7. SQL

(Structured Query Language)

SQL

- Initially, SEQUEL language (System R) project at IBM

- Three relational languages : QUEL, QBE, SQL

- (Later renamed) Structured Query Language (SQL)

- Currently, Standard Relational database Language
  - Users need to be less concerned to migrate to other types of relational DBMS

- Very-High-level language
  - **Declarative** (= Non-procedural) language
  - Users do not need to specify how (= what order of) to execute query operations.
  - Say “what to do” rather than “how to do”
  - Relational algebra is a procedural language.

SQL

- Table (= relation), row (= tuple), column (= attribute)

- A table consists of :
  - Base Table
  - View (= Virtual table)

- Basic commands;
  - Data Definition Language (DDL)
  - Data Manipulation Language (DML)
  - Data Control Language (DCL)

- DDL is used for defining schemas, tables, and views.
  - CREATE, DROP, ALTER

- DML is used for retrieving and modifying tuples in a table.
  - SELECT (FROM WHERE)
  - INSERT, DELETE, UPDATE

CREATE TABLE

- A table is defined using **CREATE TABLE** command:

  CREATE TABLE R
  ($A_1 : D_1$
  $A_2 : D_2$
  . . . . . .
  $A_n : D_n$)
  Constraints)

  - R : relation name
  - A_i : attribute name
  - D_i : domain (=data type) of A_i
  - Constraints : Integrity Constraints

- A relation defined by CREATE TABLE is called a "**base table**".
  - The relation (and its tuples) are "physically stored".
  - All created tables are initially "empty".
  - Attributes are ordered as they are specified in CREATE TABLE

CREATE TABLE : Attribute Data Types

- System Defined
  - Numeric : **INT**, **SMALLINT**, **FLOAT**(or **REAL**), . . .
  - Character String : **CHAR(n)**, **VARCHAR(n)**, **CLOB**, . .
  - Bit String : **BIT(n)**, **BIT VARYING(n)**, **BLOB**, . .
  - Boolean : **TRUE**, **FALSE**, **UNKOWN**
  - DATE : **YEAR**, **MONTH**, **DAY**, . . .
  - . . . .

- User Defined
  - It is possible for users to specify domain of attribute directly;
  - For example, we can create domain "SSN-TYPE";

    CREATE DOMAIN SSN-TYPE **AS CHAR(9)**
  - We can use this SSN-TYPE for defining attributes SSN, Super_SSN, Mger-SSN, . . .;
  - What if we change later SSN data type to another one?

CREATE TABLE : Constraints

- Key Integrity
  - Primary key and keys can be specified by **PRIMARY KEY** and **UNIQUE**, respectively

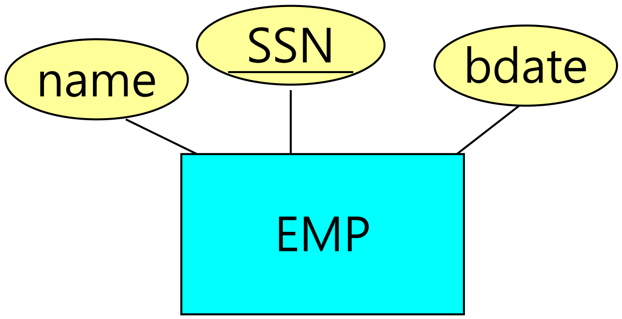
- Entity Integrity
  - This must be specified by **NOT NULL** on **PRIMARY KEY**
  - **NOT NULL** also optionally can be specified on other attributes.

    : For example, employee names, phone may be specified by **non null**; What if new employee's name is unknown?

