DATABASE SYSTEMS
Design, Implementation, and Management
12e
Carlos Coronel | Steven Morris

Chapter 7

Introduction to Structured Query Language (SQL)

Creating the Database

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

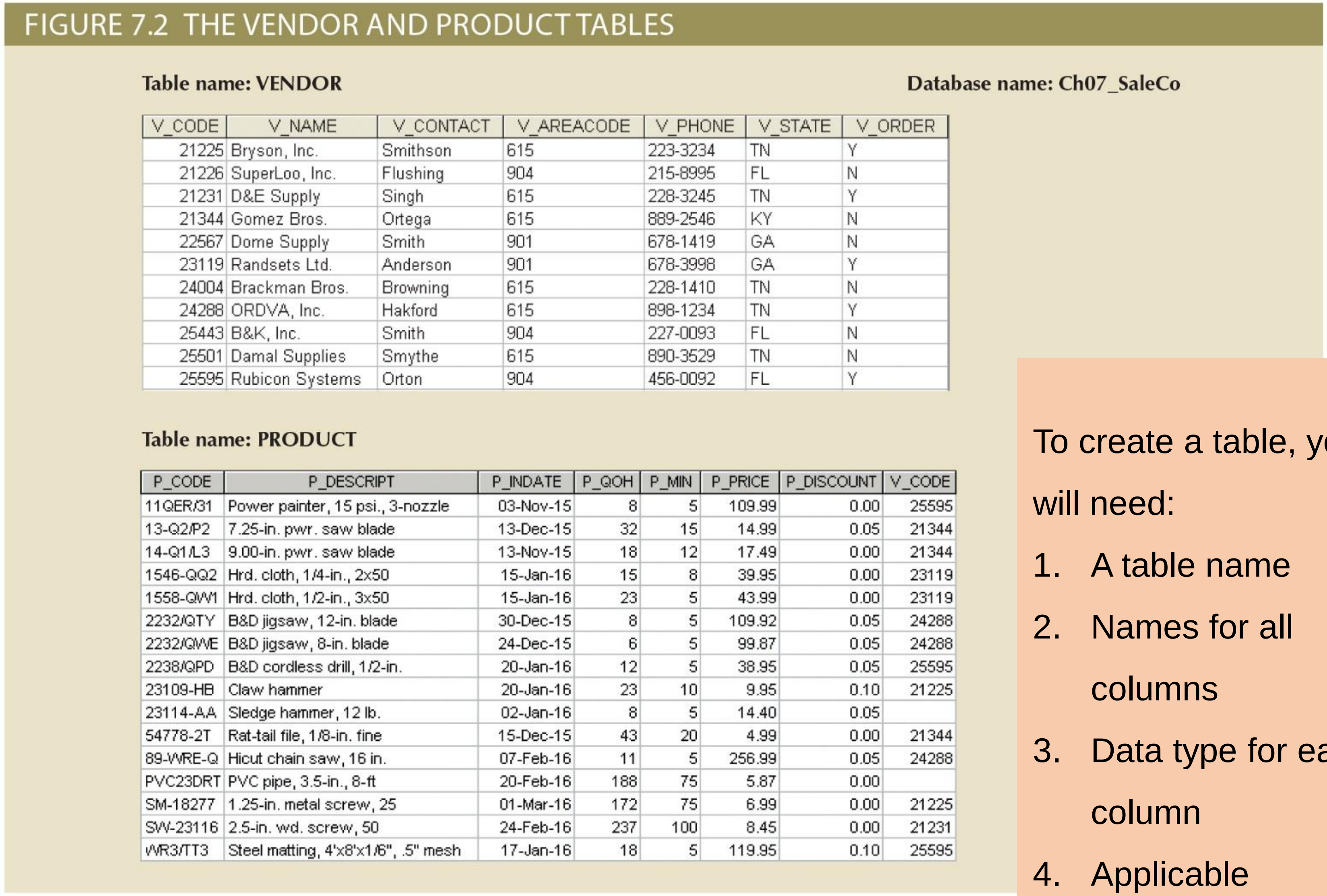


FIGURE 7.2 THE VENDOR AND PRODUCT TABLES

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

To create a table, you will need:

1. A table name
2. Names for all columns
3. Data type for each column
4. Applicable constraints

Table 7.4 - Common SQL Data Types

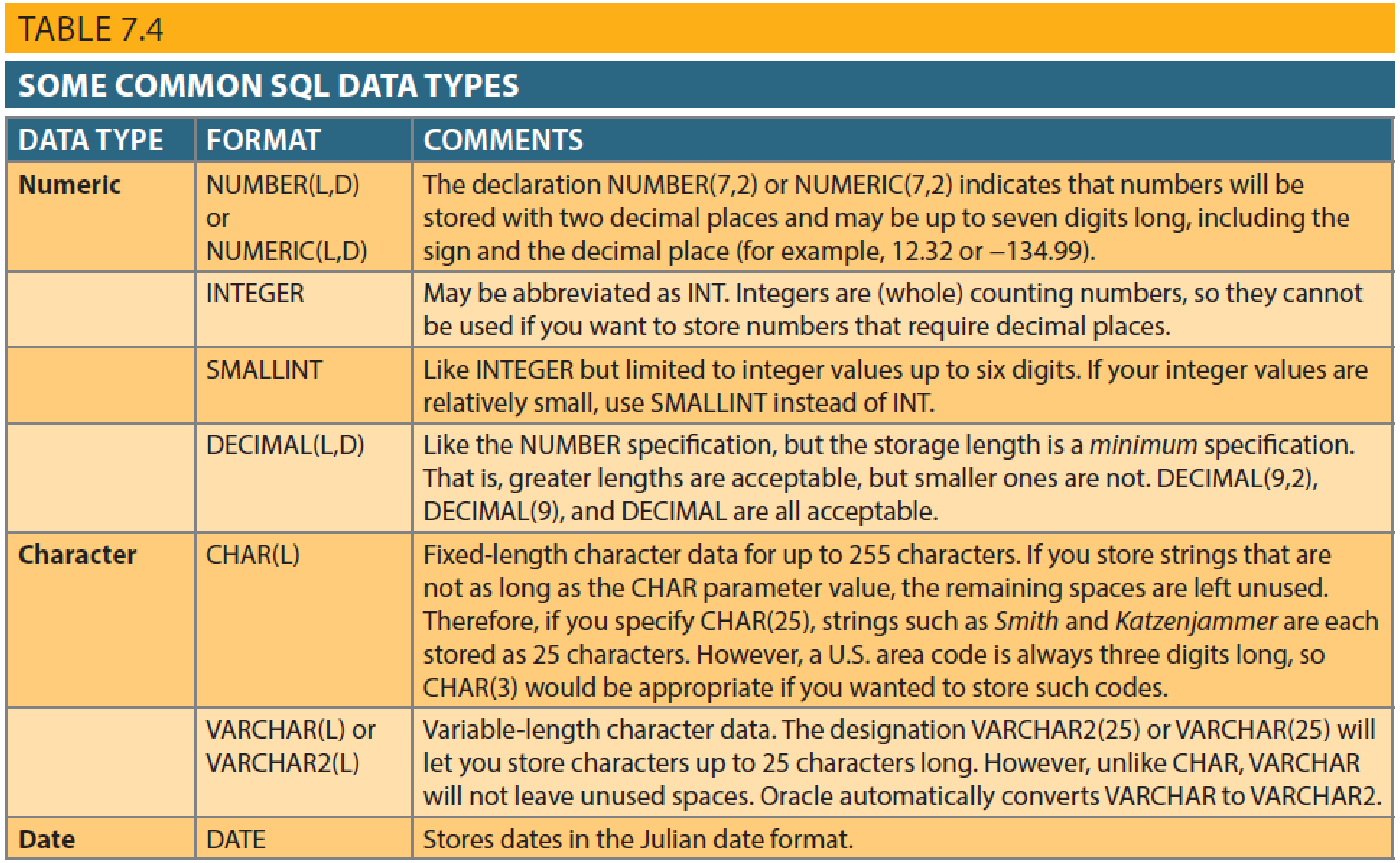


TABLE 7.4

SOME COMMON SQL DATA TYPES

DATA TYPE	FORMAT	COMMENTS
Numeric	NUMBER(L,D) or NUMERIC(L,D)	The declaration NUMBER(7,2) or NUMERIC(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) or VARCHAR(25) will let you store characters up to 25 characters long. However, unlike CHAR, VARCHAR will not leave unused spaces. Oracle automatically converts VARCHAR to VARCHAR2.
Date	DATE	Stores dates in the Julian date format.

