ISTN212: Databases

Chapter 5: Beginning Structured Query Language

What is SQL ?

- It is a DDL – Data Definition Language
  - Allows for the creation of DB objects and defines access rights to those objects
- It is a DML – Data Manipulation Language
  - Allows for manipulation of data within DB
- Easy to learn with command set of < 100 words
- Several SQL dialects (Oracle, MS SQL server, IBM)

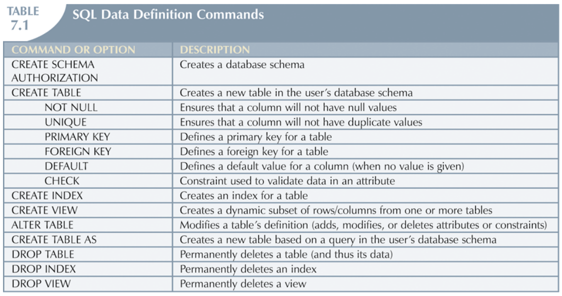


TABLE 7.1 SQL Data Definition Commands

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	Constraint used to validate data in an attribute
CREATE INDEX	Creates an index for a table
CREATE VIEW	Creates a dynamic subset of rows/columns from one or more tables
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 thus its data)
DROP INDEX	Permanently deletes an index
DROP VIEW	Permanently deletes a view

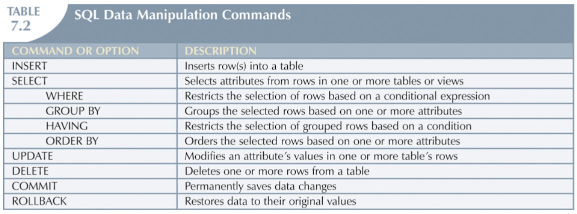


TABLE 7.2 SQL Data Manipulation Commands

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

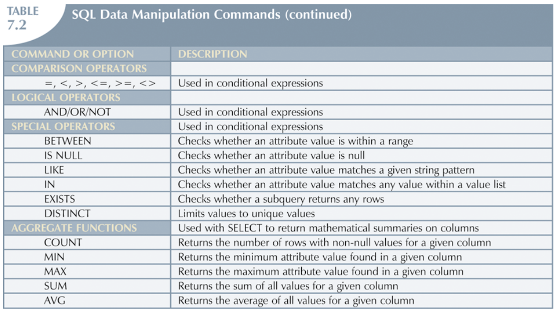
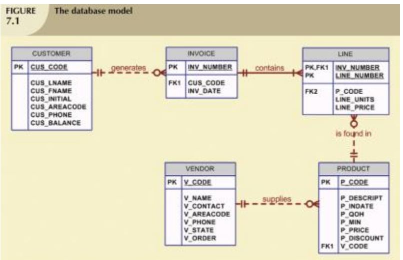


TABLE 7.2 SQL Data Manipulation Commands (continued)

COMMAND OR OPTION	DESCRIPTION
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

Example: Database Model

FIGURE 7.1 The database model



Example continued

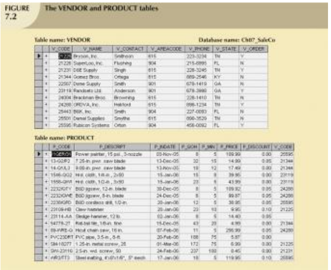


FIGURE 7.2 The VENDOR and PRODUCT tables

Table name: VENDOR

Database name: Ch07_SaleCo

V_CODE	V_NAME	V_CONTACT	V_AREACODE	V_PHONE	V_STATE	V_ORDER
21225	Bryson, Inc.	Smithson	615	223-3234	TN	Y
21226	SuperLoo, Inc.	Flushing	904	215-8995	FL	N
21231	D&E Supply	Singh	615	228-3245	TN	Y
21344	Gomez Bros.	Ortega	615	889-2546	KY	N
22567	Dome Supply	Smith	901	678-1419	GA	N
23119	Randsets Ltd.	Anderson	901	678-3998	GA	Y
24004	Brackman Bros.	Browning	615	228-1410	TN	N
24288	ORDVA, Inc.	Hakford	615	898-1234	TN	Y
25443	B&K, Inc.	Smith	904	227-0093	FL	N
25501	Damal Supplies	Smythe	615	890-3529	TN	N
25595	Rubicon Systems	Orton	904	456-0092	FL	Y

Table name: PRODUCT

P_CODE	P_DESCRIPT	P_INDATE	P_QOH	P_MIN	P_PRICE	P_DISCOUNT	V_CODE
11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-05	8	5	109.99	0.00	25595
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-05	32	15	14.99	0.05	21344
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-05	18	12	17.49	0.00	21344
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-06	15	8	39.95	0.00	23119
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-06	23	5	43.99	0.00	23119
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-05	8	5	109.92	0.05	24288
2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-05	6	5	99.87	0.05	24288
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-06	12	5	38.95	0.05	25595
23109-HB	Claw hammer	20-Jan-06	23	10	9.95	0.10	21225
23114-AA	Sledge hammer, 12 lb.	02-Jan-06	8	5	14.40	0.05	
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-05	43	20	4.99	0.00	21344
89-WRE-Q	Hicut chain saw, 16 in.	07-Feb-06	11	5	256.99	0.05	24288
PVC23DRT	PVC pipe, 3.5-in., 8-ft.	20-Feb-06	188	75	5.87	0.00	
SM-18277	1.25-in. metal screw, 25	01-Mar-06	172	75	6.99	0.00	21225
SW-23116	2.5-in. wd. screw, 50	24-Feb-06	237	100	8.45	0.00	21231
WR3/TT3	Steel matting, 4'x8'x1/6", .5" mesh	17-Jan-06	18	5	119.95	0.10	25595

