BASIC SQL

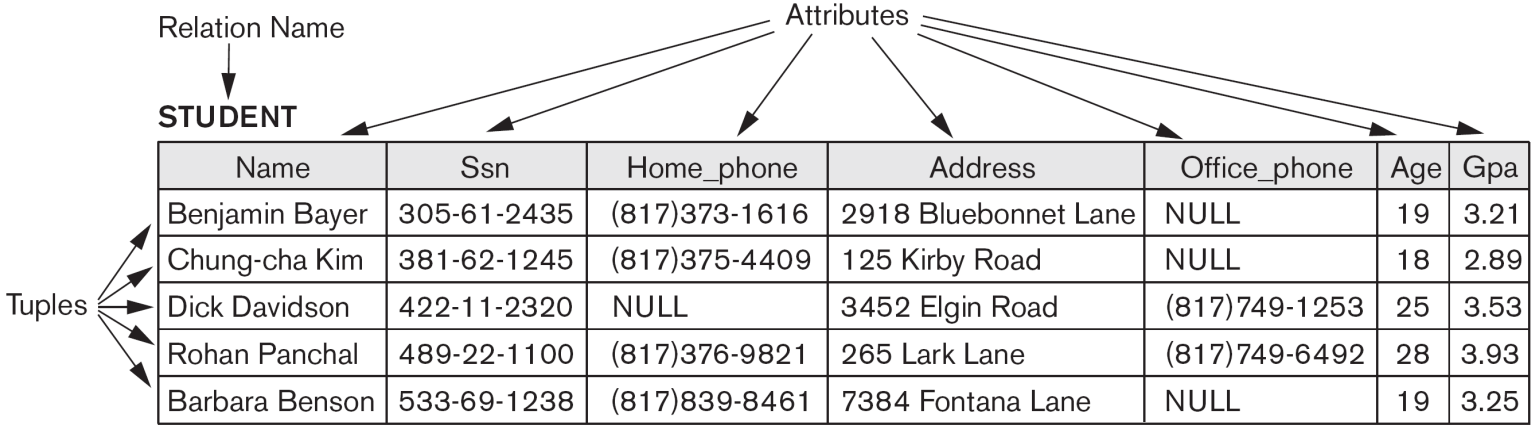
Corresponding Reading: Chapter 6.1, 6.3

SQL

- Structured Query Language – SEQUEL/SQL
- Considered one of the major reasons for success of relational databases.
- Multiple versions lead by ANSI/ISO
  - SQL-86, SQL-92, …, SQL-2006, SQL-2008
- Comprehensive database language
  - Statements for data definitions, queries and updates
  - Defining views of the DB
  - Security and authorization
  - Transaction control, etc…

SQL Terminology

- SQL uses the following terms, which are similar to the relation model:
  - Table / Relation
  - Row / Tuple
  - Column / Attribute



Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Creating Tables

- The CREATE TABLE command is used to specify a new relation by giving it a name and specifying its attributes and initial constraints.
  - Attributes are specified first:
    - Each attribute is given a name, a data type to specify its domain of values, and any attribute constraints
  - Key values, entity integrity and referential integrity constraints can also be specified within the CREATE TABLE statement.

Specifying Attribute Constraints/Defaults

- SQL allows NULL values for attributes
- A constraint NOT NULL may be specified if NULL is not permitted for a particular attribute.
  - Implicit for primary key attributes
- Possible to specify a default value for an attribute by appending a DEFAULT <value> clause to an attribute definition.
  - Default value is included in any new tuple if an explicit value is not provided for that attribute.