TABLE 7.3

DATA DICTIONARY FOR THE CH07_SALECO DATABASE

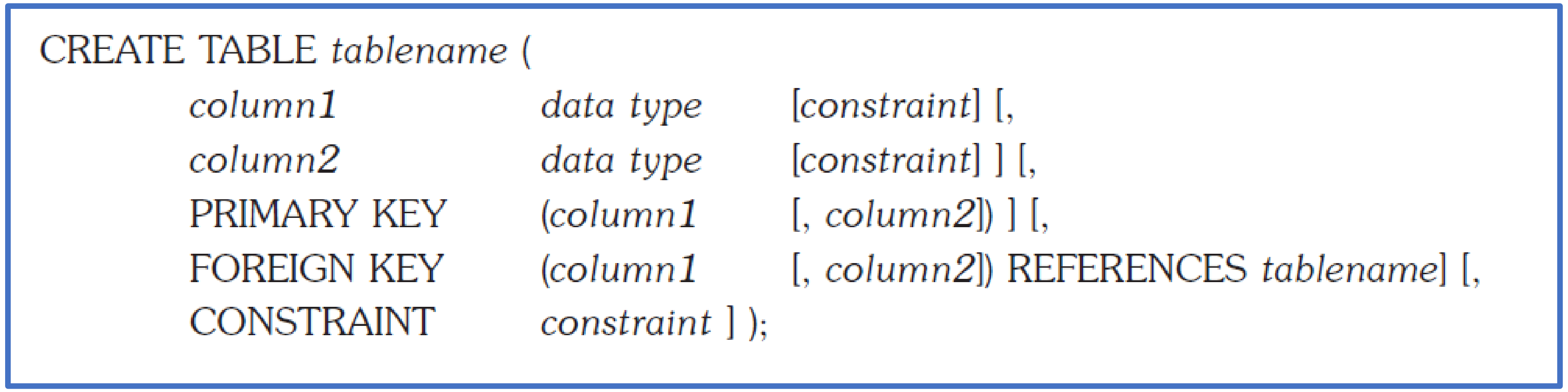
TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
PRODUCT	P_CODE	Product code	VARCHAR(10)	XXXXXXXXXX	NA	Y	PK	
	P_DESCRIPT	Product description	VARCHAR(35)	Xxxxxxxxxxxx	NA	Y		
	P_INDATE	Stocking date	DATE	DD-MON-YYYY	NA	Y		
	P_QOH	Units available	SMALLINT	####	0–9999	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)	Xxxxxxxxxxxxxx	NA	Y		
	V_CONTACT	Contact person	VARCHAR(25)	Xxxxxxxxxxxxxx	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		

FK	= Foreign key
PK	= Primary key
CHAR	= Fixed-length character data, 1 to 255 characters
VARCHAR	= Variable-length character data, 1 to 2,000 characters. VARCHAR is automatically converted to VARCHAR2 in Oracle.
NUMBER	= Numeric data. NUMBER(9,2) is used to specify numbers that have two decimal places and are up to nine digits long, including the decimal places. Some RDBMSs permit the use of a MONEY or a CURRENCY data type.
NUMERIC	= Numeric data. DBMSs that do not support the NUMBER data type typically use NUMERIC instead.
INT	= Integer values only. INT is automatically converted to NUMBER in Oracle.
SMALLINT	= Small integer values only. SMALLINT is automatically converted to NUMBER in Oracle.

DATE formats vary. Commonly accepted formats are DD-MON-YYYY, DD-MON-YY, MM/DD/YYYY, and MM/DD/YY.

*Not all the ranges shown here will be illustrated in this chapter. However, you can use these constraints to practice writing your own.