Creating the DB

- Two tasks must be completed

1. Create the DB structure
2. Create the tables that will hold the data

- RDBMS creates physical files that will hold the DB
- Data dictionary is automatically created to hold metadata
- Setup is often different between RDBMS's

Data types

- Determined by the nature of the data and intended use of data
- Must pay close attention to expected use of attributes for sorting and data retrieval purposes

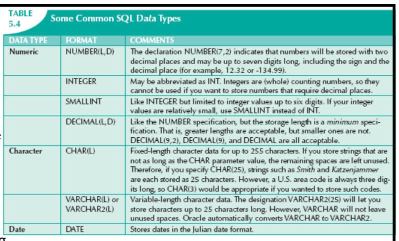


TABLE 5.4 Some Common SQL Data Types

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 a Table Structure using SQL

```
CREATE TABLE tablename (
  column1  data type [constraint] [,
  column2 data type [constraint] [,
  PRIMARY KEY (column1 [,column2]) ] [,
  FOREIGN KEY (column1 [, column2]) REFERENCES tablename] [,
  CONTRAINT constraint ]);
```

Example: Creating a table of Products

```
CREATE TABLE PRODUCT (
P_CODE         VARCHAR(10)     NOT NULL   UNIQUE,
P_DESCRIPT   VARCHAR(35)    NOT NULL,
P_INDATE        DATE                NOT NULL,
P_QOH            SMALLINT         NOT NULL,
P_MIN              SMALLINT         NOT NULL,
P_PRICE          NUMBER(8,2)    NOT NULL,
P_DISCOUNT   NUMBER(4,2)    NOT NULL,
V_CODE            INTEGER,
PRIMARY KEY (P_CODE),
FOREIGN KEY (V_CODE) REFERENCES VENDOR ON UPDATE CASCADE);
```

May not be necessary depending on which RDBMS being used

Example: Creating a table of customer

```
CREATE TABLE CUSTOMER(
CUS_CODE  NUMBER PRIMARY KEY,
CUS_LNAME VARCHAR(15) NOT NULL,
CUS_FNAME VARCHAR(15) NOT NULL,
CUS_INITIAL CHAR(1),
CUS_AREACODE CHAR(3)  DEFAULT '615' NOT NULL
                        CHECK (CUS_AREACODE IN
                          ('615', '713', '931')),
CUS_PHONE       CHAR(12)  NOT NULL,
CUS_BALANCE    NUMBER(9,2)    DEFAULT 0.00
CONSTRAINT CUS_UI1 UNIQUE (CUS_LNAME, CUS_FNAME));
```

Some suggestions and things to note

- 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 – Blanks not allowed
- UNIQUE specification – No duplicates
- DEFAULT – Can provide a default value is user does not enter any value
- CHECK – Validates data and ensures that only certain values are accepted

Some suggestions and things to note

- CONSTRAINT CUS_UI1 UNIQUE (CUS_LNAME, CUS_FNAME));
  - Unique index created called CUS_UI1
  - Prevents two customers with same first name and last name
    - NOT RECOMMENDED as there can be more than one John Smith
- Primary key attributes contain both a NOT NULL and a UNIQUE specification
- RDBMS will automatically enforce referential integrity for foreign keys – a condition by which a dependent table's foreign key must have either a null entry or a matching entry in the related table
- Command sequence ends with semicolon

Indexes

- Covered in Chapter 4, it is an orderly arrangement used to logically access rows in a table
- 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

Indexes

CREATE [UNIQUE] INDEX indexname ON tablename(column1 [,column2])

- P_INDATE stored in the PRODUCT table
- To create an index:

CREATE INDEX P_INDATEX ON PRODUCT(P_INDATE);

- to delete an index, use the drop index command:

DROP INDEX indexname

Indexes

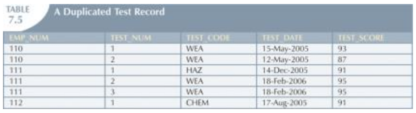


TABLE 7.5 A Duplicated Test Record

EMP_NUM	TEST_NUM	TEST_CODE	TEST_DATE	TEST_SCORE
110	1	WEA	15-May-2005	93
110	2	WEA	12-May-2005	87
111	1	HAZ	14-Dec-2005	91
111	2	WEA	18-Feb-2006	95
111	3	WEA	18-Feb-2006	95
112	1	CHEM	17-Aug-2005	91

- An employee can take a test ONLY once on a given date
- PK is combination of EMP_NUM and TEST_NUM
- Problem since WEA test taken twice by EMP 111 on 18 Feb 2006
- By creating a unique index – combo of EMP_NUM, TEST_CODE and TEST_DATE, problem can be avoided

CREATE UNIQUE INDEX EMP_TESTDEX ON TEST(EMP_NUM, TEST_CODE, TEST_DATE);