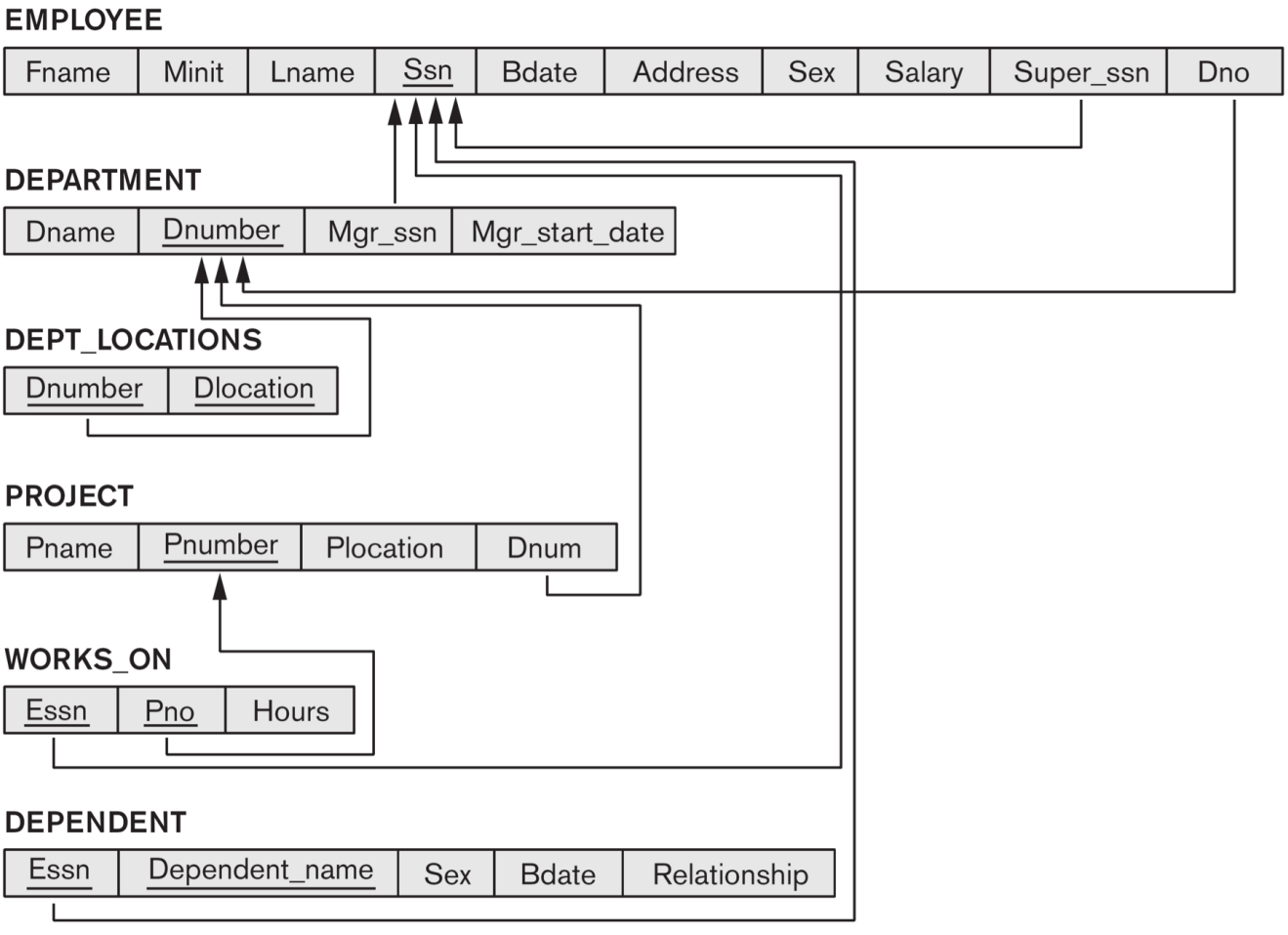
Specifying Attribute Constraints/Defaults

- Restrict attribute values using the CHECK clause following an attribute definition.
- Example: Department numbers are restricted to integer numbers between 1 and 20
  - Specify constraint in SQL:
    - CHECK (Dnumber > 0 AND Dnumber < 21);

Specifying Keys and Unique Values

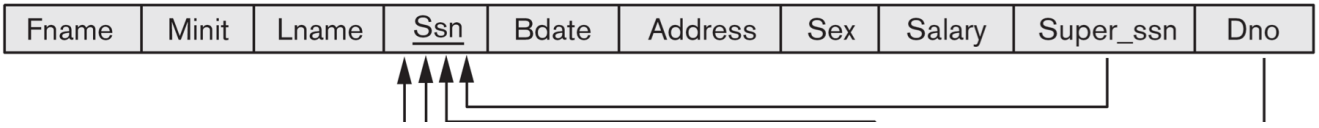
- Special clauses for keys and referential integrity constraints within the CREATE TABLE statement.
- PRIMARY KEY clause specifies one or more attributes that make up the primary key of a relation.
  - Example: Dnumber INT PRIMARY KEY;
- The UNIQUE clause specifies alternative (secondary) keys
  - Example: Dname VARCHAR(15) UNIQUE;

