Database Principles: Fundamentals of Design, Implementation, and Management

Tenth Edition

Beginning Structured Query Language

Data Definition Language

(SQL-DDL)

Objectives

In this chapter, students will learn:

- Introduction to SQL
- The basic of DDL commands
- How to use DDL command to create tables and indexes.
- How to use DDL command to modify table structures.
- How to use DDL command to modify table constraint s.

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
- Current version : SQL-2003 (ANSI/ISO std)
- Several SQL dialects exist.

  (e.g. Oracle, MySQL, DB2, MS SQL Server)

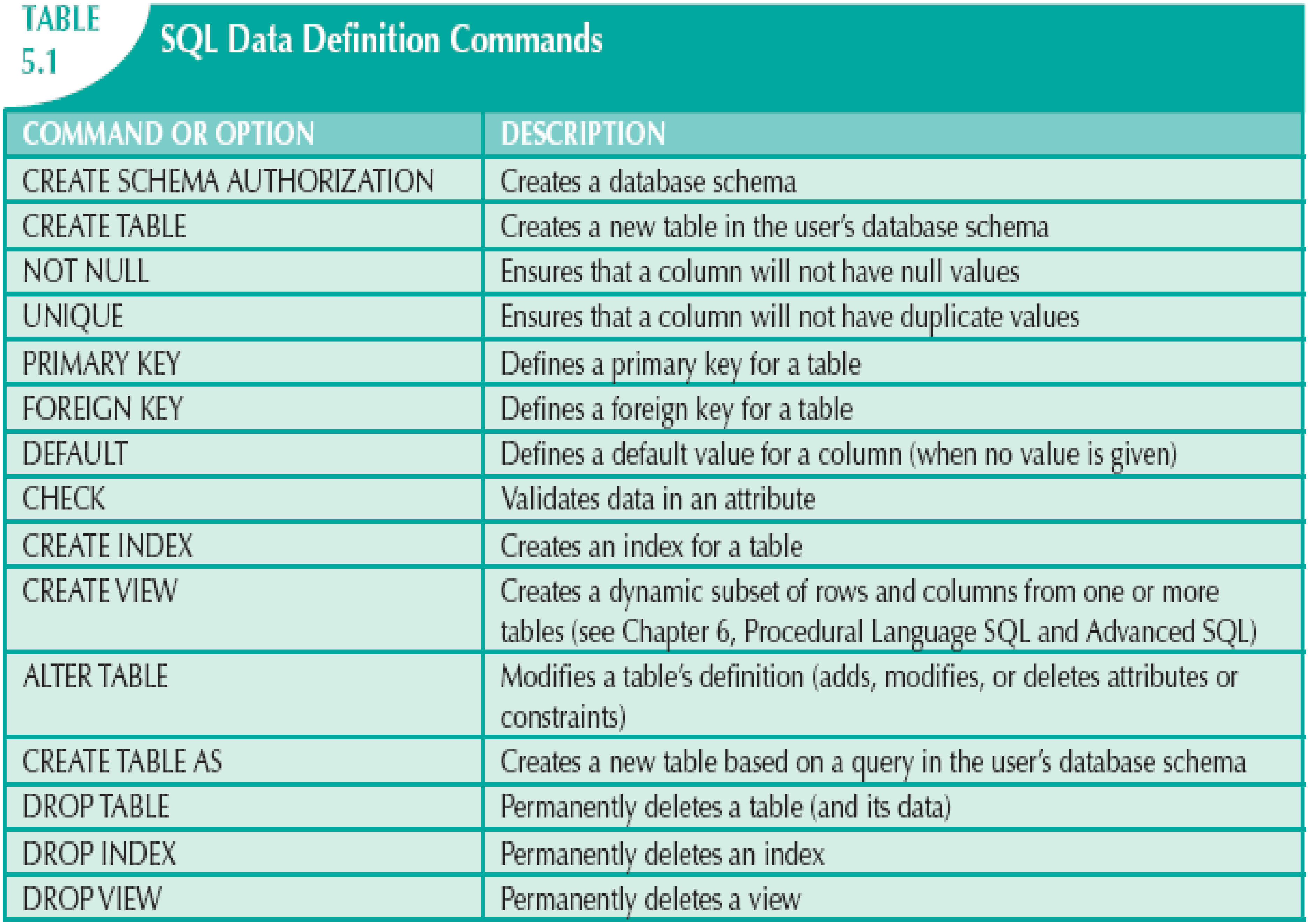


TABLE 5.1 SQL Data Definition Commands

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 6, Procedural Language SQL and 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