Data Manipulation Commands

- Adding table rows
- Saving table changes
- Listing table rows
- Updating table rows
- Restoring table contents
- Deleting table rows
- Inserting table rows with a select subquery

Adding Table Rows (Adding data to tables)

- INSERT - Used to enter data into table
- Syntax:

```
INSERT INTO tablename
VALUES (value1, value2, … , valuen);
```

- Example:

```
INSERT INTO PRODUCT
VALUES ('11QER/31', 'blade 3-nozzle', ……);
```

Adding Table Rows (Adding data to tables)

- When entering values:
  - 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 using COMMIT

- Changes made to table contents are not physically saved on disk until, one of the following occurs:
  - 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
- MS Access does not support the COMMIT command

Listing table rows

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

```
SELECT column1, column2 etc.
FROM tablename;
```

- One or more attributes, separated by commas
- Example

```
SELECT P_CODE, P_QOH FROM PRODUCT
```

- Asterisk can be used as wildcard character to list all attributes
  - SELECT * FROM *tablename;*

Updating the table data

- UPDATE - Modify data in a table
- Syntax:

```
UPDATE tablename
SET columnname = expression [, columname = expression]
[WHERE conditionlist];
```

- If more than one attribute is to be updated in row, separate corrections with commas

```
UPDATE PRODUCT
SET P_INDATE = '18-JAN-2006',
        P_PRICE = 17.99, P_MIN = 10
WHERE P_CODE = '13-Q2/P2';
```

Restoring Table Contents

- ROLLBACK
  - Used to restore database to its previous condition
  - Only applicable if COMMIT command has not been used to permanently store changes in database
- Syntax:
  - ROLLBACK;
- COMMIT and ROLLBACK only work with data manipulation commands that are used to add, modify, or delete table rows

Deleting rows from a table (Deleting data)

- 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

```
DELETE FROM PRODUCT
WHERE P_MIN = 5;
```

Inserting Table Rows with a Select Statement

- To Insert multiple rows from another table (source)
- Use INSERT with a SELECT subquery
  - Query that is embedded (or nested) inside another query is executed first
- Syntax:

INSERT INTO *tablename* SELECT *columnlist* FROM *tablename*;

```
INSERT INTO PRODUCT SELECT * FROM P;
```

- Takes all data from all attributes from table P and inserts them into table PRODUCT

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
  - Can use comparison operators
- Syntax:

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

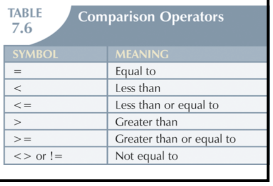


TABLE 7.6 Comparison Operators

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 (continued)

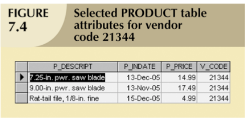


FIGURE 7.4 Selected PRODUCT table attributes for vendor code 21344

	P_DESCRIPT	P_INDATE	P_PRICE	V_CODE
▶	7.25-in. pwr. saw blade	13-Dec-05	14.99	21344
	9.00-in. pwr. saw blade	13-Nov-05	17.49	21344
	Rat-tail file, 1/8-in. fine	15-Dec-05	4.99	21344

```
SELECT P_DESCRIPT, P_INDATE, P_PRICE, V_CODE
FROM PRODUCT
WHERE V_CODE = 21344;
```

Database Principles: Fund. of Design, Impl., & Management,
10th Edition, Coronel, Morris & Rob

Selecting Rows with Conditional Restrictions (continued)

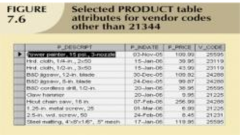


FIGURE 7.6 Selected PRODUCT table attributes for vendor codes other than 21344

	P_DESCRIPT	P_INDATE	P_PRICE	V_CODE
▶	Power painter, 15 psi., 3-nozzle	03-Nov-05	109.99	25595
	Hrd. cloth, 1/4-in., 2x50	15-Jan-06	39.95	23119
	Hrd. cloth, 1/2-in., 3x50	15-Jan-06	43.99	23119
	B&D jigsaw, 12-in. blade	30-Dec-05	109.92	24288
	B&D jigsaw, 8-in. blade	24-Dec-05	99.87	24288
	B&D cordless drill, 1/2-in.	20-Jan-06	38.95	25595
	Claw hammer	20-Jan-06	9.95	21225
	Hicut chain saw, 16 in.	07-Feb-06	256.99	24288
	1.25-in. metal screw, 25	01-Mar-06	6.99	21225
	2.5-in. wd. screw, 50	24-Feb-06	8.45	21231
	Steel matting, 4'x8'x1/6", .5" mesh	17-Jan-06	119.95	25595

```
SELECT P_DESCRIPT, P_INDATE, P_PRICE,
V_CODE
FROM PRODUCT
WHERE V_CODE <> 21344;
```

Database Principles: Fund. of Design, Impl., & Management,
10th Edition, Coronel, Morris & Rob

Selecting Rows with Conditional Restrictions (continued)