Example: CREATE TABLE for COMPANY DB



Example: CREATE TABLE for COMPANY DB

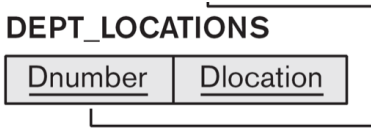
EMPLOYEE




```
CREATE TABLE EMPLOYEE
        ( Fname          VARCHAR(15)          NOT NULL,
          Minit          CHAR,
          Lname          VARCHAR(15)          NOT NULL,
          Ssn            CHAR(9)              NOT NULL,
          Bdate          DATE,
          Address        VARCHAR(30),
          Sex            CHAR,
          Salary         DECIMAL(10,2),
          Super_ssn      CHAR(9),
          Dno            INT                  NOT NULL,
        PRIMARY KEY (Ssn),
        FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn),
        FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber) );
```

Example: CREATE TABLE for COMPANY DB

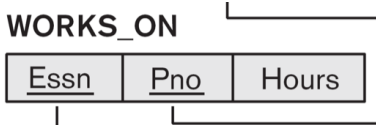

```
CREATE TABLE DEPARTMENT
        ( Dname                 VARCHAR(15)             NOT NULL,
          Dnumber               INT                     NOT NULL,
          Mgr_ssn               CHAR(9)                 NOT NULL,
          Mgr_start_date        DATE,
        PRIMARY KEY (Dnumber),
        UNIQUE (Dname),
        FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn) );
```




```
CREATE TABLE DEPT_LOCATIONS
        ( Dnumber               INT                     NOT NULL,
          Dlocation             VARCHAR(15)             NOT NULL,
        PRIMARY KEY (Dnumber, Dlocation),
        FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber) );
```

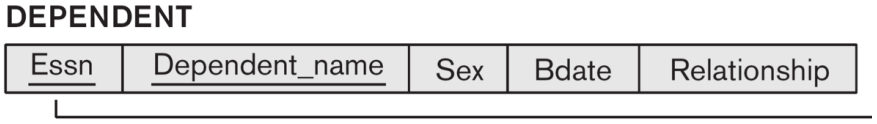
Example: CREATE TABLE for COMPANY DB


```
CREATE TABLE PROJECT
        ( Pname                 VARCHAR(15)             NOT NULL,
          Pnumber               INT                     NOT NULL,
          Plocation             VARCHAR(15),
          Dnum                  INT                     NOT NULL,
        PRIMARY KEY (Pnumber),
        UNIQUE (Pname),
        FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber) );
```




```
CREATE TABLE WORKS_ON
        ( Essn                  CHAR(9)                 NOT NULL,
          Pno                   INT                     NOT NULL,
          Hours                 DECIMAL(3,1)            NOT NULL,
        PRIMARY KEY (Essn, Pno),
        FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
        FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber) );
```

Example: CREATE TABLE for COMPANY DB




```
CREATE TABLE DEPENDENT
        ( Essn                  CHAR(9)             NOT NULL,
          Dependent_name        VARCHAR(15)         NOT NULL,
          Sex                   CHAR,
          Bdate                 DATE,
          Relationship          VARCHAR(8),
        PRIMARY KEY (Essn, Dependent_name),
        FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn) );
```

Basic Retrieval Queries

- SQL has one basic statement for retrieving information from a database:
  - The SELECT statement
- There are many options for the SELECT statement and we will see them gradually.
- Use the COMPANY DB to explore the function of the SELECT statement and its clauses.

SELECT-FROM-WHERE Queries

- Basic SQL Query with three clauses:

  SELECT <attribute list>

  FROM <table list>

  WHERE <condition>;

  - <attribute list> is a list of attributes whose values are to be retrieved by the query
  - <table list> is a list of the relation names required to process the query
  - <condition> is a conditional (Boolean) expression that identifies the tuples to be retrieved by the query.

- Basic logical comparison operators for comparing attribute values with one another: =, <, <=, >, >= and <>

Example: SELECT-FROM-WHERE

- Retrieve the birth date and address of employee(s) whose name is 'John B. Smith'

```
SELECT    Bdate, Address
FROM      EMPLOYEE
WHERE     Fname='John' AND Minit='B' AND
          Lname='Smith';
```

  - Query *selects* the individual EMPLOYEE tuples that satisfy the condition of the WHERE clause
  - Then *projects* the results on the Bdate and Address attributes listed in the SELECT clause

Example Query: Output

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Output:

- SELECT Bdate, Address

Bdate	Address
1965-01-09	731Fondren, Houston, TX