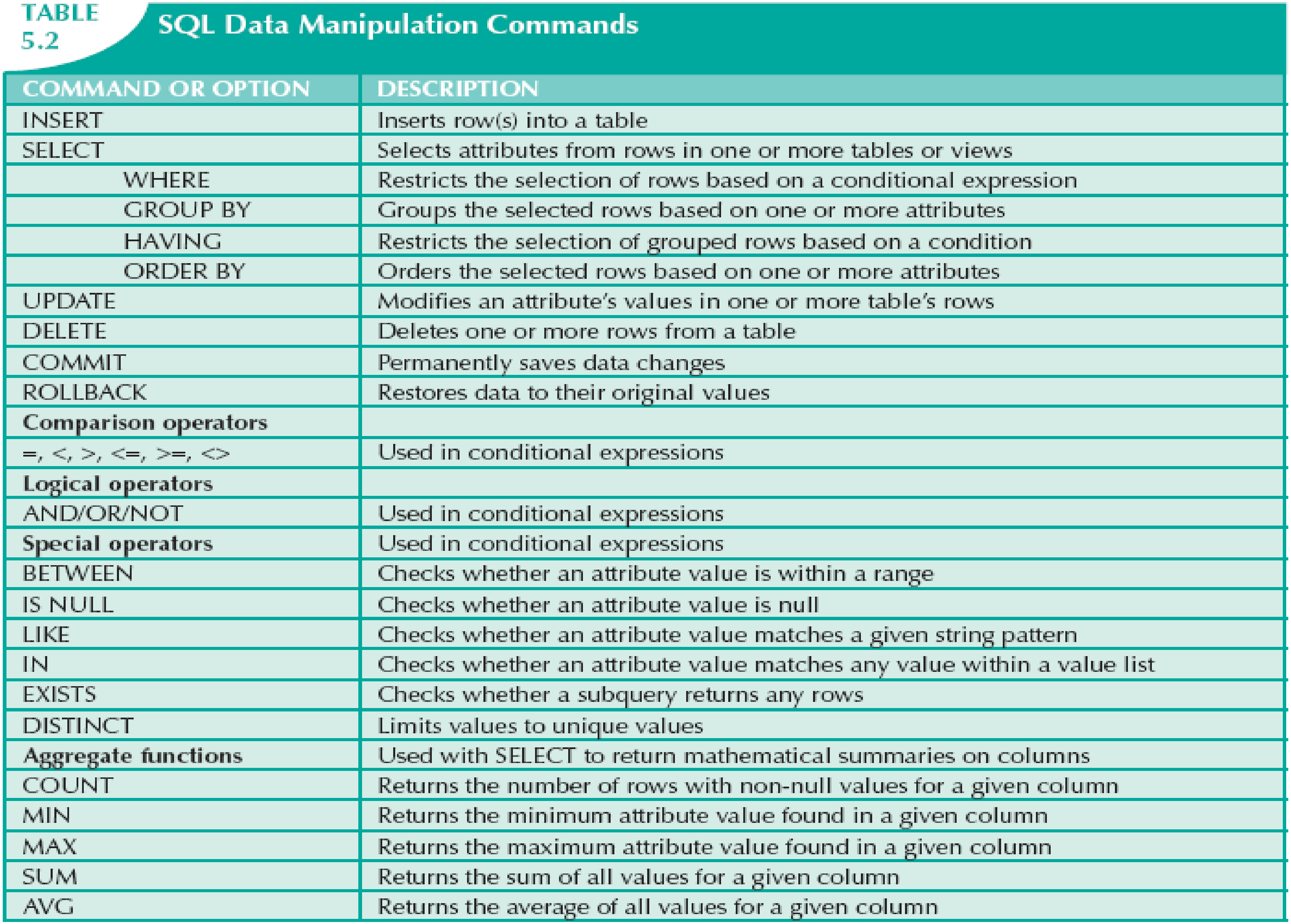


