Lecture 7
01/27/16

Instructor: Yu-San Lin
yusan@psu.edu

Course Website: http://www.cse.psu.edu/~yul189/cmpsc431w

Slides based on McGraw-Hill & Dr. Wang-Chien Lee

Basic Commands

- `SHOW DATABASES;`
- `CREATE DATABASE [database name];`
- `USE [database name];`
- `SHOW TABLES;`
- `DESCRIBE [table name];`
- `SELECT * FROM [table name];`

Integrity Constraints

- Integrity constraints (ICs): conditions that must be true for any instance of the database
- Examples of ICs:
  - Domain Constraints
  - No two students have the same *sid*
- ICs are specified and enforced when:
  - Schema is defined
  - Relations are modified
- DBMS should not allow illegal instances

Key Constraints

- A set of fields is a ___Candidate Key___ for a relation if:
  - No two distinct tuples can have same values in all key fields
  - No subset of the set of fields in a key is a unique identifier for a tuple
- Can *{sid, name}* be candidate key?

sid	name	login	age	gpa
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8

Key Constraints (cont.)

- We call {sid, name} ___Superkey_____
- Superkey
  - A set of fields that contains a key
  - A tuple is always a superkey
- Only one candidate key is chosen to be the primary key
- DBMS may create an index with the primary key fields as the search key

Key Constraints in SQL

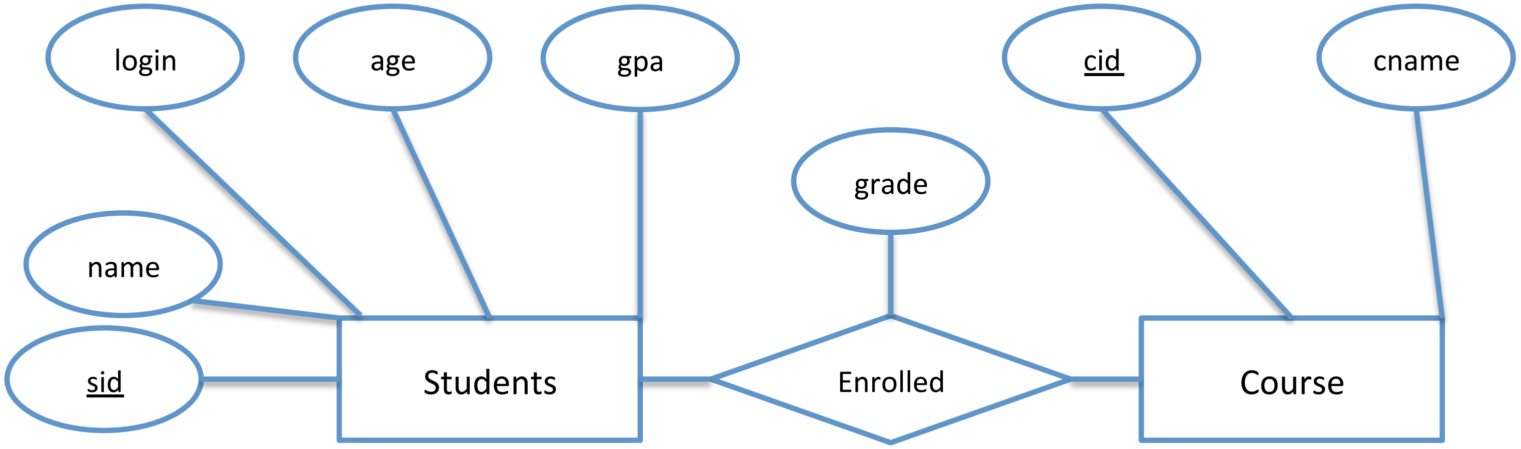
```
CREATE TABLE Students
(  sid   CHAR(20),
   name  CHAR(30),
   login CHAR(20),
   age   INTEGER,
   gpa   REAL,
   UNIQUE (name, age),
   PRIMARY KEY (sid))
```

Candidate Key

Primary Key

SQL does not require PRIMARY KEY to be declared.

Foreign Key Constraints



Student (<u>sid</u>: string, name: string, login: string,
age: integer, gpa: real)
Enrolled (studid: string, cid: string, grade: string)

- *studid* in Enrolled is the Foreign key that refers to Students.

Foreign Key (cont.)

