The National Higher School of Artificial Intelligence

DATABASES

Chapter 4 : Introduction to Structured Query Language (SQL)

Pr. Kamel BOUKHALFA

16/10/2022

Slides From the Textbook :
Carlos Coronel and Steven Morris, Database Systems: Design, Implementation, and Management
Tenth Edition

Objectives

In this chapter, students 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

Introduction to SQL

- SQL functions fit into two broad categories:
  - Data definition language
  - Data manipulation language
- Basic command set has vocabulary of fewer than 100 words
- American National Standards Institute (ANSI) prescribes a standard SQL
- Several SQL dialects exist

TABLE 7.1 SQL Data Definition Commands

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

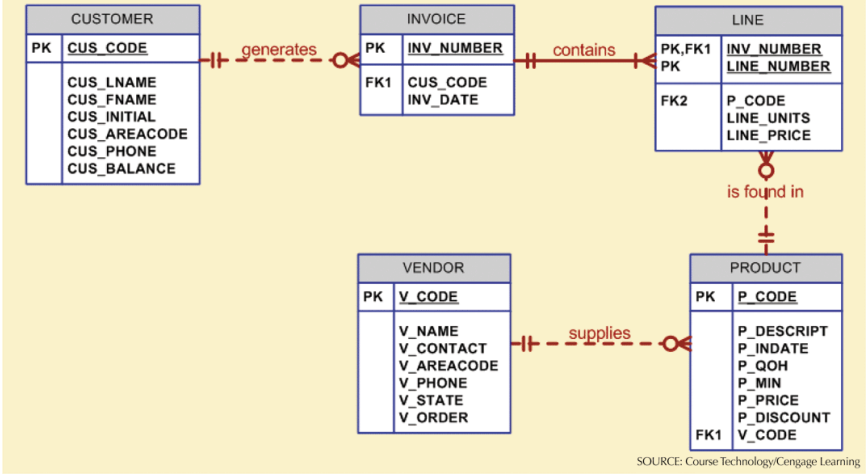
TABLE 7.2 SQL Data Manipulation Commands

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

Data Definition Commands

- The database model
  - In this chapter, a simple database with these tables is used to illustrate commands:
    - CUSTOMER
    - INVOICE
    - LINE
    - PRODUCT
    - VENDOR
  - Focus on PRODUCT and VENDOR tables

FIGURE 7.1 The database model



SOURCE: Course Technology/Cengage Learning

Creating the Database

- Two tasks must be completed:
  - Create database structure
  - Create tables that will hold end-user data
- First task:
  - RDBMS creates physical files that will hold database
  - Differs substantially from one RDBMS to another

Creating the Database (cont'd.)

- Authentication
  - DBMS verifies that only registered users are able to access database
  - Log on to RDBMS using user ID and password created by database administrator

The Database Schema

- Schema
  - Group of database objects that are related to each other
- CREATE SCHEMA AUTHORIZATION {creator};
  - Command is seldom used directly

Data Types

- Data type selection is usually dictated by nature of data and by intended use
- Supported data types:
  - Number(L,D), Integer, Smallint, Decimal(L,D)
  - Char(L), Varchar(L), Varchar2(L)
  - Date, Time, Timestamp
  - Real, Double, Float
  - Interval day to hour
  - Many other types

TABLE 7.3 Data Dictionary for the Ch07_SaleCo Database

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE*	REQUIRED	PK OR FK	FK REFERENCED TABLE
PRODUCT	P_CODE	Product code	VARCHAR(10)	XXXXXXXXXX	NA	Y	PK	
	P_DESCRIPT	Product description	VARCHAR(35)	Xxxxxxxxxxxx	NA	Y		
	P_INDATE	Stocking date	DATE	DD-MON-YYYY	NA	Y		
	P_QOH	Units available	SMALLINT	####	0–9999	Y		
	P_MIN	Minimum units	SMALLINT	####	0–9999	Y		
	P_PRICE	Product price	NUMBER(8,2)	####.##	0.00–9999.00	Y		
	P_DISCOUNT	Discount rate	NUMBER(5,2)	0.##	0.00–0.20	Y		
	V_CODE	Vendor code	INTEGER	###	100–999		FK	VENDOR
VENDOR	V_CODE	Vendor code	INTEGER	#####	1000–9999	Y	PK	
	V_NAME	Vendor name	VARCHAR(35)	Xxxxxxxxxxxxx	NA	Y		
	V_CONTACT	Contact person	VARCHAR(25)	Xxxxxxxxxxxxx	NA	Y		
	V_AREACODE	Area code	CHAR(3)	999	NA	Y		
	V_PHONE	Phone number	CHAR(8)	999–9999	NA	Y		
	V_STATE	State	CHAR(2)	XX	NA	Y		
	V_ORDER	Previous order	CHAR(1)	X	Y or N	Y		