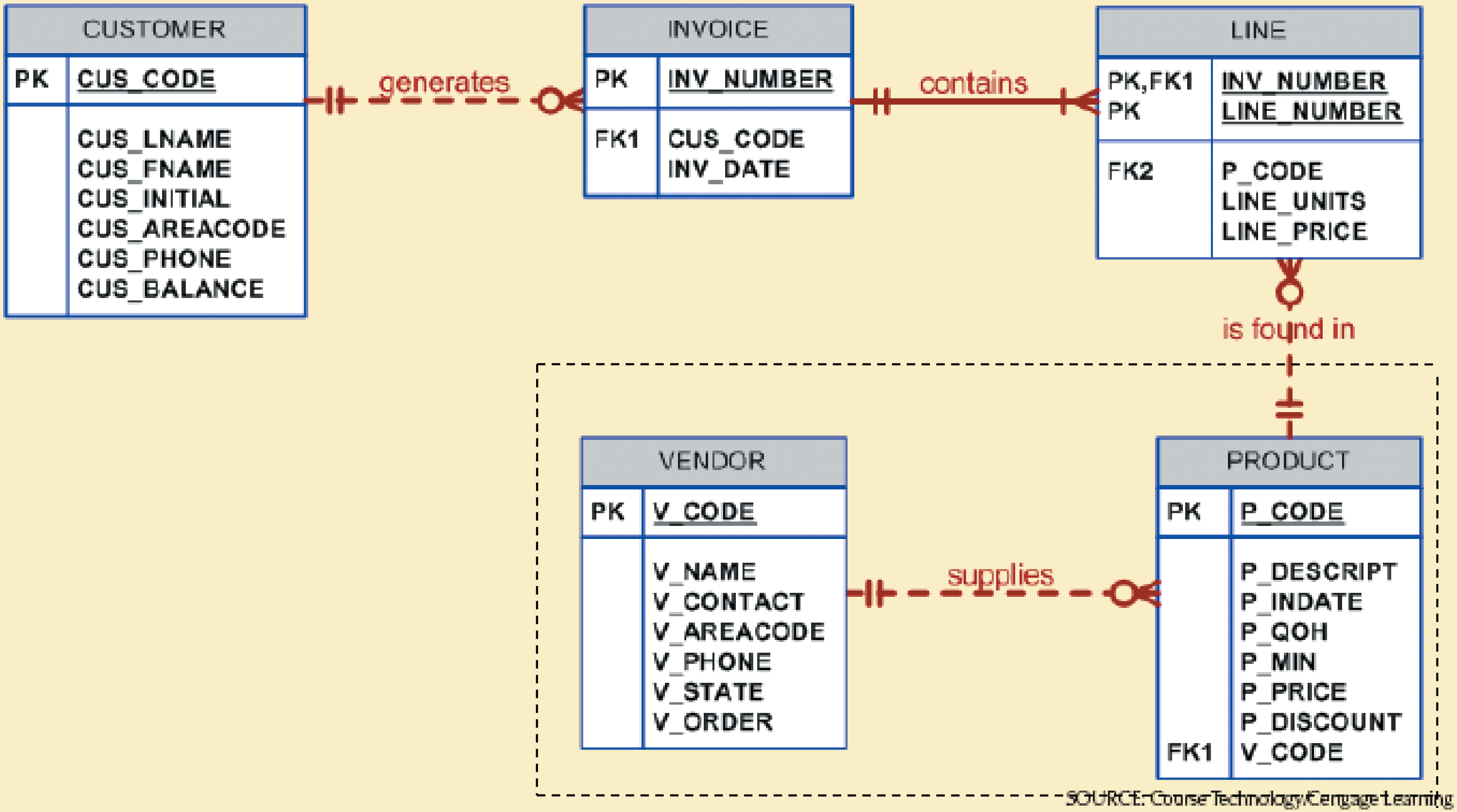
TABLE 5.2 SQL Data Manipulation Commands