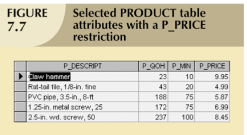


FIGURE 7.7 Selected PRODUCT table attributes with a P_PRICE restriction

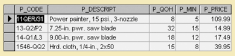
	P_DESCRIPT	P_QOH	P_MIN	P_PRICE
▶	Claw hammer	23	10	9.95
	Rat-tail file, 1/8-in. fine	43	20	4.99
	PVC pipe, 3.5-in., 8-ft	188	75	5.87
	1.25-in. metal screw, 25	172	75	6.99
	2.5-in. wd. screw, 50	237	100	8.45

```
SELECT P_DESCRIPT, P_QOH, P_MIN, P_PRICE
FROM PRODUCT
WHERE P_PRICE <=10;
```

Database Principles: Fund. of Design, Impl., & Management,
10th Edition, Coronel, Morris & Rob

Selecting Rows with Conditional Restrictions (continued)

FIGURE 7.8 Selected PRODUCT table attributes: the ASCII code effect



P_CODE	P_DESCRIPT	P_QOH	P_MIN	P_PRICE
11QER/31	Power painter, 15 psi., 3-nozzle	8	5	109.99
13-Q2/P2	7.25-in. pwr. saw blade	32	15	14.99
14-Q1/L3	9.00-in. pwr. saw blade	18	12	17.49
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15	8	39.95

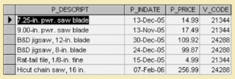
SELECT P_CODE, P_DESCRIPT, P_QOH, P_MIN, P_PRICE

FROM PRODUCT

WHERE P_CODE < ‘1558-QW1’;

Logical Operators: AND, OR, and NOT

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-05	14.99	21344
9.00-in. pwr. saw blade	13-Nov-05	17.49	21344
B&D jigsaw, 12-in. blade	30-Dec-05	109.92	24288
B&D jigsaw, 8-in. blade	24-Dec-05	99.87	24288
Rat-tail file, 1/8-in. fine	15-Dec-05	4.99	21344
Hicut chain saw, 16 in.	07-Feb-06	256.99	24288

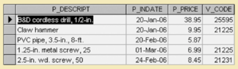
SELECT P_DESCRIPT, P_INDATE, P_PRICE, V_CODE

FROM PRODUCT

WHERE V_CODE = 21344 OR V_CODE = 24288;

Logical Operators: AND, OR, and NOT (continued)

FIGURE 7.13 Selected PRODUCT table attributes: the logical AND

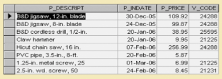


P_DESCRIPT	P_INDATE	P_PRICE	V_CODE
B&D cordless drill, 1/2-in.	20-Jan-06	38.95	25595
Claw hammer	20-Jan-06	9.95	21225
PVC pipe, 3.5-in., 8-ft.	20-Feb-06	5.87	
1.25-in. metal screw, 25	01-Mar-06	6.99	21225
2.5-in. wd. screw, 50	24-Feb-06	8.45	21231

SELECT P_DESCRIPT, P_INDATE, P_PRICE, V_CODE
FROM PRODUCT
WHERE P_PRICE < 50 AND P_INDATE > ’15-Jan-2006’;

Logical Operators: AND, OR, and NOT (continued)

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-05	109.92	24288
B&D jigsaw, 8-in. blade	24-Dec-05	99.87	24288
B&D cordless drill, 1/2-in.	20-Jan-06	38.95	25595
Claw hammer	20-Jan-06	9.95	21225
Hicut chain saw, 16 in.	07-Feb-06	256.99	24288
PVC pipe, 3.5-in., 8-ft.	20-Feb-06	5.87	
1.25-in. metal screw, 25	01-Mar-06	6.99	21225
2.5-in. wd. screw, 50	24-Feb-06	8.45	21231

SELECT P_DESCRIPT, P_INDATE, P_PRICE, V_CODE
FROM PRODUCT
WHERE (P_PRICE < 50 AND P_INDATE > ’15-Jan-2006’)
OR V_CODE = 24288;

Special operators

- BETWEEN - Used to check whether attribute value is within a range

SELECT * FROM PRODUCT

WHERE P_PRICE BETWEEN 50.00 AND 100.00

- IS NULL - Used to check whether attribute value is null

SELECT P_CODE, P_DESCRIPT, V_CODE

FROM PRODUCT

WHERE V_CODE IS NULL;

Special Operators

- LIKE - Used to check whether attribute value matches given string pattern

SELECT V_NAME, V_CONTACT
FROM VENDOR
WHERE V_CONTACT LIKE ‘Smith%’

- Wildcard string search based on the string ‘_m%’ can yield the string Am, Amber, Crumpets….
  - In MS Access
    - * matches with zero or more characters
    - ? matches with just one character
  - In other SQLs,
    - % matches with zero or more characters
    - _ (underscore) matches with just one character

Special Operators

- IN - Used to check whether attribute value matches any value within a value list

```
SELECT * FROM PRODUCT
WHERE V_CODE IN (21344, 24288)
```

- EXISTS
  - Used to check if subquery returns any rows
  - discussed in next chapter

Additional Data Definition Commands

- All changes in table structure are made by using ALTER command
  - Followed by keyword that produces specific change
  - Following three options are available:
    - ADD
    - MODIFY
    - DROP
  - Can be used to change data type or add columns etc.

```
ALTER TABLE PRODUCT              ALTER TABLE PRODUCT
MODIFY (V_CODE CHAR(5));         ADD (P_SALECODE CHAR(1));
```