- Referential Integrity
  - This is specified by **FOREIGN KEY REFERENCES**
  - SQL's default action is to **reject** the operation that violates referential integrity violation
  - SQL also provides **user specified trigger** action

CREATE TABLE : Key/Entity Constraints

ER :



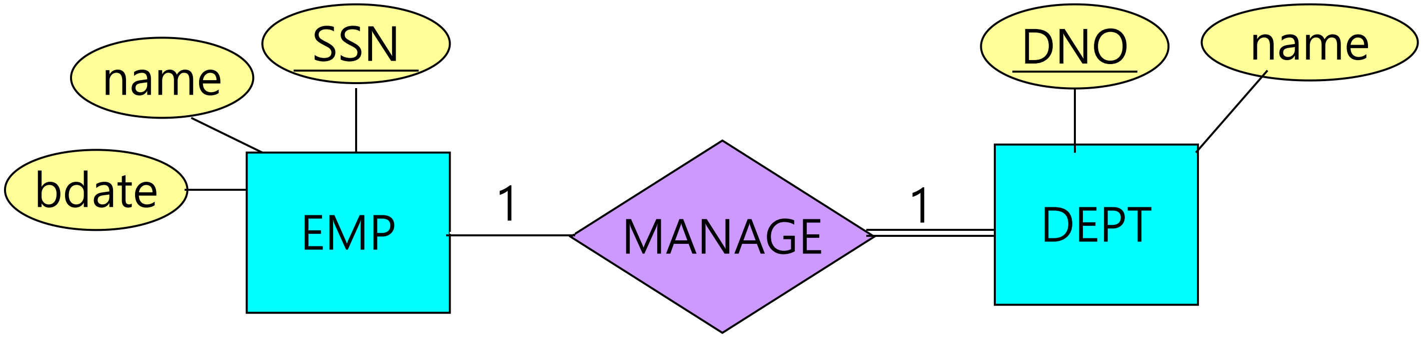
Relation :

EMP (SSN, name, bdate)

Table :

```
 CREATE TABLE   EMP
( SSN     CHAR(9)          NOT NULL
 name    VARCHAR(10) NOT NULL
 bdate   DATE
 PRIMARY KEY (SSN)
 UNIQUE (name)
 )
```

CREATE TABLE : Referential Constraints



EMP (SSN, name, bdate)

DEPT (DNO, name, Mgr_SSN)

```
 CREATE TABLE   EMP
( SSN    CHAR(9)     NOT NULL
 name   CHAR(10)   NOT NULL
 bdate   DATE
 PRIMARY KEY (SSN)
 UNIQUE (name)   )
```

```
CREATE TABLE   DEPT
(DNO          INT              NOT NULL
 name          CHAR(10)
 Mgr_SSN   CHAR(9)    NOT NULL
 PRIMARY KEY(DNO),
 FOREIGN KEY(Mgr-SSN) REFERENCES
EMP(SSN)
```

CREATE TABLE : Default/DOMAIN/CHECK

- **DEFAULT**
  - Default value is included in new tuple if explicit value is not provided
  - For example, default manager for new department is;

    CREATE TABLE DEPT

    Mgr_SSN CHAR(9) **DEFAULT** 999999999
  - If no default value is specified, the default DEFAULT value is NULL.

- **CHECK**
  - We can restrict domain to specific values;
  - For example, department numbers are between 1 and 20;

    DNO **INT** . . .**CHECK** (DNO > 0) **AND** (DNO < 21)
  - CHECK can be also used with **CREATE DOMAIN**

    CREATE DOMAIN DNO **AS INT**
    DNO **INT** . . .**CHECK** (DNO > 0) **AND** (DNO < 21)

CREATE DOMAIN / DEFAULT

```
CREATE DOMAIN SSN-TYPE AS CHAR(9)
CREATE TABLE   EMP
 ( SSN    SSN-TYPE      NOT NULL
   name   CHAR(10)      NOT NULL
   bdate  DATE
   PRIMARY KEY (SSN)
   UNIQUE (name)   )
```

```
CREATE TABLE   DEPT
( DNO          INT            NOT NULL
  name         CHAR(10)   NOT NULL
  Mgr-SSN      SSN-TYPE
  PRIMARY KEY(DNO),
  UNIQUE(name),
  FOREIGN KEY(Mgr-SSN) REFERENCES EMP(SSN)
```

- What if we change later SSN into another type?