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 DDL commands:
    - CUSTOMER
    - INVOICE
    - LINE
    - PRODUCT
    - VENDOR
  - Focus on PRODUCT and VENDOR tables

FIGURE 5.1 The database model



CUSTOMER	
PK	CUS_CODE
	CUS_LNAME CUS_FNAME CUS_INITIAL CUS_AREACODE CUS_PHONE CUS_BALANCE

INVOICE	
PK	INV_NUMBER
FK1	CUS_CODE INV_DATE

LINE	
PK,FK1 PK	INV_NUMBER LINE_NUMBER
FK2	P_CODE LINE_UNITS LINE_PRICE

VENDOR	
PK	V_CODE
	V_NAME V_CONTACT V_AREACODE V_PHONE V_STATE V_ORDER

PRODUCT	
PK	P_CODE
      FK1	P_DESCRIPT P_INDATE P_QOH P_MIN P_PRICE P_DISCOUNT V_CODE

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

- 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

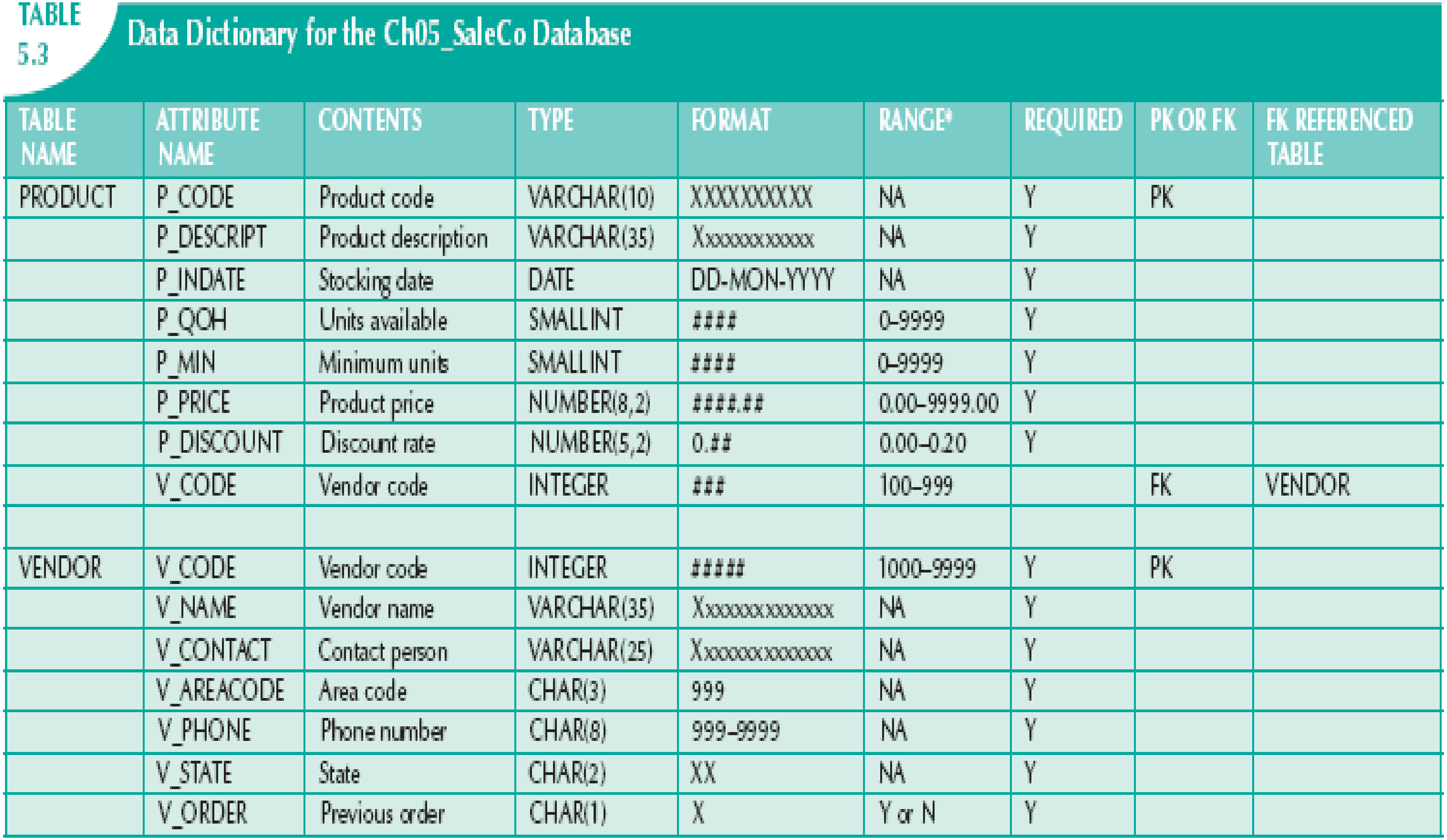


TABLE 5.3 Data Dictionary for the Ch05_SaleCo Database

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		

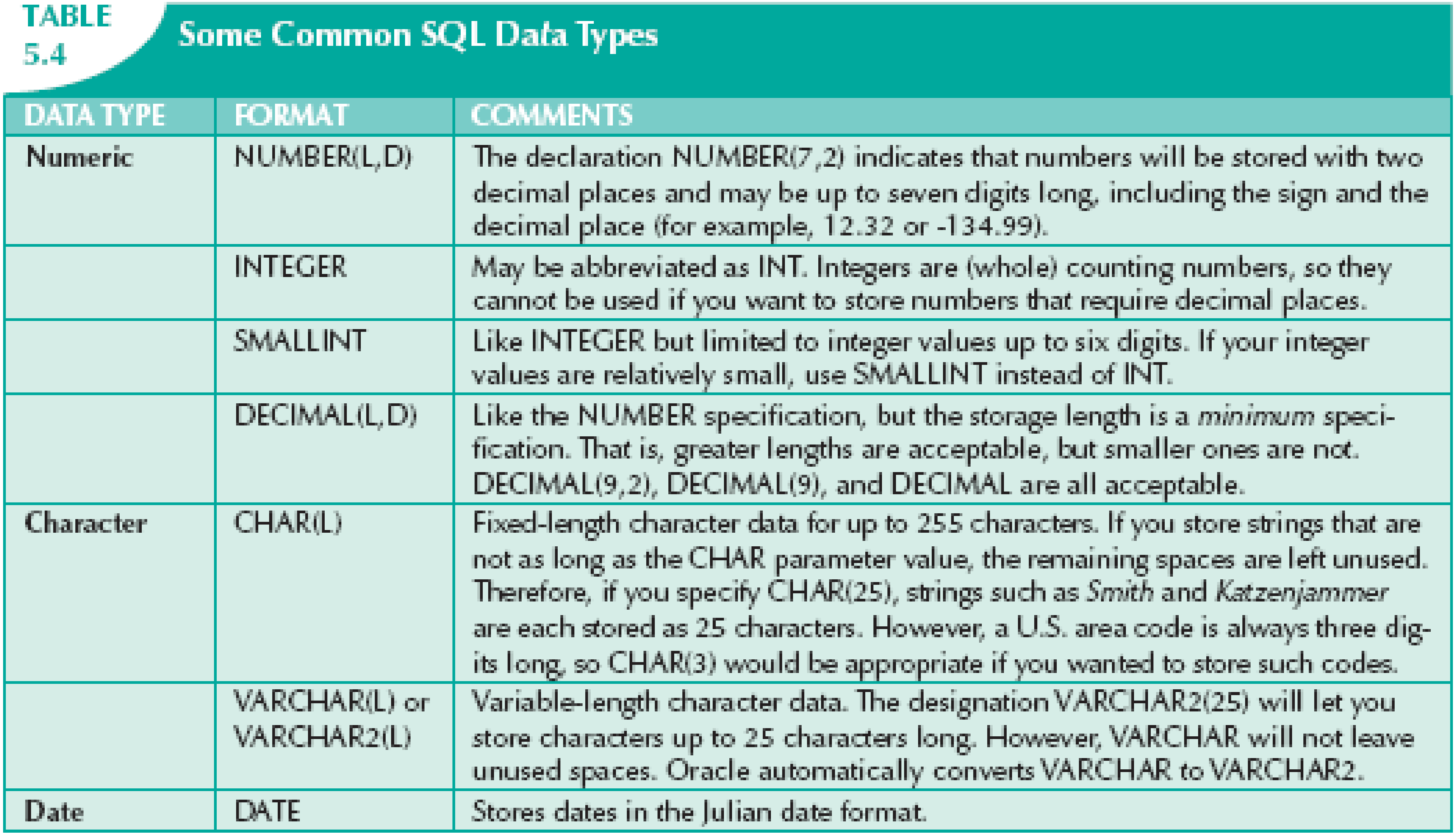


TABLE 5.4 Some Common SQL Data Types

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 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

- 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

DDL for VENDOR table

```
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)
CHECK (V_CODE>=1000 AND V_CODE<=9999),
CHECK (V_ORDER=‘Y’ OR V_ORDER=‘N’));