CREATE TABLE



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

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
VENDOR	V_CODE	Vendor code	INTEGER	#####	1000–9999	Y	PK	
	V_NAME	Vendor name	VARCHAR(35)	Xxxxxxxxxxxxxx	NA	Y		
	V_CONTACT	Contact person	VARCHAR(25)	Xxxxxxxxxxxxxx	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		

```
CREATE TABLE VENDOR (
V_CODE              INTEGER          NOT NULL   UNIQUE,
V_NAME              VARCHAR(35)      NOT NULL,
V_CONTACT           VARCHAR(15)      NOT NULL,
V_AREACODE          CHAR(3)          NOT NULL,
V_PHONE             CHAR(8)          NOT NULL,
V_STATE             CHAR(2)          NOT NULL,
V_ORDER             CHAR(1)          NOT NULL,
PRIMARY KEY (V_CODE));
```

Or, name the constraint

```
CREATE TABLE VENDOR(
V_CODE integer NOT NULL,
V_NAME VARCHAR(35) NOT NULL,
…,
constraint constraintName PRIMARY KEY (V_CODE));
```

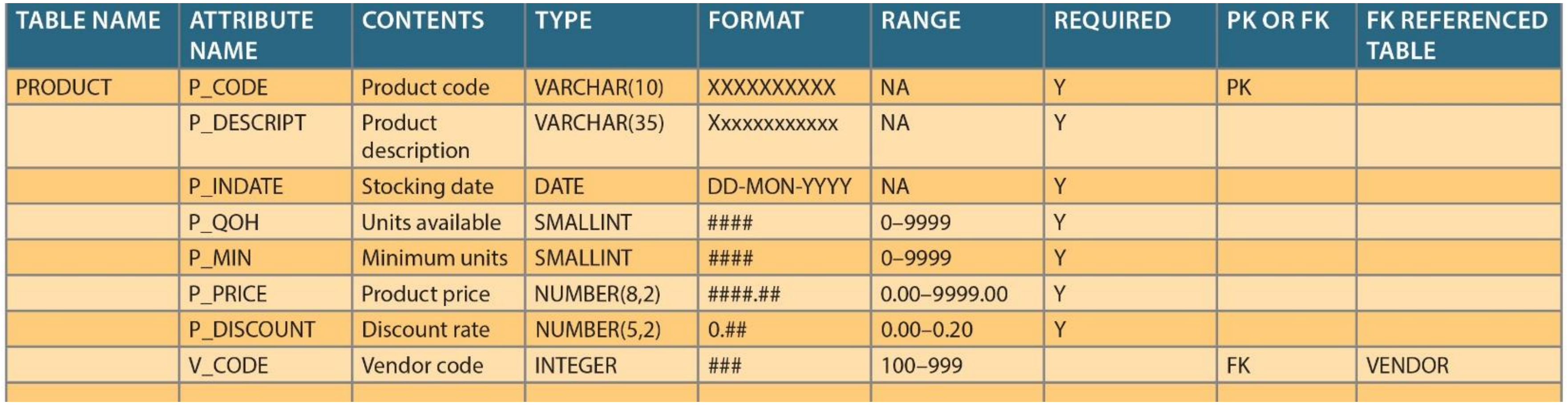
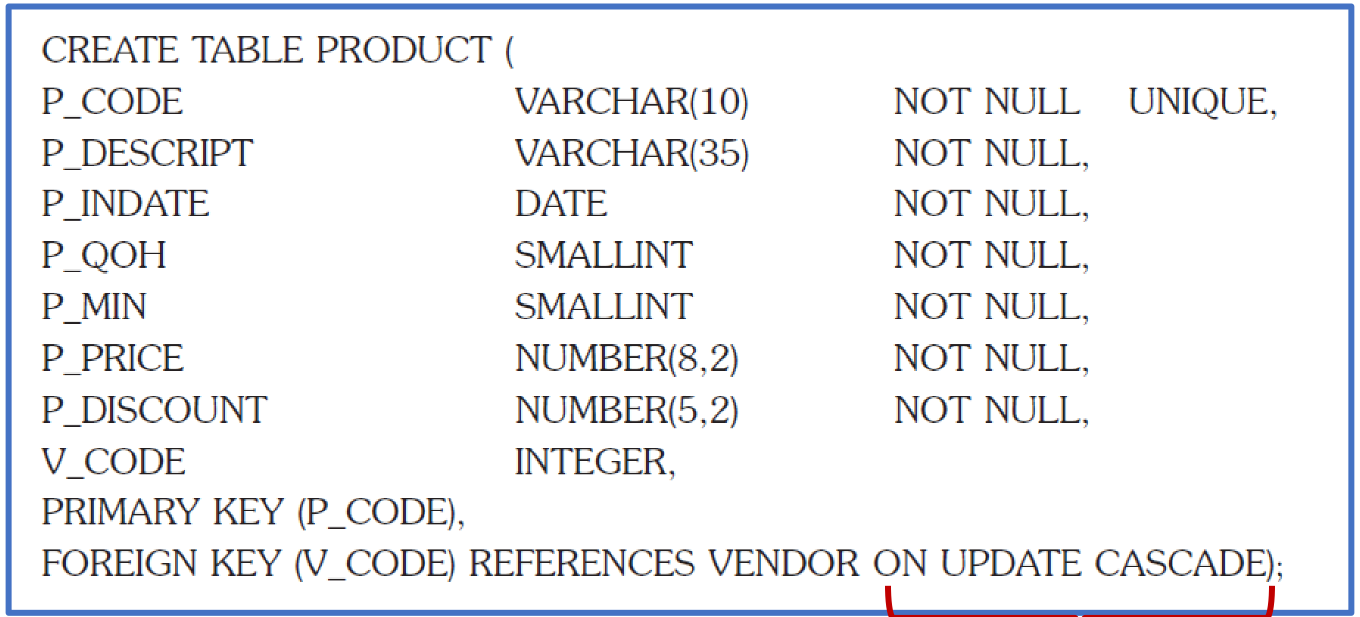


TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
PRODUCT	P_CODE	Product code	VARCHAR(10)	XXXXXXXXXX	NA	Y	PK	
	P_DESCRIPT	Product description	VARCHAR(35)	Xxxxxxxxxxxx	NA	Y		
	P_INDATE	Stocking date	DATE	DD-MON-YYYY	NA	Y		
	P_QOH	Units available	SMALLINT	####	0–9999	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



```
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(5,2)         NOT NULL,
V_CODE              INTEGER,
PRIMARY KEY (P_CODE),
FOREIGN KEY (V_CODE) REFERENCES VENDOR ON UPDATE CASCADE);
```

Delete this part if not supported by your RDBMS.

Creating Table Structures

- Use one line per column (attribute) definition
- Use spaces to line up attribute characteristics and constraints
- Table and attribute names are fully capitalized
- Features of table creating command sequence:
  - NOT NULL specification ensures data entry
  - UNIQUE specification avoids duplicated values
- Table definition enclosed in parentheses
- RDBMS automatically enforces referential integrity for foreign keys.

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

Constraint Examples

```
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(8)         NOT NULL,
CUS_BALANCE       NUMBER(9,2)     DEFAULT 0.00,
CONSTRAINT CUS_UI1 UNIQUE (CUS_LNAME, CUS_FNAME));
```

Data Manipulation Commands

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

Adding Table Rows

INSERT: Command to insert data into table

- Syntax : INSERT INTO tablename [(columnnames)] VALUES (value1, value2, … , valueN);
- Used to add table rows with NULL and NOT NULL attributes