CREATE TABLE : Trigger

- SQL provides the following "Referential Trigger Action";

Action (Referencing)	Event (Referenced)

CASCADE
SET NULL
SET DEFAULT
	ON DELETE
	ON UPDATE

- Total 6 trigger actions possible; For example:

(1) **ON UPDATE CASCADE**

: Change the value of referencing FK to the updated (new) PK value for all the referencing tuples;

(2) **ON DELETE CASCADE**

: Delete all the referencing tuples;

(3) **ON UPDATE SET DEFAULT**

(4) **ON DELETE SET DEFAULT**

: The value of FK in referencing tuples is changed to default value;

CREATE TABLE : Trigger

(5) **ON UPDATE SET NULL**

(6) **ON DELETE SET NULL**

: The value of FK in referencing tuples is changed to NULL value;

- Note : This **SET NULL** option is not allowed if the FK in referencing relation is a part of its own primary key.

- It is the responsibility of database designer to specify appropriate trigger actions.

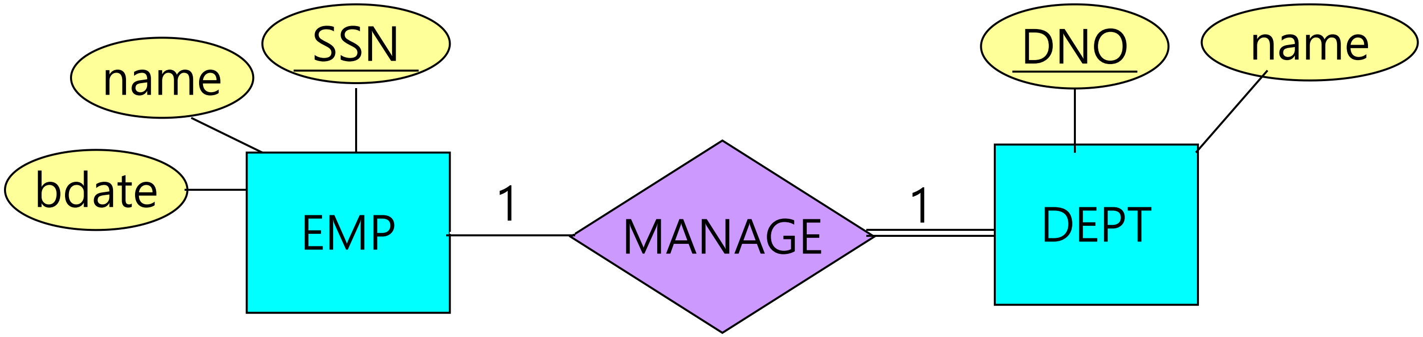
- In general, **CASCADE** is useful for following “relationship” relations.

  - binary relationships (i.e., WORK-ON)

  - weak entity types (i.e., DEPENDENT)

  - multi-valued attributes (i.e., DEPT_LOCATIONS)

CREATE TABLE : Trigger



EMP (SSN, name, DNO)

DEPT (DNO, name, Mgr-SSN)

```
CREATE TABLE   DEPT
( DNO          INT                  NOT NULL
  name         CHAR(10)
  Mgr-SSN      CHAR(9)              NOT NULL
  PRIMARY KEY(DNO),
  FOREIGN KEY(Mgr-SSN) REFERENCES
        EMP(SSN)
          ON DELETE    SET NULL
          ON UPDATE    CASCADE )
```

```
CREATE TABLE   EMP
( SSN  CHAR(9)       NOT NULL
 name CHAR(10)      NOT NULL
 bdate DATE
  PRIMARY KEY (SSN)
  UNIQUE (name)      )
```

CREATE TABLE : Trigger

COURSE

CID	name
CS200	OS
CS250	DB
CS300	PL

Referenced:

PK : CID

ENROLL

CID	SID	credit
CS200	12345	3
CS200	23456	3
CS300	23456	4
CS250	23456	3
CS300	45678	3

Referencing:

FK : CID and SID

PK : (CID, SID)

STUDENT

SID	name	age
12345	Bob	22
23456	An	18
34567	Jim	30
45678	Eve	27

Referenced:

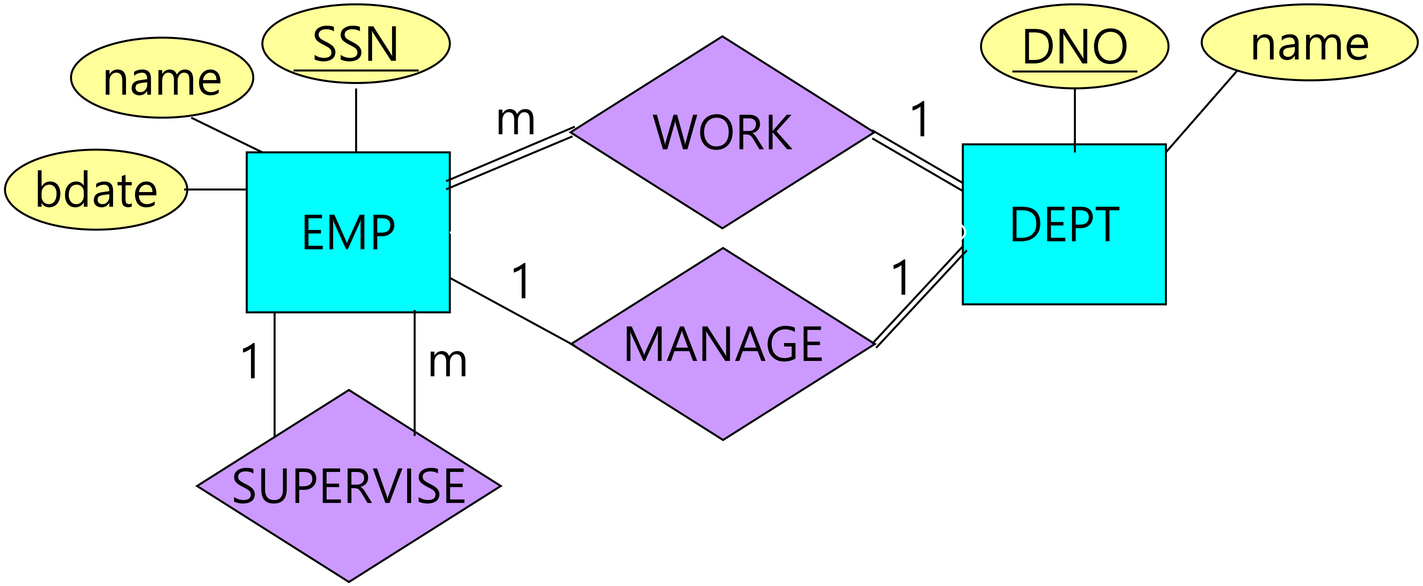
PK : SID

- **CASCADE ON DELETE** STUDENT

- **CASCADE ON UPDATE** COURSE

- **SET NULL ON DELETE** COURSE : This is not allowed; Why??

