# Basic SQL

What is SQL?

* + Structured Query Language.
  + Used in most of relational databases, i.e. databases based on RM model.
  + It is a language; set of commands or statements.
  + Each statement ends with a semicolon **;**
  + It is a comprehensive language, is both DDL and DML.

Terminology:

* **Table**, **row**, and **column** used for terms **relation, tuple**, and **attribute** in relational models.

CREATE statement

* + Main SQL command for **defining database, table, view, etc.**

## First create a database/schema

What is a schema (based on chapter 5th definition)?

* + Set of relations with a set of constraints.
  + Schema is synonym for database here.
  + Relations == tables
  + Database is a set of tables.
* SQL statement for creating a database/schema:

CREATE SCHEMA db\_name ;

CREATE DATABASE db\_name ;

* On a multiuser DBMS, you need to specify which user owns a database.

CREATE DATABASE db\_name AUTHORIZATION ‘user1’;

## Second create tables

* SQL statement:

CREATE TABLE table\_name (

col1\_name col1\_datatype col1\_properties,

col2\_name col2\_datatype col2\_properties,

.

.

.

colN\_name colN\_datatype colN\_properties

);

You can optionally specify database name:

CREATE TABLE COMPANY.EMPLOYEE ...

is equivalent to

CREATE TABLE EMPLOYEE ...

## Attribute Data Types

#### Basic data types

* + **Numeric** data types
    - 1. Integer numbers: SMALLINT, INT, and BIGINT; 2, 4 and 8 bytes respectively.
      2. Floating-point (real) numbers: FLOAT or DOUBLE
  + Character-string data types
    1. Fixed length: CHAR(*n*)
    2. Varying length: VARCHAR(*n*)
    3. What is the difference between these two?
  + **Bit** data types
    1. Bit 0 or 1
    2. Fixed length: BIT(n)
       1. Varying length: BIT VARYING (n)
  + **DATE** data type

Components are YEAR, MONTH, and DAY in the form YYYY-MM-DD

Multiple mapping functions available in DBMSs to change date formats (to US or EU formats).

#### Additional data types

* + DATETIME; Format: YYYY-MM-DD HH:MI:SS
  + TIME; Format: HH:MI:SS
  + TIMESTAMP; Format: YYYY-MM-DD HH:MI:SS

TIMESTAMP automatically set itself to the current date and time.

* + BLOB (Binary Large Object) for storing images, files, etc.

#### Your own data types

* + You can also specify your own data type or domain, and then use it for defining table columns;

**CREATE DOMAIN** SSN\_TYPE **AS** CHAR(9);

You can use SSN\_TYPE in place of CHAR(9) in Figure 6.1 for Ssn and Super\_ssn of EMPLOYEE, Mgr\_ssn of DEPARTMENT, Essn of WORKS\_ON, and Essn of DEPENDENT.

# Specifying Constraints in SQL

**Basic constraints:** Relational Model has 3 basic constraint types that are supported in SQL:

* 1. **Key** constraint: A primary key value cannot be duplicated
  2. **Entity Integrity** constraint: A primary key value cannot be null
  3. **Referential integrity** constraints: The foreign key must have a value that is already present as a primary key, or may be null.

## Specifying Attribute Constraints

Restrictions on attribute domains:

* + Default value of an attribute

**DEFAULT** <value>

* + NULL is not permitted for a particular attribute (NOT NULL)
  + **CHECK** clause

Example:

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

## Specifying Key and Referential Integrity Constraints

* **PRIMARY KEY** clause
  1. Specifies one or more attributes that make up the primary key of a relation
  2. Dnumber INT PRIMARY KEY,
* **UNIQUE** clause
  1. Specifies alternate (secondary) keys (called CANDIDATE keys in the relational model).
  2. Dname VARCHAR(15) UNIQUE,
* **FOREIGN KEY** clause
  1. With this clause you can specify the default actions if integrity constraints get violated.
     + Actions include SET NULL, CASCADE, and SET DEFAULT
     + Actions taken by the DBMS for SET NULL or SET DEFAULT is the same for both ON DELETE and ON UPDATE.
* Using the Keyword **CONSTRAINT**
  + You can name a constraint
  + Naming a constraint is useful when you want to change or delete constraints later.

Create Table SQL statement:

CREATE TABLE table\_name (

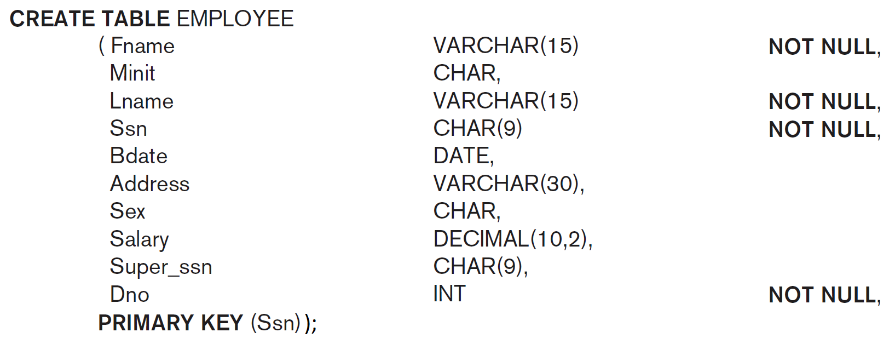
col1\_name col1\_datatype col1\_properties,

