## Lecture 5 Practice - SQL

**Part I. Consider the following relations:**

## Hotel (hotelNo, hotelName, city)

## Room (roomNo, hotelNo, type, price)

## Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)

## Guest (guestNo, guestName, guestAddress)

(The underlined attributes form the primary key.)

Write SQL statements to perform the following:

1. Create the Hotel table, Guest table

CREATE TABLE HOTEL(

hotelNo INT NOT NULL,

hotelName VARCHAR(15) NOT NULL,

city VARCHAR(15) NOT NULL,

PRIMARY KEY(hotelNo)

);

CREATE TABLE GUEST(

guestNo INT NOT NULL,

guestName VARCHAR(15) NOT NULL,

guestAddress VARCHAR(30),

PRIMARY KEY(guestNo)

);

1. Create the Room table with the following conditions:
   1. Room types with the following possibilities: Standard, Deluxe, Suite.
   2. Hotel numbers that are present in the Hotel table.
   3. Room prices between $50.00 and $500.00.
   4. Room numbers between 1 and 9999

CREATE TABLE ROOM(

roomNo INT NOT NULL CHECK (roomNo <= 9999 AND roomNo >= 1),

hotelNo INT NOT NULL,

type VARCHAR(10) NOT NULL CHECK (type IN ('Standard', 'Deluxe', ' Suite')),

price DECIMAL(3, 2) NOT NULL CHECK (price <= 500.00 AND price >= 50.00),

PRIMARY KEY(roomNo,hotelno),

FOREIGN KEY(hotelNo) REFERENCES HOTEL(hotelNo) on delete cascade on update cascade

);

1. Create the Booking table:
   1. Allowing only guests that are present in the Guest table
   2. The hotel room must be in the Room table
   3. dateFrom must precede the dateTo.

CREATE TABLE BOOKING(

hotelNo INT NOT NULL,

guestNo INT,

dateFrom DATE NOT NULL,

dateTo DATE,

roomNo INT NOT NULL,

PRIMARY KEY(hotelNo,dateFrom,roomNo),

FOREIGN KEY(hotelno,roomNo) REFERENCES ROOM(roomNo,hotelno),

FOREIGN KEY(guestNo) REFERENCES GUEST(guestNo),

CHECK (dateFrom < dateTo)

);

**Useful commands:**

CREATE TABLE TableName

({columnName DataType [NOT NULL] [UNIQUE]

[DEFAULT defaultOption] [CHECK (searchCondition) [,…]}

[PRIMARY KEY (listOfColumns),] [UNIQUE (listOfColumns) [,…]]

[FOREIGN KEY (listofForeignKeyColumns)

REFERENCES ParentTableName [(listofCandidateKeyColumns)],

[ON UPDATE referentialAction] [ON DELETE referentialAction] [,…]]

[CHECK (searchCondition) [,…]]);

**Data types:**

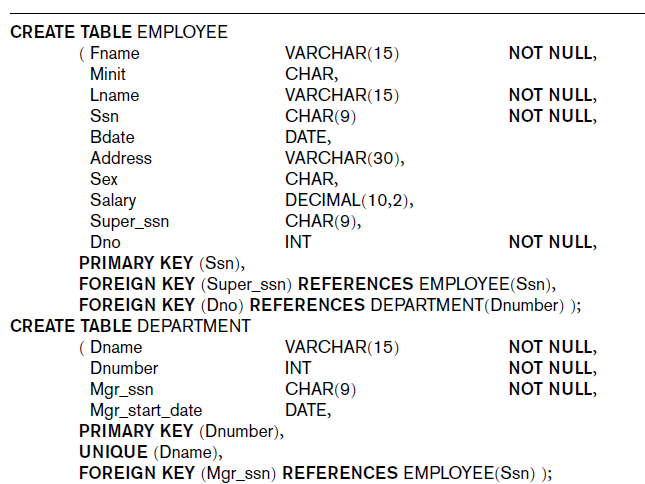
CHAR(length)

VARCHAR(maxLength)

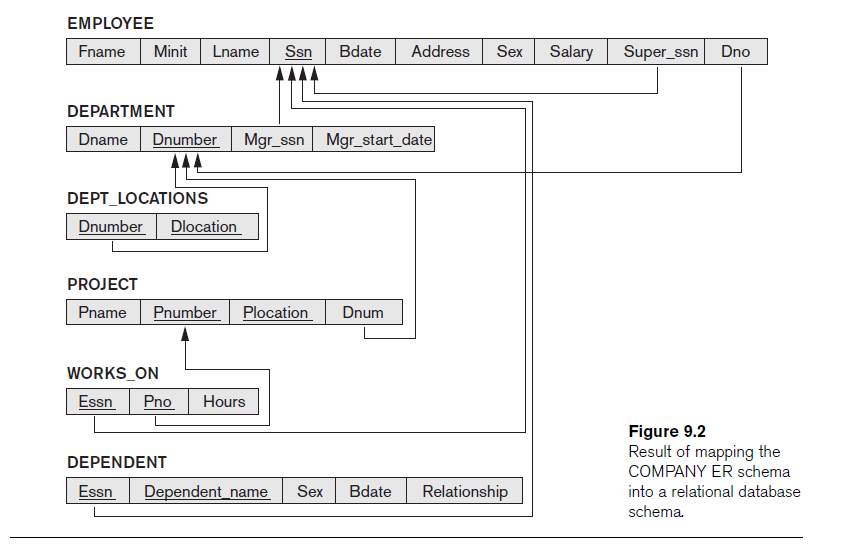
DECIMAL(digits,precision) - the precision is the number of digits after the decimal point

DATE

**Example:**



## Part II Query



**Query syntax:**

SELECT [DISTINCT] {\* | [columnExpression [AS newName]] [,…]}

FROM TableName [alias] [,…]

[WHERE condition]

[GROUP BY columnList

[HAVING condition] ]

[ORDER BY columnList]

Part II. SQL query

1. For every project located in ‘Stafford’, list the project number, the controlling department number, and the department manager’s last name, address, and birth date.

SELECT Pnumber, Dnum, Lname, Address, Bdate

FROM PROJECT, DEPARTMENT, EMPLOYEE

WHERE Dnum=Dnumber AND Mgr\_ssn=Ssn AND Plocation='Stafford';

1. List first name, last name, and address of all employees working in ‘Research’ department

SELECT Fname, Lname, Address

FROM EMPLOYEE, DEPARTMENT

WHERE Dname='Research' AND Dnumber=Dno;

1. List location(s) of ‘Research’ department

SELECT Dlocation

FROM DEPARTMENT, DEPT\_LOCATIONS

WHERE DEPARTMENT.Dnumber=DEPT\_LOCATIONS.Dnumber AND Dname='Research';

1. List every employee name and his/her supervisor name

SELECT E.Fname, E.Lname, S.Fname, S.Lname

FROM EMPLOYEE AS E, EMPLOYEE AS S

WHERE E.Super\_ssn=S.Ssn;

1. List all distinct employee salary

SELECT DISTINCT Salary

FROM EMPLOYEE;

1. Retrieval all employees’ names whose address is in Houston

SELECT Fname, Lname

FROM EMPLOYEE

WHERE Address LIKE '%Houston%';

1. Retrieval all employees’ names who were born during the 1970s

SELECT Fname, Lname

FROM EMPLOYEE

WHERE Bdate LIKE '\_\_7\_\_\_\_\_\_\_';