```

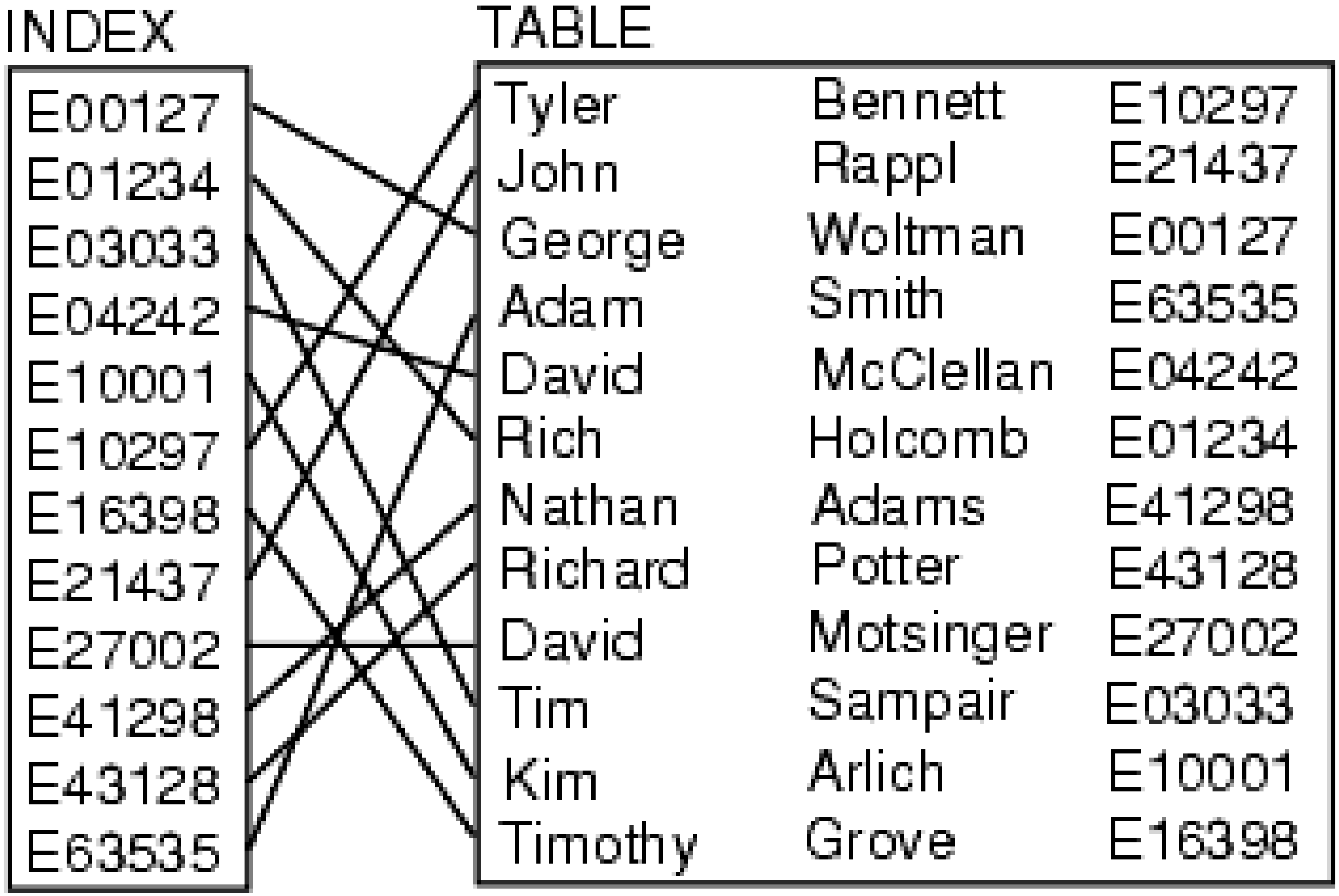
DDL for PRODUCT table

```
CREATE TABLE  PRODUCT (
P_CODE                VARCHAR2(10)    NOT NULL    UNIQUE,
P_DESCRIPT            VARCHAR2(35)    NOT NULL,
P_INDATE              DATE            NOT NULL,
P_QOH                 NUMBER          NOT NULL,
P_MIN                 NUMBER          NOT NULL,
P_PRICE               NUMBER(8,2)     NOT NULL,
P_DISCOUNT            NUMBER(5,2)     DEFAULT 0.00,
V_CODE                INTEGER,
PRIMARY KEY (P_CODE),
FOREIGN KEY (V_CODE) REFERENCES VENDOR (V_CODE)
CHECK (P_DISCOUNT>=0.00 AND P_DISCOUNT<=0.20);
```