TABLE 7.4 Some Common SQL Data Types

DATA TYPE	FORMAT	COMMENTS
Numeric	NUMBER(L,D)	The declaration NUMBER(7,2) indicates that numbers will be stored with two decimal places and may be up to seven digits long, including the sign and the decimal place (for example, 12.32 or -134.99).
	INTEGER	May be abbreviated as INT. Integers are (whole) counting numbers, so they cannot be used if you want to store numbers that require decimal places.
	SMALLINT	Like INTEGER but limited to integer values up to six digits. If your integer values are relatively small, use SMALLINT instead of INT.
	DECIMAL(L,D)	Like the NUMBER specification, but the storage length is a *minimum* specification. That is, greater lengths are acceptable, but smaller ones are not. DECIMAL(9,2), DECIMAL(9), and DECIMAL are all acceptable.
Character	CHAR(L)	Fixed-length character data for up to 255 characters. If you store strings that are not as long as the CHAR parameter value, the remaining spaces are left unused. Therefore, if you specify CHAR(25), strings such as *Smith* and *Katzenjammer* are each stored as 25 characters. However, a U.S. area code is always three digits long, so CHAR(3) would be appropriate if you wanted to store such codes.
	VARCHAR(L) or VARCHAR2(L)	Variable-length character data. The designation VARCHAR2(25) will let you store characters up to 25 characters long. However, VARCHAR will not leave unused spaces. Oracle automatically converts VARCHAR to VARCHAR2.
Date	DATE	Stores dates in the Julian date format.

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
- NOT NULL specification
- UNIQUE specification

Creating Table Structures (cont'd.)

- 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

SQL Constraints

- NOT NULL constraint
  - Ensures that column does not accept nulls
- UNIQUE constraint
  - Ensures that all values in column are unique
- DEFAULT constraint
  - Assigns value to attribute when a new row is added to table
- CHECK constraint
  - Validates data when attribute value is entered

SQL Indexes

- When primary key is declared, DBMS automatically creates unique index
- Often need additional indexes
- Using CREATE INDEX command, SQL indexes can be created on basis of any selected attribute
- Composite index
  - Index based on two or more attributes
  - Often used to prevent data duplication

Data Manipulation Commands

- INSERT
- SELECT
- COMMIT
- UPDATE
- ROLLBACK
- DELETE

Adding Table Rows

- INSERT
  - Used to enter data into table
  - Syntax:
    - INSERT INTO columnname
      VALUES (value1, value2, … , valueN);

Adding Table Rows (cont'd.)

- When entering values, notice that:
  - Row contents are entered between parentheses
  - Character and date values are entered between apostrophes
  - Numerical entries are not enclosed in apostrophes
  - Attribute entries are separated by commas
  - A value is required for each column
- Use NULL for unknown values

Saving Table Changes

- Changes made to table contents are not physically saved on disk until:
  - Database is closed
  - Program is closed
  - COMMIT command is used
- Syntax:
  - COMMIT [WORK];
- Will permanently save any changes made to any table in the database

Listing Table Rows

- SELECT
  - Used to list contents of table
  - Syntax:

    SELECT columnlist
    FROM tablename;
- Columnlist represents one or more attributes, separated by commas
- Asterisk can be used as wildcard character to list all attributes

Updating Table Rows

- UPDATE
  - Modify data in a table
  - Syntax:

    UPDATE tablename
    SET columnname = expression [, columnname = expression]
    [WHERE conditionlist];
- If more than one attribute is to be updated in row, separate corrections with commas

Restoring Table Contents

- ROLLBACK
  - Undoes changes since last COMMIT
  - Brings data back to prechange values
- Syntax:

  ROLLBACK;
- COMMIT and ROLLBACK only work with commands to add, modify, or delete table rows

Deleting Table Rows

- DELETE
  - Deletes a table row
  - Syntax:

    DELETE FROM tablename
    [WHERE conditionlist];
- WHERE condition is optional
- If WHERE condition is not specified, all rows from specified table will be deleted

Inserting Table Rows with a SELECT Subquery

- INSERT
  - Inserts multiple rows from another table (source)
  - Uses SELECT subquery
  - Subquery: query embedded (or nested or inner) inside another query
  - Subquery executed first
  - Syntax:

    INSERT INTO tablename SELECT columnlist
    FROM tablename;

SELECT Queries

- Fine-tune SELECT command by adding restrictions to search criteria using:
  - Conditional restrictions
  - Arithmetic operators
  - Logical operators
  - Special operators

Selecting Rows with Conditional Restrictions

- Select partial table contents by placing restrictions on rows to be included in output
  - Add conditional restrictions to SELECT statement, using WHERE clause
- Syntax:

```
SELECT columnlist
  FROM tablelist
  [ WHERE conditionlist ] ;
```

FIGURE 7.5 The Microsoft Access QBE and its SQL

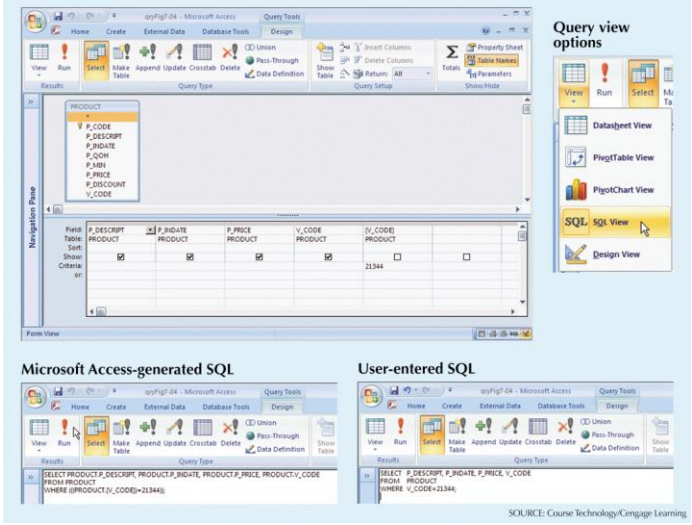


TABLE 7.6 Comparison Operators

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

Selecting Rows with Conditional Restrictions (cont'd.)

- Using comparison operators on dates
  - Date procedures are often more software-specific than other SQL procedures
- Using computed columns and column aliases
  - SQL accepts any valid expressions (or formulas) in the computed columns
  - Alias
    - Alternate name given to a column or table in any SQL statement

Arithmetic Operators: The Rule of Precedence

- Perform operations within parentheses
- Perform power operations
- Perform multiplications and divisions
- Perform 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 ^)

Logical Operators: AND, OR, and NOT

- Searching data involves multiple conditions
- Logical operators: AND, OR, and NOT
- Can be combined
  - Parentheses enforce precedence order
    - Conditions in parentheses are always executed first
- Boolean algebra: mathematical field dedicated to use of logical operators
- NOT negates result of conditional expression

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

- All changes in table structure are made by using ALTER command
- Three options:
  - ADD adds a column
  - MODIFY changes column characteristics
  - DROP deletes a column
- Can also be used to:
  - Add table constraints
  - Remove table constraints

Changing a 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

Changing a 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

Adding a Column Dropping a Column

- Use ALTER to add column
  - Do not include the NOT NULL clause for new column
- Use ALTER to drop column
  - Some RDBMSs impose restrictions on the deletion of an attribute

Advanced Data Updates

- UPDATE command updates only data in existing rows
- If relationship between entries and existing columns, can assign values to 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)
- First create the table structure
- Next add rows to new table using table rows from another table

Adding Primary and Foreign Key Designations

- When table is copied, integrity rules do not copy
  - Primary and foreign keys are manually defined on new table
- User ALTER TABLE command
  - Syntax:
    - ALTER TABLE tablename ADD PRIMARY KEY(fieldname);
  - For foreign key, use FOREIGN KEY in place of PRIMARY KEY

Deleting a Table from the Database

- DROP
  - Deletes table from database
  - Syntax:
    - DROP TABLE tablename;
- Can drop a table only if it is not the “one” side of any relationship
  - Otherwise, RDBMS generates an error message
  - Foreign key integrity violation