Advanced Data Updates

```
UPDATE PRODUCT
SET P_QOH = P_QOH + 20
WHERE P_CODE = ‘2232/QWE’;

UPDATE PRODUCT
SET P_PRICE = P_PRICE * 1.10
WHERE P_PRICE  < 50.00;
```

Copying data from one table to another

- SQL permits copying contents of selected table columns so that the data need not be reentered manually into newly created table(s)
- First create the PART table structure

```
CREATE TABLE PART(
PART_CODE          CHAR(8)
PART_DESCRIPT      CHAR(35),
PART_PRICE         DECIMAL(8,2),
V_CODE             INTEGER,
PRIMARY KEY (PART_CODE));
```

Copying data from one table to another

- Next add rows to new PART table using PRODUCT table rows

```
INSERT INTO PART (PART_CODE, PART_DESCRIPT, PART_PRICE, V_CODE)
SELECT P_CODE, P_DESCRIPT, P_PRICE, V_CODE FROM PRODUCT;
```

- Can also rapidly create new table based on selected columns and rows of an existing table
- **In oracle (new table copies the attribute name, data characteristics and rows of original table**

```
    CREATE TABLE PART AS
    SELECT P_CODE AS PART_CODE, P_DESCRIPT AS
    PART_DESCRIPT, P_PRICE AS PART_PRICE, V_CODE
    FROM PRODUCT;
```

Adding Primary and Foreign Key Designations

- When table is copied, integrity rules do not copy, so primary and foreign keys need to be 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*;

Advanced Select Queries

- SQL provides useful functions that can:
  - Count
  - Find minimum and maximum values
  - Calculate averages
  - Distinct

Ordering a listing

- The order by clause is especially useful when listing order in important
- The syntax is:

SELECT columnlist

FROM tablelist

WHERE conditionlist]

ORDER BY columnlist [ASC | DESC)];

- Although you have an option of declaring the order type – ascending or descending – the default value is ascending

Ordering a Listing

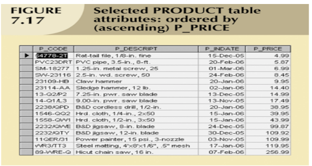


FIGURE 7.17 Selected PRODUCT table attributes: ordered by (ascending) P_PRICE

P_CODE	P_DESCRIPT	P_INDATE	P_PRICE
SM-18277	Rat-tail file, 1/8-in. fine	15-Dec-05	4.99
PVC23DRT	PVC pipe, 3.5-in., 8-ft.	20-Feb-06	5.87
SM-18277	1.25-in. metal screw, 25	01-Mar-06	6.99
SW-23116	2.5-in. wd. screw, 50	24-Feb-06	8.45
23109-HB	Claw hammer	20-Jan-06	9.95
23114-AA	Sledge hammer, 12 lb.	02-Jan-06	14.40
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-05	14.99
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-05	17.49
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-06	38.95
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-06	39.95
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-06	43.99
2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-05	99.87
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-05	109.92
11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-05	109.99
WR3/TT3	Steel matting, 4'x8'x1/6", .5" mesh	17-Jan-06	119.95
89-WRE-Q	Hicut chain saw, 16 in.	07-Feb-06	256.99

```
SELECT P_CODE, P_DESCRIPT, P_INDATE,
P_PRICE
FROM PRODUCT
ORDER BY P_PRICE;
```

Ordering a Listing (continued)

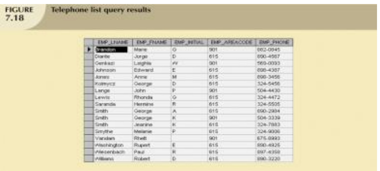


FIGURE 7.18 Telephone list query results

EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_AREACODE	EMP_PHONE
Brandon	Marie	G	901	882-0845
Diante	Jorge	D	615	890-4567
Genkazi	Leighla	W	901	569-0093
Johnson	Edward	E	615	898-4387
Jones	Anne	M	615	898-3456
Kolmycz	George	D	615	324-5456
Lange	John	P	901	504-4430
Lewis	Rhonda	G	615	324-4472
Saranda	Hermine	R	615	324-5505
Smith	George	A	615	890-2984
Smith	George	K	901	504-3339
Smith	Jeanine	K	615	324-7883
Smythe	Melanie	P	615	324-9006
Vandam	Rhett		901	675-8993
Washington	Rupert	E	615	890-4925
Wiesenbach	Paul	R	615	897-4358
Williams	Robert	D	615	890-3220

```
SELECT EMP_LNAME, EMP_FNAME, EMP_INITIAL,
EMP_AREACODE, EMP_PHONE
FROM EMPLOYEE
ORDER BY EMP_LNAME, EMP_FNAME, EMP_INITIAL;
```

Ordering a Listing (continued)

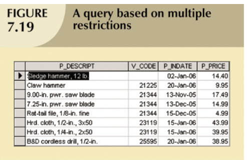
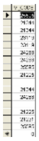