Indexes

- An index is a database structure that you can use to improve the performance of database activity.
- When primary key is declared, DBMS automatically creates unique index.
- A database table can have one or more indexes associated with it.

Indexes



INDEX

E00127
E01234
E03033
E04242
E10001
E10297
E16398
E21437
E27002
E41298
E43128
E63535

TABLE

Tyler	Bennett	E10297
John	Rappl	E21437
George	Woltman	E00127
Adam	Smith	E63535
David	McClellan	E04242
Rich	Holcomb	E01234
Nathan	Adams	E41298
Richard	Potter	E43128
David	Motsinger	E27002
Tim	Sampair	E03033
Kim	Arlich	E10001
Timothy	Grove	E16398

Indexes

- Index on any selected attribute can be created using CREATE INDEX command.

e.g.

```
CREATE INDEX  V_NAME_IDX
ON VENDOR (V_NAME)
```

- Composite index
  - – Index based on two or more attributes
  - – Often used to prevent data duplication
    - use CREATE **UNIQUE** INDEX command

Modifying Table Structures

- 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.

e.g. change V_CODE data type from integer to character.

```
ALTER TABLE PRODUCT
MODIFY (V_CODE CHAR(5));
```

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

e.g. increase the width of P_PRICE from five digit to nine digit.

```
ALTER TABLE PRODUCT
MODIFY (P_PRICE (9,2));
```

Adding a Column

- Use ALTER to add column
  - Do not include the NOT NULL clause for new column

e.g. add P_SALECODE attribute to PRODUCT table.

```
ALTER TABLE PRODUCT
ADD (P_SALECODE CHAR(1));
```

Dropping a Column

- Use ALTER to drop column
  - Some RDBMSs impose restrictions on the deletion of an attribute

e.g. delete V_ORDER attributes from VENDOR table

```
ALTER TABLE VENDOR
DROP COLUMN V_ORDER;
```

Deleting a Table

- Use DROP TABLE command to delete all data and structure in the table.

e.g. delete table VENDOR with all it’s data.

DROP TABLE VENDOR

Summary

- SQL commands can be divided into two overall categories:
  - Data definition language commands (DDL)
  - Data manipulation language commands (DML)
- 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

- Use CREATE TABLE command to create new table structure and constraint.
- Use ALTER TABLE command to modify table structure and constraint.
- Types of constraint: PK, FK, NOT NULL, UNIQUE, DEFAULT and CHECK.
- Use DROP TABLE command to delete all data and structure of a table.

