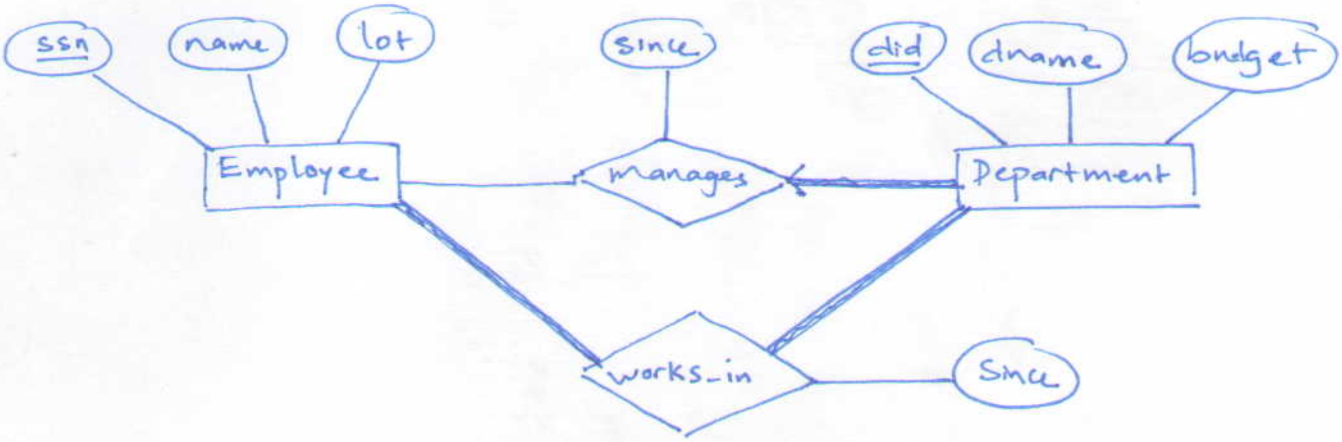
CS541 SPRING 2012

ASSIGNMENT 1

DUONG NGUYEN

Part 1 Standard exercises

1. ER-diagram



convert Employee entity set to relation

```
CREATE TABLE Employee ( ssn   CHAR (10),
                        name  CHAR (20),
                        lot   INTEGER,
                        PRIMARY KEY (ssn))
```

Convert Department entity set and Manage relationship to relation

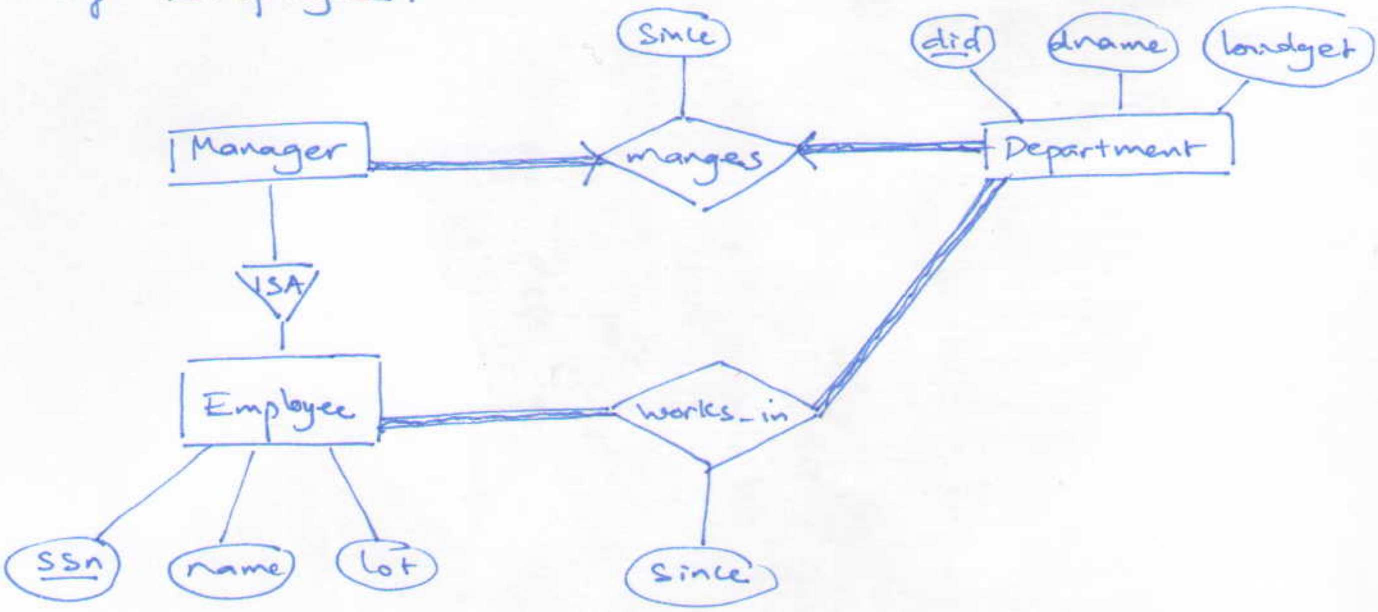
```
CREATE TABLE Dept_Mgr ( did    INTEGER,
                        dname  CHAR (20),
                        budget INTEGER,
                        ssn    CHAR (10) NOT NULL,
                        since  INTEGER,
                        PRIMARY KEY (did),
                        FOREIGN KEY (ssn) REFERENCES
                        Employee )
```