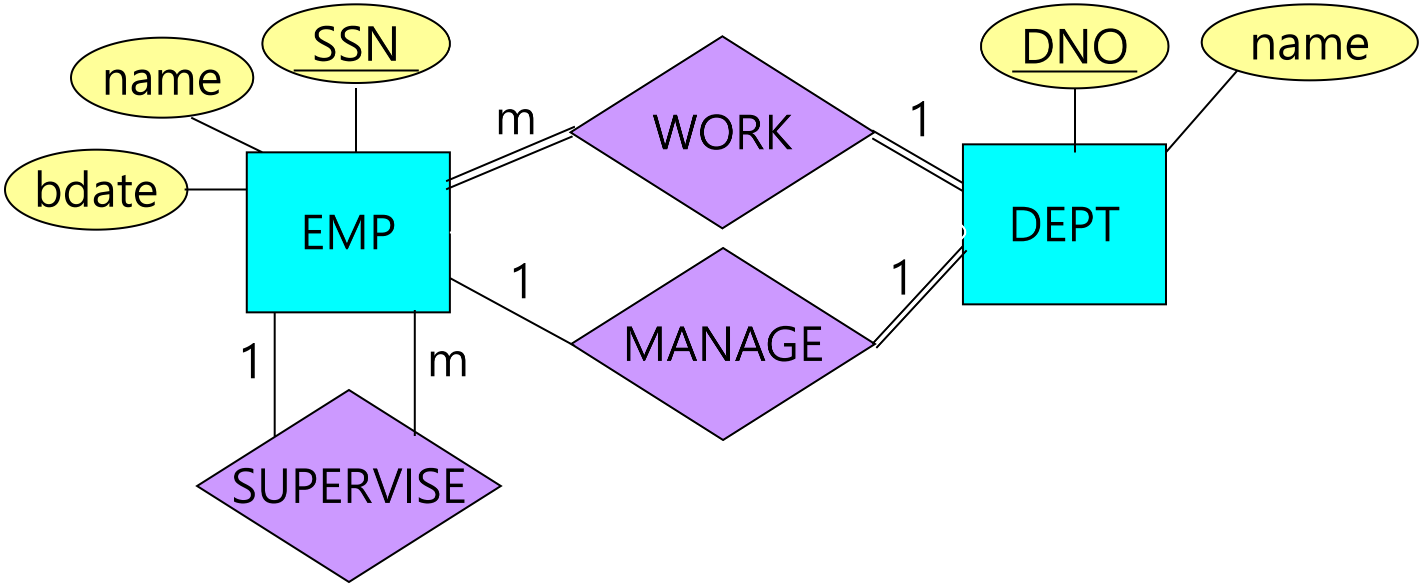
CREATE TABLE : Exercise



- Convert the above ER schema into relational schema;

- Create tables by using SQL; (Specify all constraints/triggers)

CREATE TABLE



EMP (SSN, name, bdate, DNO, Super-SSN)

DEPT (DNO, name, Mgr-SSN)

CREATE TABLE : Exercise

```
● CREATE TABLE   DEPT
  (     DNO             INTEGER             NOT NULL
        name            VARCHAR(10)
        Mgr-SSR         CHAR(9)             NOT NULL
        PRIMARY KEY (DNO),
        UNIQUE (name),
        FOREIGN KEY (MgrSSN)  REFERENCES   EMP(SSN)
             ON DELETE     SET DEFAULT
             ON UPDATE     CASCADE )


  CREATE TABLE   EMP
  (     SSN             CHAR(9)             NOT NULL
        name            CHAR(10)            NOT NULL
        bdate           DATE
        DNO             INTEGER
        Super-SSN       CHAR(9)
        PRIMARY KEY (SSN)
        FOREIGN KEY (DNO) REFERENCES  DEPT(DNO)
             ON DELETE SET DEFAULT
             ON UPDATE CASCADE,
        FOREIGN KEY (Super-SSN) REFERENCES  EMP(SSN)
             ON DELETE SET NULL
             ON UPDATE   CASCADE   )
```

DROP TABLE

- Used to remove a relation (base table) *and its definition*
- The dropped table can no longer be used in queries, updates, or any other commands since it does no longer exist.
- Two DROP options: **CASCADE** and **RESTRICT**
- Example:

DROP TABLE DEPENDENT **RESTRICT**

- This table is removed only if it is not referenced in any constraints or view. Otherwise, the DROP command will not be executed.
- For example, by foreign keys in another tables or views

DROP TABLE DEPENDENT **CASCADE**

- All constraints and views that reference this table are also removed automatically

ALTER TABLE

- Used to add or drop columns and change column definition to an existing table.

- Example:

ALTER TABLE EMP **ADD COLUMN** hobby **CHAR(12)**

- The new column ‘hobby’ will have NULLs in all tuples if no default value is specified.
- Users must enter a value for ‘hobby’ column by using **UPDATE** command.

ALTER TABLE EMP **DROP COLUMN** bdate **CASCADE**

- All constraints and views that ‘bdate’ column are also dropped automatically.
- If **RESTRICT** is used, this command is executed only if no views or constraints reference ‘bdate’ column.