Exercise 1

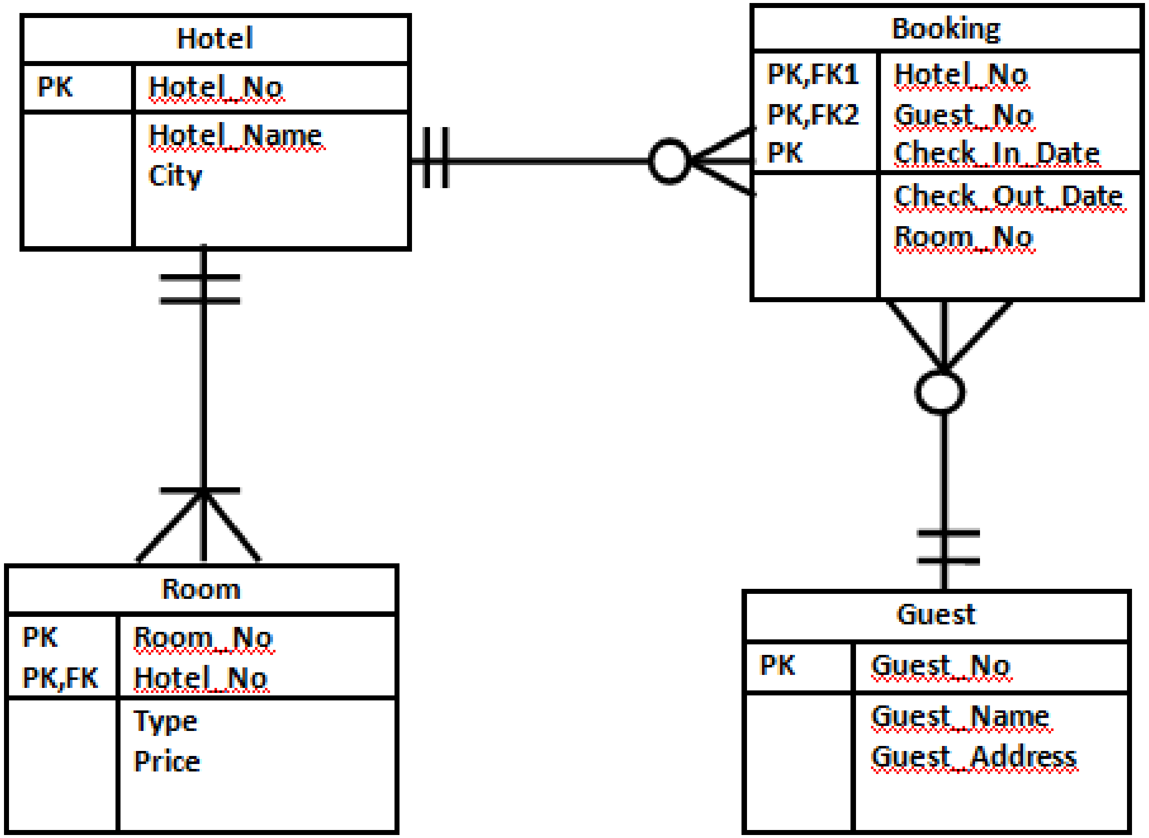
Using the same database model from this lecture

- Construct the Data Dictionary for Customer, Invoice and Line tables.

- Write the DDL statement to create the table structure and constraint for Customer, Invoice and Line tables.

Exercise 2

Hotel Database Model



Hotel	
PK	Hotel_No
	Hotel_Name City

Booking	
PK,FK1 PK,FK2 PK	Hotel_No Guest_No Check_In_Date
	Check_Out_Date Room_No

Room	
PK PK,FK	Room_No Hotel_No
	Type Price

Guest	
PK	Guest_No
	Guest_Name Guest_Address

Exercise 2 (continue..)

The tables in Hotel Database has the following constraints:

a) Room Type must be one of Single, Double or Family.

b) Room Price must be between 100 and 500.

c) Room_No must be between 1 and 100

d) Check_In_Date and Check_Out_Date must be greater than today's date.

e) The same room cannot be double-booked

f) The same quest cannot have overlapping

Exercise 2 (continue..)

- Construct the data dictionary for all tables in the Hotel Database.
- Write the DDL statement to create all table structures and constraints in the Hotel Database.