Basic SQL

Part 2

Specifying Constraints in SQL

Specifying Attribute Constraints and Attribute Defaults

- Because SQL allows NULLs as attribute values, a *constraint* NOT NULL may be specified if NULL is not permitted for a particular attribute.
- This is always implicitly specified for the attributes that are part of the primary key of each relation, but it can be specified for any other attributes whose values are required not to be NULL.

- It is also possible to define a *default value* for an attribute by appending the clause **DEFAULT** <value> to an attribute definition.
- The default value is included in any new tuple if an explicit value is not provided for that attribute.
- If no default clause is specified, the default *default value* is NULL for attributes *that do not have* the NOT NULL constraint.

- Another type of constraint can restrict attribute or domain values using the **CHECK** clause following an attribute or domain definition.
  - Dnumber INT **NOT NULL CHECK** (Dnumber > 0 **AND** Dnumber < 21);
  - **CREATE DOMAIN** D_NUM **AS** INTEGER CHECK (D_NUM > 0 **AND** D_NUM < 21);

Specifying Key and Referential Constraints

- The **PRIMARY KEY** clause specifies one or more attributes that make up the primary key of a relation.
- If a primary key has a *single* attribute, the clause can follow the attribute directly.
  - `Dnumber INT` **`PRIMARY KEY`**`;`
- In case the primary key consists of two or more columns, you define the primary key constraint as follows:
  - **`PRIMARY KEY`** `(Dnumber, Dlocation)`

- The **UNIQUE** clause specifies alternate (secondary) keys.
- The UNIQUE clause can also be specified directly for a secondary key if the secondary key is a single attribute:
  - `Dname VARCHAR(15)` **`UNIQUE`**`;`
- In case the secondary key consists of two or more columns, you define the unique constraint as follows:
  - **`UNIQUE`** `(`*`attribute1`*`, `*`attribute2`*`)`

- Referential integrity is specified via the **FOREIGN KEY** clause.
- A referential integrity constraint can be violated when tuples are inserted or deleted, or when a foreign key or primary key attribute value is modified.
- The default action that SQL takes for an integrity violation is to reject the update operation that will cause a violation, which is known as the `RESTRICT` option.
- However, the schema designer can specify an alternative action to be taken by attaching a referential triggered action clause to any foreign key constraint.
- The options include `SET NULL`, `CASCADE`, and `SET DEFAULT`.
- An option must be qualified with either `ON DELETE` or `ON UPDATE`.

```
CREATE TABLE EMPLOYEE
      (   . . . ,
          Dno          INT                 NOT NULL          DEFAULT 1,
      CONSTRAINT EMPPK
          PRIMARY KEY (Ssn),
      CONSTRAINT EMPSUPERFK
          FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn)
                              ON DELETE SET NULL           ON UPDATE CASCADE,
      CONSTRAINT EMPDEPTFK
          FOREIGN KEY(Dno) REFERENCES DEPARTMENT(Dnumber)
                             ON DELETE SET DEFAULT         ON UPDATE CASCADE);
CREATE TABLE DEPARTMENT
      (   . . . ,
          Mgr_ssn     CHAR(9)          NOT NULL               DEFAULT '888665555',
          . . . ,
      CONSTRAINT DEPTPK
          PRIMARY KEY(Dnumber),
      CONSTRAINT DEPTSK
          UNIQUE (Dname),
      CONSTRAINT DEPTMGRFK
          FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
                              ON DELETE SET DEFAULT   ON UPDATE CASCADE);
CREATE TABLE DEPT_LOCATIONS
      (   . . . ,
      PRIMARY KEY (Dnumber, Dlocation),
      FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber)
                          ON DELETE CASCADE                ON UPDATE CASCADE);
```

Giving Names to Constraints

- A constraint may be given a constraint name, following the keyword **CONSTRAINT**.
- The names of all constraints within a particular schema must be unique.
- A constraint name is used to identify a particular constraint in case the constraint must be dropped later and replaced with another constraint.
- Giving names to constraints is optional.

```
CREATE TABLE EMPLOYEE
    (    . . . ,
         Dno          INT               NOT NULL          DEFAULT 1,
    CONSTRAINT EMPPK
         PRIMARY KEY (Ssn),
    CONSTRAINT EMPSUPERFK
         FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn)
                           ON DELETE SET NULL           ON UPDATE CASCADE,
    CONSTRAINT EMPDEPTFK
         FOREIGN KEY(Dno) REFERENCES DEPARTMENT(Dnumber)
                          ON DELETE SET DEFAULT         ON UPDATE CASCADE);
CREATE TABLE DEPARTMENT
    (    . . . ,
         Mgr_ssn     CHAR(9)            NOT NULL             DEFAULT '888665555',
         . . . ,
    CONSTRAINT DEPTPK
         PRIMARY KEY(Dnumber),
    CONSTRAINT DEPTSK
         UNIQUE (Dname),
    CONSTRAINT DEPTMGRFK
         FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
                           ON DELETE SET DEFAULT   ON UPDATE CASCADE);
CREATE TABLE DEPT_LOCATIONS
    (    . . . ,
    PRIMARY KEY (Dnumber, Dlocation),
    FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber)
                      ON DELETE CASCADE               ON UPDATE CASCADE);
```