FIGURE 7.19 A query based on multiple restrictions

P_DESCRIPT	V_CODE	P_INDATE	P_PRICE
Sledge hammer, 12 lb.		02-Jan-06	14.40
Claw hammer	21225	20-Jan-06	9.95
9.00-in. pwr. saw blade	21344	13-Nov-05	17.49
7.25-in. pwr. saw blade	21344	13-Dec-05	14.99
Rat-tail file, 1/8-in. fine	21344	15-Dec-05	4.99
Hrd. cloth, 1/2-in., 3x50	23119	15-Jan-06	43.99
Hrd. cloth, 1/4-in., 2x50	23119	15-Jan-06	39.95
B&D cordless drill, 1/2-in.	25595	20-Jan-06	38.95

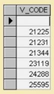
```
SELECT P_DESCRIPT, V_CODE, P_INDATE, P_PRICE
FROM PRODUCT
WHERE P_INDATE < ’21-Jan-2006’ AND P_PRICE <=50.00
ORDER BY V_CODE, P_PRICE DESC;
```

Listing Unique Values



	V_CODE
▶	[illegible]
	21344
	21344
	23119
	23119
	24288
	24288
	25595
	21225
	21344
	24288
	21225
	21231
	25595
*	0

FIGURE 7.20 A listing of distinct (different) V_CODE values in the PRODUCT table



	V_CODE
▶	
	21225
	21231
	21344
	23119
	24288
	25595

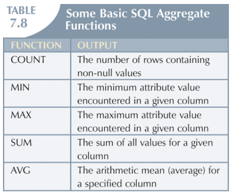
```
SELECT V_CODE
FROM PRODUCT;
```

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

Database Principles: Fund. of Design, Impl., & Management, 10th Edition, Coronel, Morris & Rob

Aggregate Functions

TABLE 7.8 Some Basic SQL Aggregate 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

Database Principles: Fund. of Design, Impl., & Management, 10th Edition, Coronel, Morris & Rob

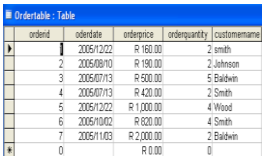
Count

- Function used to tally the number of non-null values of an attribute
- E.g. need to know how many vendors in the product tables
- Function uses one parameter within brackets
  - COUNT(V_CODE)
- Parameter may also be an expression
  - COUNT(DISTINCT P_CODE), COUNT(P_PRICE + 10)
- COUNT(*) returns the total number of rows including rows that contain nulls

SELECT COUNT(*)

FROM (SELECT DISTINCT V_CODE FROM PRODUCT WHERE V_CODE IS NOT NULL);

- Table used for count example

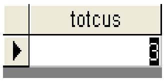


Ordertable : Table

	orderid	odendate	orderprice	orderquantity	customername
▶	1	2005/12/22	R 160.00	2	smith
	2	2005/08/10	R 190.00	2	Johnson
	3	2005/07/13	R 500.00	5	Baldwin
	4	2005/07/13	R 420.00	2	Smith
	5	2005/12/22	R 1,000.00	4	Wood
	6	2005/10/02	R 820.00	4	Smith
	7	2005/11/03	R 2,000.00	2	Baldwin
*	0		R 0.00	0	

Database Principles: Fund. of Design, Impl., & Management, 10th Edition, Coronel, Morris & Rob

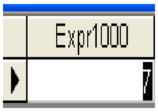
```
SELECT count(customername) AS totcus
FROM ordertable
WHERE customername='SMITH';
```



	totcus
▶	3

Database Principles: Fund. of Design, Impl., & Management, 10th Edition, Coronel, Morris & Rob

```
SELECT count(*)
FROM ordertable;
```

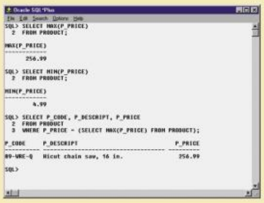


	Expr1000
▶	7

Database Principles: Fund. of Design, Impl., & Management, 10th Edition, Coronel, Morris & Rob

Aggregate Functions (continued)

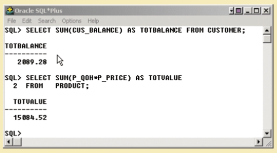
FIGURE 7.22 MAX and MIN function output examples



```
SQL> SELECT MAX(P_PRICE)
  2  FROM PRODUCT;

MAX(P_PRICE)
------------
      256.99

SQL> SELECT MIN(P_PRICE)
  2  FROM PRODUCT;

MIN(P_PRICE)
------------
        4.99

SQL> SELECT P_CODE, P_DESCRIPT, P_PRICE
  2  FROM PRODUCT
  3  WHERE P_PRICE = (SELECT MAX(P_PRICE) FROM PRODUCT);

P_CODE     P_DESCRIPT                          P_PRICE
---------- ----------------------------------- ---------
89-WRE-Q   Hicut chain saw, 16 in.                256.99

SQL>
```

Aggregate Functions (continued)

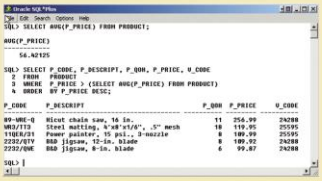
FIGURE 7.23 The total value of all items in the PRODUCT table



```
SQL> SELECT SUM(CUS_BALANCE) AS TOTBALANCE FROM CUSTOMER;

TOTBALANCE
----------
   2089.28

SQL> SELECT SUM(P_QOH*P_PRICE) AS TOTVALUE
  2  FROM   PRODUCT;

  TOTVALUE
----------
  15084.52

SQL>
```

