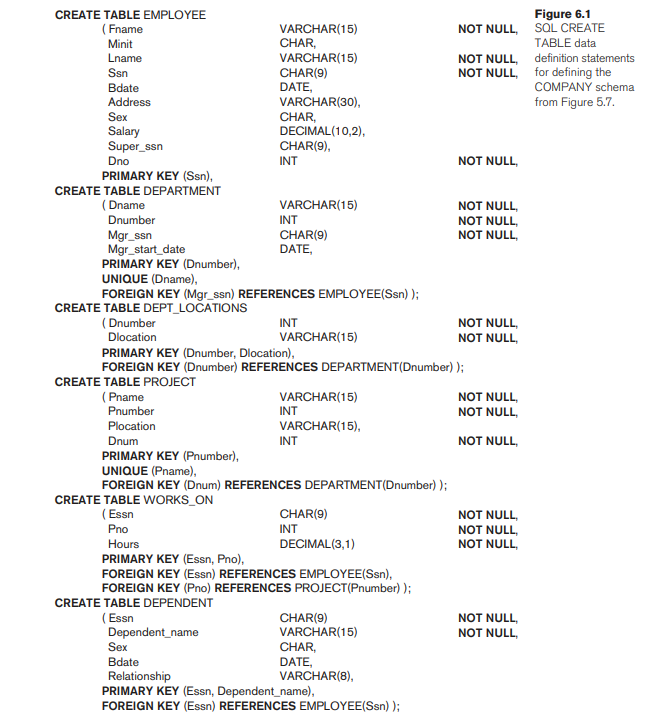
**The CREATE TABLE Command in SQL**

The CREATE TABLE command is used to specify a new relation by giving it a name and specifying its attributes and initial constraints. The attributes are specified first, and each attribute is given a name, a data type to specify its domain of values, and possibly attribute constraints, such as NOT NULL. The key, entity integrity, and referential integrity constraints can be specified within the CREATE TABLE statement after the attributes are declared, or they can be added later using the ALTER TABLE command.





**Attribute Data Types and Domains in SQL**

The basic data types available for attributes include numeric, character string, bit string, Boolean, date, and time.

Numeric data types include integer numbers of various sizes (INTEGER or INT, and SMALLINT) and floating-point (real) numbers of various precision (FLOAT or REAL, and DOUBLE PRECISION). Formatted numbers can be declared by using DECIMAL (i, j) or DEC(i, j) or NUMERIC(i, j)—where i, the precision, is the total number of decimal digits and j, the scale, is the number of digits after the decimal point.



Character-string data types are either fixed length—CHAR(n) or CHARACTER(n), where n is the number of characters—or varying length—VARCHAR(n) or CHAR VARYING(n) or CHARACTER VARYING(n), where n is the maximum number of characters.



Bit-string data types are either of fixed length n—BIT(n)—or varying length—BIT VARYING(n), where n is the maximum number of bits.



A Boolean data type has the traditional values of TRUE or FALSE.

The DATE data type has ten positions, and its components are YEAR, MONTH, and DAY in the form YYYY-MM-DD.



The TIME data type has at least eight positions, with the components HOUR, MINUTE, and SECOND in the form HH:MM:SS.

A timestamp data type (TIMESTAMP) includes the DATE and TIME fields, plus a minimum of six positions for decimal fractions of seconds and an optional WITH TIME ZONE qualifier.

Literal values are represented by single-quoted strings preceded by the keyword TIMESTAMP, with a blank space between data and time; for example, TIMESTAMP ‘2014-09-27 09:12:47.648302’.

**Specifying Constraints in SQL**

**Specifying Attribute Constraints and Attribute Defaults**

SQL allows NULLs as attribute values, a constraint NOT NULL may be specified if NULL is not permitted for a particular attribute.

It is also possible to define a default value for an attribute by appending the clause DEFAULT <value> to an attribute definition.

CREATE TABLE EMPLOYEE

( … ,

Dno INT NOT NULL DEFAULT 1,

PRIMARY KEY (Ssn),

FOREIGN KEY (Super\_ssn) REFERENCES EMPLOYEE(Ssn)

FOREIGN KEY(Dno) REFERENCES DEPARTMENT(Dnumber));

CREATE TABLE DEPARTMENT

( … ,

Mgr\_ssn CHAR(9) NOT NULL DEFAULT ‘888665555’,

… ,

PRIMARY KEY(Dnumber),

FOREIGN KEY (Mgr\_ssn) REFERENCES EMPLOYEE(Ssn));

CREATE TABLE DEPT\_LOCATIONS

( … ,

PRIMARY KEY (Dnumber, Dlocation),

FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber));

For example, suppose that department numbers are restricted to integer numbers between 1 and 20; then, we can change the attribute declaration of Dnumber in the DEPARTMENT table (see Figure 6.1) to the following:

Dnumber INT NOT NULL CHECK (Dnumber > 0 AND Dnumber < 21);

**Specifying Constraints on Tuples Using CHECK**

In addition to key and referential integrity constraints, table constraints can be specified through additional CHECK clauses at the end of a CREATE TABLE statement.

These can be called row-based constraints because they apply to each row individually and are checked whenever a row is inserted or modified.

For example, suppose that the DEPARTMENT table , additional attribute Dept\_create\_date, which stores the date when the department was created. Then we could add the following CHECK clause at the end of the CREATE TABLE statement for the DEPARTMENT table to make sure that a manager’s start date is later than the department creation date.

CHECK (Dept\_create\_date <= Mgr\_start\_date);