Additional SELECT Query Keywords

- Logical operators work well in the query environment
- SQL provides useful functions that:
  - Count
  - Find minimum and maximum values
  - Calculate averages, etc.
- SQL allows user to limit queries to:
  - Entries having no duplicates
  - Entries 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]];
```

- Ascending order by default

Listing Unique Values

- DISTINCT clause produces list of only values that are different from one another
- Example:

```
SELECT DISTINCT V_CODE
FROM  PRODUCT;
```

- Access places nulls at the top of the list
  - –Oracle places it at the bottom
  - –Placement of nulls does not affect list contents

Aggregate Functions

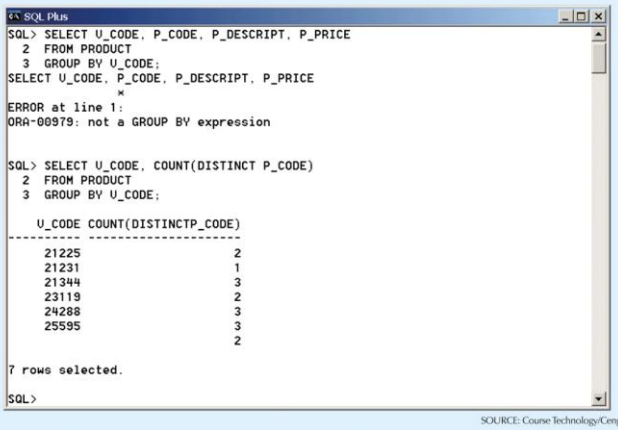
- COUNT function tallies number of non-null values of an attribute
  - – Takes one parameter: usually a column name
- MAX and MIN find highest (lowest) value in a table
  - – Compute MAX value in inner query
  - – Compare to each value returned by the query
- SUM computes total sum for any specified attribute
- AVG function format is similar to MIN and MAX

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] ] ;
```


FIGURE 7.26 Incorrect and correct use of the GROUP BY clause



```
SQL Plus
SQL> SELECT V_CODE, P_CODE, P_DESCRIPT, P_PRICE
  2  FROM PRODUCT
  3  GROUP BY V_CODE;
SELECT V_CODE, P_CODE, P_DESCRIPT, P_PRICE
               *
ERROR at line 1:
ORA-00979: not a GROUP BY expression


SQL> SELECT V_CODE, COUNT(DISTINCT P_CODE)
  2  FROM PRODUCT
  3  GROUP BY V_CODE;

    V_CODE COUNT(DISTINCTP_CODE)
---------- --------------------
     21225                    2
     21231                    1
     21344                    3
     23119                    2
     24288                    3
     25595                    3
                              2

7 rows selected.

SQL>
```

SOURCE: Course Technology/Cengage Learning

Joining Database Tables

- Joining tables is the most important distinction between relational database and other DBs
- Join is performed when data are retrieved from more than one table at a time
  - Equality comparison between foreign key and primary key of related tables
- Join tables by listing tables in FROM clause of SELECT statement
  - DBMS creates Cartesian product of every table

Joining Tables with an Alias

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

Recursive Joins

- Alias is especially useful when a table must be joined to itself
  - Recursive query
  - Use aliases to differentiate the table from itself

Summary

- SQL commands can be divided into two overall categories:
  - Data definition language commands
  - Data manipulation language commands
- The ANSI standard data types are supported by all RDBMS vendors in different ways
- Basic data definition commands allow you to create tables and indexes

Summary (cont'd.)

- DML commands allow you to add, modify, and delete rows from tables
- The basic DML commands:
  - SELECT, INSERT, UPDATE, DELETE, COMMIT, and ROLLBACK
- SELECT statement is main data retrieval command in SQL

Summary (cont'd.)

- WHERE clause can be used with SELECT, UPDATE, and DELETE statements
- Aggregate functions
  - Special functions that perform arithmetic computations over a set of rows
- ORDER BY clause
  - Used to sort output of SELECT statement
  - Can sort by one or more columns
  - Ascending or descending order

Summary (cont'd.)

- Join output of multiple tables with SELECT statement
  - Join performed every time you specify two or more tables in FROM clause
  - If no join condition is specified, DBMX performs Cartesian product
- Natural join uses join condition to match only rows with equal values in specified columns