- Only students listed in the Students relation should be allowed to enroll for course

sid	name	login	age	gpa
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

studid	cid	grade
53831	Carnatic101	C
53832	Reggae203	B
53650	Topology112	A
53666	History105	B

Foreign Keys in SQL

```
CREATE TABLE Enrolled
(  studid  CHAR(20),
   cid     CHAR(20),
   grade   CHAR(10),
   PRIMARY KEY (studid, cid),
   FOREIGN KEY (studid) REFERENCES Students(sid));
```

Enforcing Integrity Constraints

- If an insert, delete, or update command causes a violation, it is rejected.

```
INSERT INTO Students (sid,name,login,age,gpa)
VALUES (53688, 'Mike', 'mike@ee', 17,3.4);

INSERT INTO Students
VALUES (null, 'Mike', 'mike@ee', 17, 3.4);
```

Referential Integrity

- What should be done if an Enrolled tuple with a non-existent student id is inserted?

Referential Integrity (cont.)

- What should be done if a Students tuple is deleted?
  - Delete all Enrolled rows that refer to the deleted Students row
  - Disallow the deletion of the Students row if an Enrolled row refers to it
  - Set the studid column to the sid of some default students
  - Set studid column to null ←→ sid is primary key in Students

Referential Integrity (cont.)

- What should we do if the primary key value of a Students row is updated?

Referential Integrity (cont.)

```
CREATE TABLE Enrolled
(  studid   CHAR(20),
   cid      CHAR(20),
   grade    CHAR(10),
   PRIMARY KEY (studid,cid),
   FOREIGN KEY (studid) REFERENCES Students (sid)
                ON DELETE CASCADE
                ON UPDATE NO ACTION);
```

Rows refer to it are to be deleted as well

Means reject

Foreign key declaration

Referential Integrity (cont.)

- Options on DELETE and UPDATE:
  - `NO ACTION`: reject
  - `CASCADE`: delete/update all tuples that refer to the deleted/updated tuple
  - `SET NULL`
  - `SET DEFAULT`

Referential Integrity: Example 1

studid	cid	grade
53831	Carnatic101	C
53832	Reggae203	B
53650	Topology112	A
53666	History105	B

sid	name	login	age	gpa
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

What happen when update Enrolled as follow:

- Delete the tuple with studid = 53650
- Insert a tuple with studid 53600
- Update the tuple with studid = 53831 → 53666
- Update the tuple with studid = 53666 → 53600

Referential Integrity: Example 2

studid	cid	grade
53831	Carnatic101	C
53832	Reggae203	B
53650	Topology112	A
53666	History105	B

sid	name	login	age	gpa
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

What happen when update Students as follow:

- Insert a tuple with sid = 53600
- Delete the tuple with sid = 53666
- Update the tuple with sid = 53650 → 53600

Querying Relational Data

- A relational database query is a question about the data, and the answer consists of a new relation containing the result.
- Select tuple with condition:

```
SELECT *
FROM Students S
WHERE S.age < 18;
```

Querying Relational Data (cont.)

- A query can extract a subset of fields:

```
SELECT S.name, S.login
FROM   Students S
WHERE  S.age < 18;
```

Querying Relational Data (cont.)

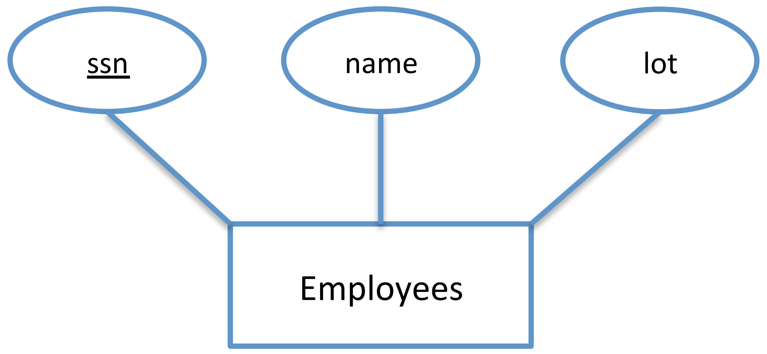
- A query can also combine information in multiple relations:

```
SELECT   S.name, E.cid
FROM     Students S, Enrolled E
WHERE    S.sid = E.studid AND E.grade = ‘A’;
```

