit changes to data types unless column is empty
- Syntax –
  - ALTER TABLE *tablename* MODIFY *(columnname(datatype))*;

Changing Column's Data Characteristics

- Use ALTER to change data characteristics
- Changes in column's characteristics are permitted if changes do not alter the existing data type
- Syntax
  - ALTER TABLE *tablename* MODIFY *(columnname(characterstic))*;

Adding Column, Dropping Column

- Adding a column
  - Use ALTER and ADD
  - Do not include the NOT NULL clause for new column
- Dropping a column
  - Use ALTER and DROP
  - Some RDBMSs impose restrictions on the deletion of an attribute

Advanced Data Updates

- UPDATE command updates only data in existing rows
- If a relationship is established between entries and existing columns, the relationship can assign values to appropriate slots
- Arithmetic operators are useful in data updates
- In Oracle, ROLLBACK command undoes changes made by last two UPDATE statements

Copying Parts of Tables

- SQL permits copying contents of selected table columns
  - Data need not be reentered manually into newly created table(s)
- Table structure is created
- Rows are added to new table using rows from another table

Adding Primary and Foreign Key Designations

- **ALTER TABLE** command
  - Followed by a keyword that produces the specific change one wants to make
  - Options include ADD, MODIFY, and DROP
- Syntax to add or modify columns
  - ALTER TABLE *tablename*
    - {ADD | MODIFY} (*columnname datatype* [{ADD | MODIFY} *columnname datatype*]) ;
  - ALTER TABLE *tablename*
    - ADD *constraint* [ADD *constraint*] ;

Deleting a Table from the Database

- **DROP TABLE**: Deletes table from database
  - Syntax - DROP TABLE tablename;
- Can drop a table only if it is not the one side of any relationship
  - RDBMS generates a foreign key integrity violation error message if the table is dropped

Additional SELECT Query Keywords

- Logical operators work well in the query environment
- SQL provides useful functions that:
  - Counts
  - Find minimum and maximum values
  - Calculate averages
- SQL allows user to limit queries to entries:
  - Having no duplicates
  - Whose duplicates may be grouped

Ordering a Listing

- **ORDER BY** clause is useful when listing order is important
- Syntax - SELECT *columnlist*

  FROM *tablelist*

  [WHERE *conditionlist*]

  [ORDER BY *columnlist* [ASC | DESC]];
- **Cascading order sequence**: Multilevel ordered sequence
  - Created by listing several attributes after the ORDER BY clause

Listing Unique Values

- **DISTINCT** clause: Produces list of values that are unique
- Syntax - SELECT DISTINCT *columnlist*

  FROM *tablelist*;
- Access places nulls at the top of the list
  - Oracle places it at the bottom
  - Placement of nulls does not affect list contents

Table 7.8 - Some Basic SQL Aggerate Functions

FUNCTION	OUTPUT
COUNT	The number of rows containing non-null values
MIN	The minimum attribute value encountered in a given column
MAX	The maximum attribute value encountered in a given column
SUM	The sum of all values for a given column
AVG	The arithmetic mean (average) for a specified column

Cengage Learning © 2015

Grouping Data

- Frequency distributions created by **GROUP BY** clause within SELECT statement
- Syntax - SELECT *columnlist*

  FROM *tablelist*

  [WHERE *conditionlist*]

  [GROUP BY *columnlist*]

  [HAVING *conditionlist*]

  [ORDER BY *columnlist* [ASC | DESC]];

HAVING Clause

- Extension of GROUP BY feature
- Applied to output of GROUP BY operation
- Used in conjunction with GROUP BY clause in second SQL command set
- Similar to WHERE clause in SELECT statement

Joining Database Tables

- Performed when data are retrieved from more than one table at a time
  - Equality comparison between foreign key and primary key of related tables
- Tables are joined by listing tables in FROM clause of SELECT statement
  - DBMS creates Cartesian product of every table in the FROM clause

Joining Tables With an Alias

- Alias identifies the source table from which data are taken
- Any legal table name can be used as alias
- Add alias after table name in FROM clause
  - FROM tablename alias

Recursive Joins

- **Recursive query**: Table is joined to itself using alias
- Use aliases to differentiate the table from itself