.

colN\_name colN\_datatype colN\_properties,

**define constraints here**

);

Example:

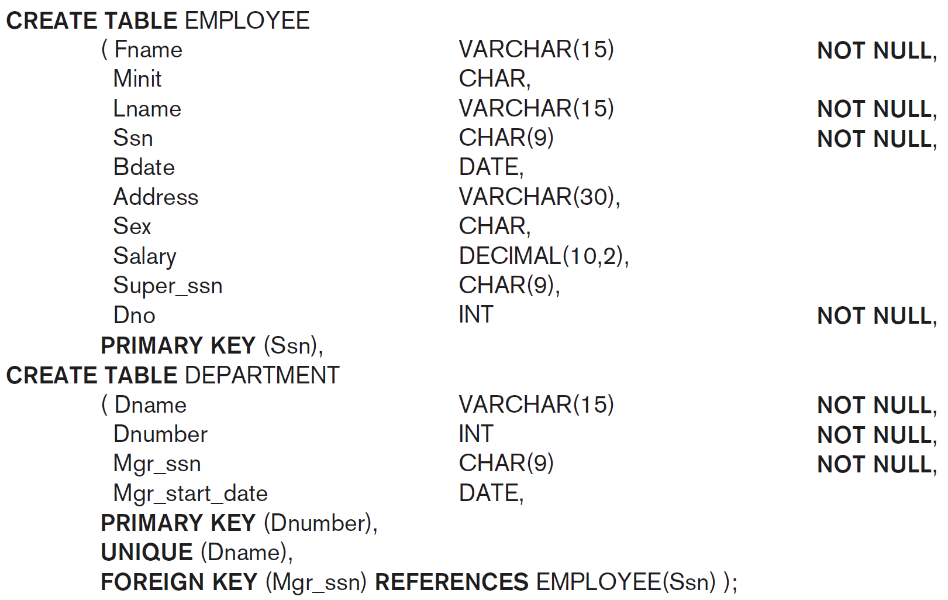
Multiple create statements could be combined as one big statement:

Separated with ‘,’

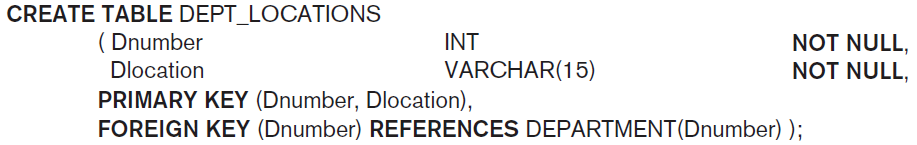
Remove ‘;’ from each.

Add one ‘;’ at end.

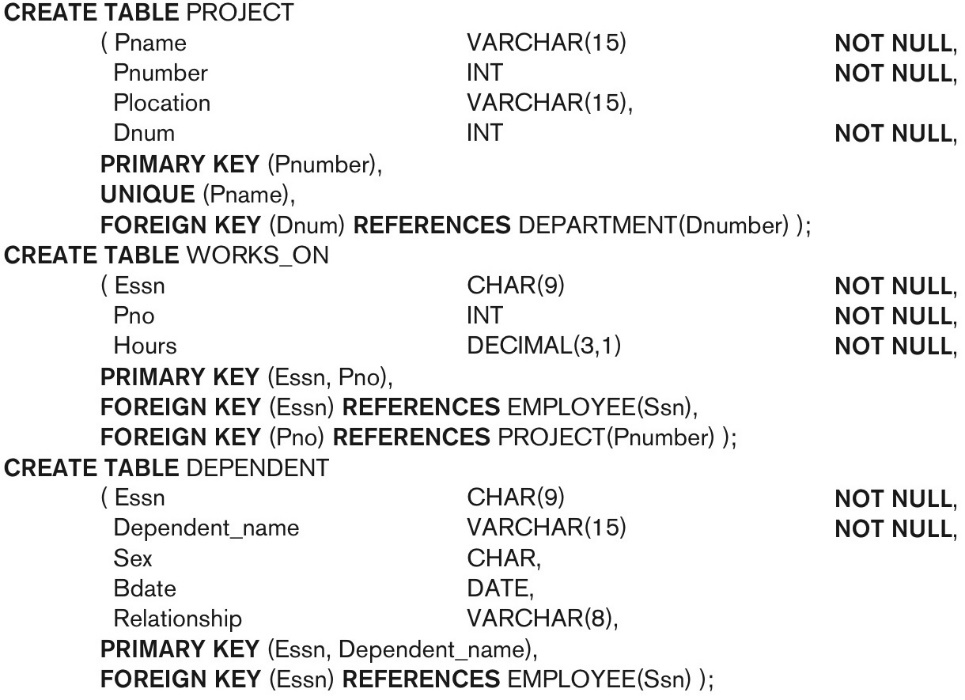
See Fig 6.1 in the book.



###### Example of compound primary key:



##### The other COMPANY tables



Default attribute values and referential integrity triggered action specification (Fig. 6.2)