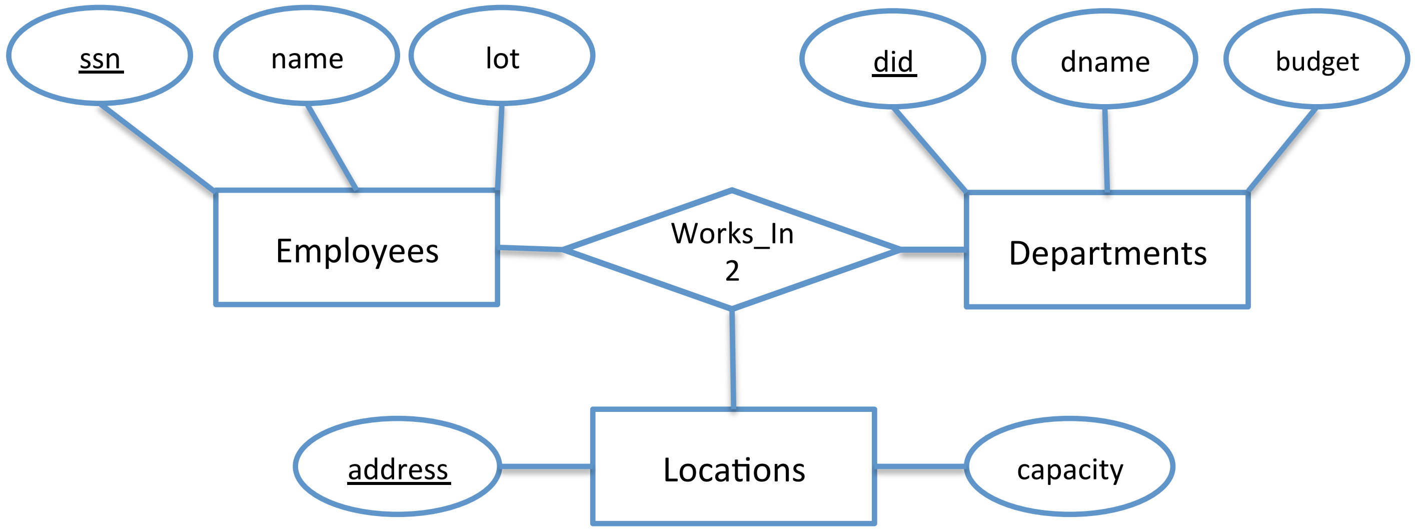
ER to Relational

- Entity sets to tables

```
CREATE TABLE Employees
(ssn     CHAR(11),
 name    CHAR(30),
 lot     INTEGER,
 PRIMARY KEY  (ssn));
```



Relationship Sets to Tables

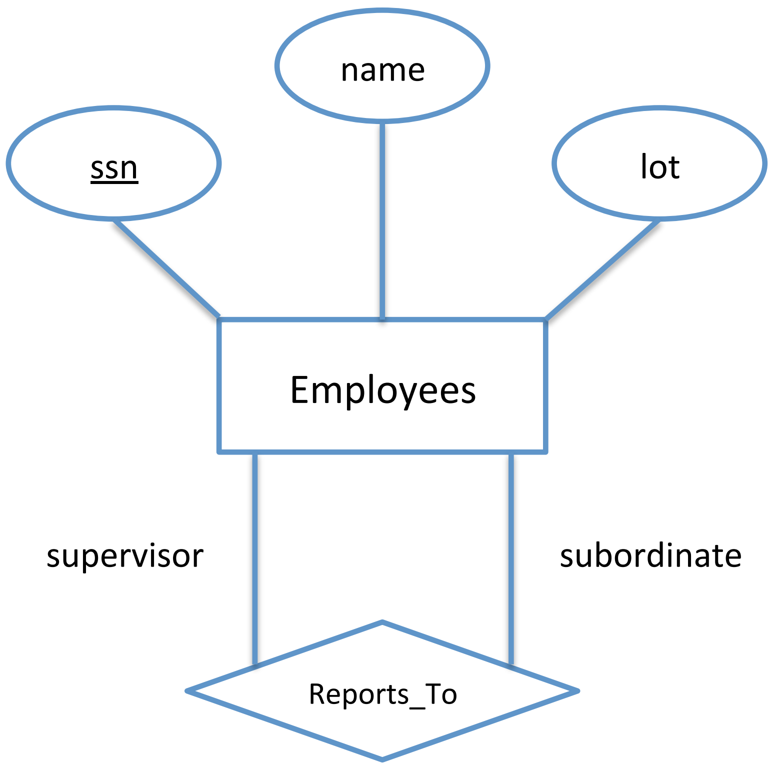


- When translating a relationship set to a relation, attributes of the relation must include:
  - Keys for each participating entity set (as Foreign Keys)
    - This set of attributes forms a Primary Keys for the relation
  - All descriptive attributes

Relationship Sets to Tables (cont.)