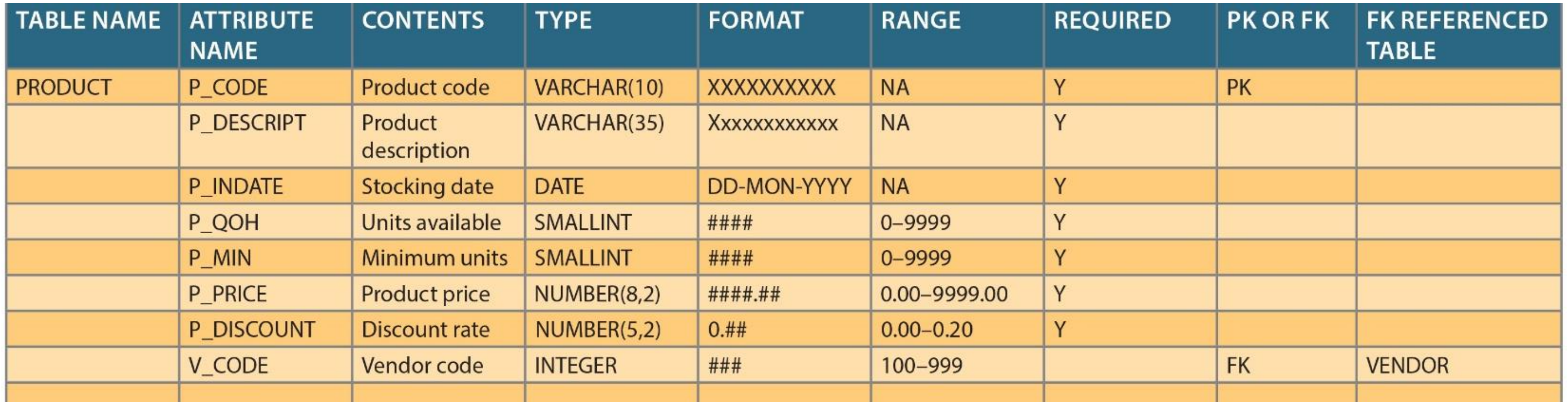


TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
PRODUCT	P_CODE	Product code	VARCHAR(10)	XXXXXXXXXX	NA	Y	PK	
	P_DESCRIPT	Product description	VARCHAR(35)	Xxxxxxxxxxxx	NA	Y		
	P_INDATE	Stocking date	DATE	DD-MON-YYYY	NA	Y		
	P_QOH	Units available	SMALLINT	####	0–9999	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

```
INSERT INTO PRODUCT
      VALUES ('11QER/31','Power painter, 15 psi., 3-nozzle','03-Nov-09',8,5,109.99,0.00,25595);
INSERT INTO PRODUCT
      VALUES ('13-Q2/P2','7.25-in. pwr. saw blade','13-Dec-09',32,15,14.99, 0.05, 21344);
```

```
INSERT INTO PRODUCT
      VALUES ('BRT-345','Titanium drill bit','18-Oct-09', 75, 10, 4.50, 0.06, NULL);
```

```
INSERT INTO PRODUCT(P_CODE, P_DESCRIPT) VALUES ('BRT-345','Titanium drill bit');
```

Save and Restore

COMMIT: Command to save changes

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

ROLLBACK: Command to restore the database

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

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

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

Updating Table Rows

UPDATE: Command to modify data

- 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

```
UPDATE      PRODUCT
SET         P_INDATE = '18-JAN-2010'
WHERE       P_CODE = '13-Q2/P2';
```

```
UPDATE      PRODUCT
SET         P_INDATE = '18-JAN-2010', P_PRICE = 17.99, P_MIN = 10
WHERE       P_CODE = '13-Q2/P2';
```

Deleting Table Rows

DELETE: Command to delete

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

```
DELETE FROM          PRODUCT
WHERE                P_CODE = 'BRT-345';
```

- WHERE condition is optional
- If WHERE condition is not specified, all rows from specified table will be deleted

More SQL commands

The Database Schema

- Logical group of database objects – such as tables and indexes - related to each other
- Command:
  - CREATE SCHEMA AUTHORIZATION {creator};
  - Seldom used directly as command is usually optional

Additional 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 a Column’s Data Type and Data Characteristics

- ALTER used to change data type and characteristics
  - Some RDBMSs do not permit changes to data types unless column is empty
  - Changes in characteristics are permitted if they do not alter the existing data type
- Syntax:
  - Data Type: ALTER TABLE *tablename* MODIFY *(columnname(datatype))*;
  - Data Characteristic: ALTER TABLE *tablename* MODIFY *(columnname(characteristic))*;

Adding and Dropping Columns

- 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

- A created new table based on another table does not include old table’s integrity rule (no primary key)
- Can re-establish integrity rules using **ALTER** command
- Use **ALTER TABLE** command to ADD primary and foreign keys
  - Composite primary keys and multiple foreign keys can be designated in a single SQL command

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