Aggregate Functions (continued)

FIGURE 7.24 AVG function output examples



```
SQL> SELECT AVG(P_PRICE) FROM PRODUCT;

AVG(P_PRICE)
------------
    56.42125

SQL> SELECT P_CODE, P_DESCRIPT, P_QOH, P_PRICE, V_CODE
  2  FROM   PRODUCT
  3  WHERE  P_PRICE > (SELECT AVG(P_PRICE) FROM PRODUCT)
  4  ORDER  BY P_PRICE DESC;

P_CODE     P_DESCRIPT                              P_QOH   P_PRICE     V_CODE
---------- ----------------------------------- ---------- -------- ----------
89-WRE-Q   Hicut chain saw, 16 in.                     11   256.99      24288
WR3/TT3    Steel matting, 4'x8'x1/6", .5" mesh         18   119.95      25595
11QER/31   Power painter, 15 psi., 3-nozzle             8   109.99      25595
2232/QTY   B&D jigsaw, 12-in. blade                     8   109.92      24288
2232/QWE   B&D jigsaw, 8-in. blade                      6    99.87      24288

SQL> |
```

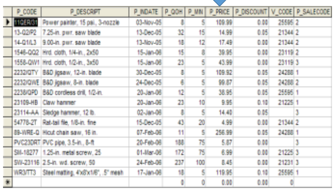
Grouping Data

- Forms groups of rows with same value of some columns
- The syntax is:

SELECT columnlist
FROM tablelist
WHERE conditionlist
GROUP BY columnlist
HAVING conditionlist
ORDER BY columnlist [ASC|DESC];

- GROUP BY is valid only in conjunction with one of the SQL aggregate functions, such as COUNT, MIN, MAX, AVG, SUM

Grouping Data



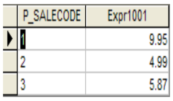
	P_CODE	P_DESCRIPT	P_INDATE	P_QOH	P_MIN	P_PRICE	P_DISCOUNT	V_CODE	P_SALECODE
▶	11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-05	8	5	109.99	0.00	25595	2
	13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-05	32	15	14.99	0.05	21344	2
	14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-05	18	12	17.49	0.00	21344	2
	1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-06	15	8	39.95	0.00	23119	2
	1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-06	23	5	43.99	0.00	23119	3
	2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-05	8	5	109.92	0.05	24288	1
	2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-05	6	5	99.87	0.05	24288	2
	2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-06	12	5	38.95	0.05	25595	1
	23109-HB	Claw hammer	20-Jan-06	23	10	9.95	0.10	21225	1
	23114-AA	Sledge hammer, 12 lb.	02-Jan-06	8	5	14.40	0.05		3
	54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-05	43	20	4.99	0.00	21344	2
	89-WRE-Q	Hicut chain saw, 16 in.	07-Feb-06	11	5	256.99	0.05	24288	1
	PVC23DRT	PVC pipe, 3.5-in., 8-ft	20-Feb-06	188	75	5.87	0.00		3
	SM-18277	1.25-in. metal screw, 25	01-Mar-06	172	75	6.99	0.00	21225	3
	SW-23116	2.5-in. wd. screw, 50	24-Feb-06	237	100	8.45	0.00	21231	3
	WR3/TT3	Steel matting, 4'x8'x1/6", .5" mesh	17-Jan-06	18	5	119.95	0.10	25595	1
*				0	0	0.00	0.00	0	

SELECT P_SALECODE, MIN(P_PRICE)
FROM PRODUCT_3
GROUP BY P_SALECODE;

We want to find the minimum/lowest price
for each salecode in Table PRODUCT_3

SELECT P_SALECODE, MIN(P_PRICE)
FROM PRODUCT_3
GROUP BY P_SALECODE;

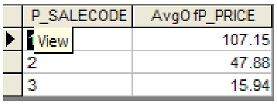
Only 3 values in column for P_SALECODE



	P_SALECODE	Expr1001
▶	1	9.95
	2	4.99
	3	5.87

SELECT P_SALECODE, Avg(P_PRICE) AS AvgOfP_PRICE
FROM PRODUCT_3
GROUP BY P_SALECODE;

AS…
What are you going to name the column with the Average results

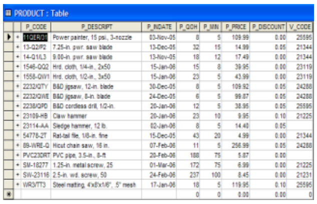


P_SALECODE	AvgOfP_PRICE
View	107.15
2	47.88
3	15.94

The GROUP BY feature's HAVING Clause

- HAVING operates like the WHERE clause in the SELECT statement
- However, WHERE applies to columns and expressions of individual rows, while HAVING is applied to the output of a GROUP BY operation

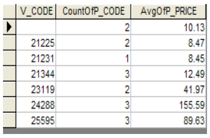
SELECT V_CODE, Count(P_CODE) AS CountOfP_CODE, Avg(P_PRICE) AS AvgOfP_PRICE
FROM PRODUCT_3
GROUP BY V_CODE;