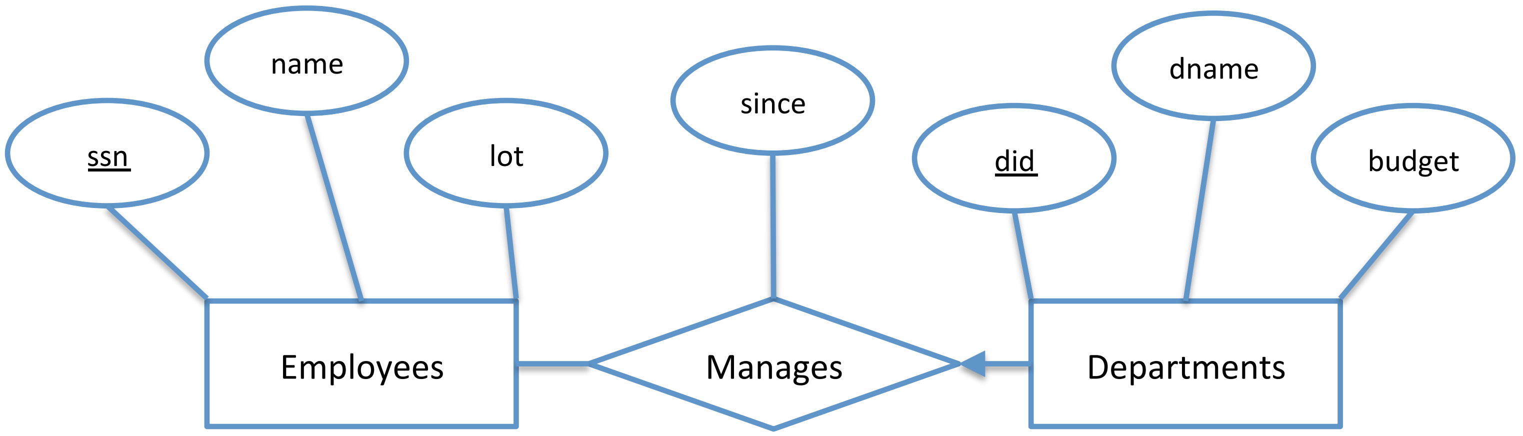
```
CREATE TABLE Works_In2
(ssn            CHAR(11),
 did            INTEGER,
 address        CHAR(20),
 since          DATE,
 PRIMARY KEY (ssn, did, address          ),
 FOREIGN KEY (ssn)     REFERENCES Employees(ssn),
 FOREIGN KEY (address) REFERENCES Locations(address),
 FOREIGN KEY (did)     REFERENCES Departments(did));
```

Relationship with Same Entity Set



```
CREATE TABLE Reports_To
(supervisor_ssn  CHAR(11),
 subordinate_ssn CHAR(11),
 PRIMARY KEY (supervisor_ssn, subordinate_ssn),
 FOREIGN KEY (supervisor_ssn)  REFERENCES employees (ssn),
 FOREIGN KEY (subordinate_ssn) REFERENCES employees (ssn));
```

Relationship Sets with Key Constraints



- For each Department, there is ___________ Employee managing it
  - Is it ok for a Department to not be managed by any employee?
  - Is it ok for an Employee to manage more than one Departments?

→ For Manages relationship, Employees to Department is a ____________ relationship

Relationship Sets with Key Constraints (cont.)

- Option 1:

```
CREATE TABLE Manages
(ssn    CHAR(11),
 did    CHAR(11),
 since  DATE,
 PRIMARY KEY (______________),
 FOREIGN KEY (ssn) REFERENCES Employees(ssn),
 FOREIGN KEY (did) REFERENCES Departments(did));
```

Relationship Sets with Key Constraints (cont.)

- Option 2:

```
CREATE TABLE Dept_Mgr
(did     INTEGER,
 dname   CHAR(20),
 budget  REAL,
 ssn     CHAR(11),
 since   DATE,
 PRIMARY KEY (did),
 FOREIGN KEY (___) REFERENCES ________________);
```

Don’t Forget

- Project phase 1 due next Friday (9/16)
  - Project report (>= 10 pages)
  - Project presentation in class (4 minutes)
- Reading: 3.5 – 3.7