PRODUCT : Table

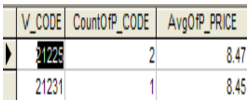
P_CODE	P_DESCRIPT	P_INDATE	P_QOH	P_MIN	P_PRICE	P_DISCOUNT	V_CODE
11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-05	8	5	109.99	0.00	25595
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-05	32	15	14.99	0.05	21344
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-05	18	12	17.49	0.00	21344
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-06	15	8	39.95	0.00	23119
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-06	23	5	43.99	0.00	23119
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-05	8	5	109.92	0.05	24288
2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-05	6	5	99.87	0.05	24288
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-06	12	5	38.95	0.05	25595
23109-HB	Claw hammer	20-Jan-06	23	10	9.95	0.10	21225
23114-AA	Sledge hammer, 12 lb.	02-Jan-06	8	5	14.40	0.05	
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-05	43	20	4.99	0.00	21344
89-WRE-Q	Hicut chain saw, 16 in.	07-Feb-06	11	5	256.99	0.05	24288
PVC23DRT	PVC pipe, 3.5-in., 8-ft	20-Feb-06	188	75	5.87	0.00	
SM-18277	1.25-in. metal screw, 25	01-Mar-06	172	75	6.99	0.00	21225
SW-23116	2.5-in. wd. screw, 50	24-Feb-06	237	100	8.45	0.00	21231
WR3/TT3	Steel matting, 4'x8'x1/6", .5" mesh	17-Jan-06	18	5	119.95	0.10	25595
			0	0	0.00	0.00	0

SELECT V_CODE, Count(P_CODE) AS CountOfP_CODE, Avg(P_PRICE) AS AvgOfP_PRICE
FROM PRODUCT_3
GROUP BY V_CODE;



V_CODE	CountOfP_CODE	AvgOfP_PRICE
	2	10.13
21225	2	8.47
21231	1	8.45
21344	3	12.49
23119	2	41.97
24288	3	155.59
25595	3	89.63

SELECT V_CODE, Count(P_CODE) AS CountOfP_CODE, Avg(P_PRICE) AS AvgOfP_PRICE
FROM PRODUCT_3
GROUP BY V_CODE
HAVING Avg(P_PRICE)<10;



V_CODE	CountOfP_CODE	AvgOfP_PRICE
21225	2	8.47
21231	1	8.45

JOINing DB Tables

- Chapter 4 – JOIN relational set operator
- To perform JOIN, simply list tables in the FROM clause

SELECT P_DESCRIPT, P_PRICE, V_NAME, V_CONTACT, V_AREACODE
FROM PRODUCT, VENDOR
WHERE PRODUCT.V_CODE = VENDOR.V_CODE

- Use full stop to reference attributes from two different tables

Grouping Data (continued)

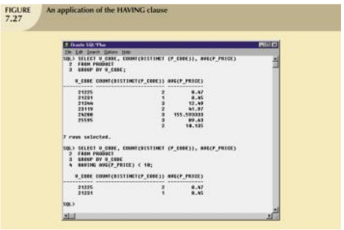


FIGURE 7.27 An application of the HAVING clause

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

   V_CODE COUNT(DISTINCT(P_CODE)) AVG(P_PRICE)
--------- ----------------------- ------------
    21225                       2         8.47
    21231                       1         8.45
    21344                       3        12.49
    23119                       2        41.97
    24288                       3   155.593333
    25595                       3        89.43
                                2       10.135

7 rows selected.

SQL> SELECT V_CODE, COUNT(DISTINCT (P_CODE)), AVG(P_PRICE)
  2  FROM PRODUCT
  3  GROUP BY V_CODE
  4  HAVING AVG(P_PRICE) < 10;

   V_CODE COUNT(DISTINCT(P_CODE)) AVG(P_PRICE)
--------- ----------------------- ------------
    21225                       2         8.47
    21231                       1         8.45

SQL>
```

Virtual Tables: Creating a View

- View is virtual table based on SELECT query
  - Can contain columns, computed columns, aliases, and aggregate functions from one or more tables
- Base tables are tables on which view is based
- Create view by using CREATE VIEW command

CREATE VIEW viewname AS SELECT query

Virtual Tables: Creating a View (continued)

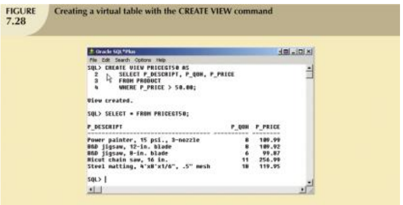


FIGURE 7.28 Creating a virtual table with the CREATE VIEW command

```
SQL> CREATE VIEW PRICEGT50 AS
  2      SELECT P_DESCRIPT, P_QOH, P_PRICE
  3      FROM PRODUCT
  4      WHERE P_PRICE > 50.00;

View created.

SQL> SELECT * FROM PRICEGT50;

P_DESCRIPT                              P_QOH  P_PRICE
----------------------------------- --------- --------
Power painter, 15 psi., 3-nozzle            8   109.99
B&D jigsaw, 12-in. blade                    8   109.92
B&D jigsaw, 8-in. blade                     6    99.87
Hicut chain saw, 16 in.                    11   256.99
Steel matting, 4'x8'x1/6", .5" mesh        18   119.95

SQL> |
```

Relational view has several special characteristics

- Name of view can be used anywhere a table name is expected in a SQL statement
- Views are dynamically updated
- Views provide a level of security