Convert works-in relationship to relation

```
CREATE TABLE works-in ( did   INTEGER,
                        ssn   CHAR (10),
                        since INTEGER,
                        PRIMARY KEY (did, ssn))
```

The participation constraints of works_in cannot be captured in the definition of works_in relation schema. It requires using ASSERTION.

2. We may introduce Manager entity set as a subclass of Employee.



Every manager entity must participate in manages relationship.
⇒ total participation.
We can assume that each manages at most 1 department
⇒ key constraint.
With the above E-R diagram, we will have the following changes

define Manager relation:

```
CREATE TABLE Manager ( ssn CHAR(10),
                       PRIMARY KEY (ssn),
                       FOREIGN KEY (ssn)
                       REFERENCES Employee )
```

change in Dept_Mgr

```
CREATE TABLE Dept_Mgr ( did     INTEGER,
                        dname   CHAR (20),
                        budget  INTEGER,
                        ssn     CHAR( 10) NOT NULL,
                        since   INTEGER,
                        PRIMARY KEY (did),
                        UNIQUE (ssn),
                        FOREIGN KEY (ssn) REFERENCES
                        Employee )
```

Total participation of Manager need to be expressed by ASSERTION.

There is no change to Employee relation and works_in relation.

3. within the Employee ~~entity set~~ relation

ssn → (name, lot)

~~within the Department entity set~~

~~did →~~

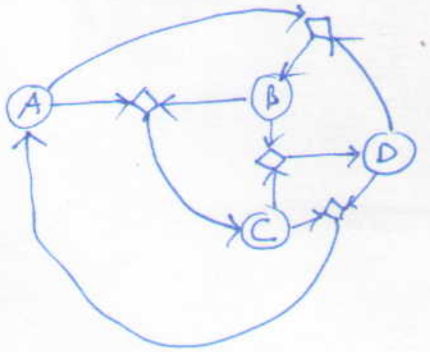
within the Dept_Mgr relation

did → (dname, budget, Manages.since, ssn)

within works_in relation

(did, ssn) → works_in.since

4.



4.1. List all notrivial FDś

$AB \to C$
$BC \to D$.
$CD \to A$
$AD \to B$.

We have $AB \to C$ and $AB \to B$ then $AB \to BC$
But $BC \to D$. Thus $AB \to C$, $AB \to D$. Due to reflexity, $AB \to A$, $AB \to B$. Therefore $AB \to ABCD$ (AB is a superkey)
However, with the given FDs, we cannot imply a new FD that the left hand side is A or B. In other words, A and B are not the key of the schema ⇒ AB is a key.
Due to its symmetry, we can easily show that BC, CD, AD also are keys of the schema ABCD.
The list of notrivial FDs will be.

$AB \to C$ $AB \to D$
$AB \to AC$ $AB \to AD$ $AB \to BC$ $AB \to BD$ $AB \to CD$
$AB \to ABC$ $AB \to ABD$ $AB \to BCD$
$AB \to ABCD$.

The list of notrivial FDs whose left hand side is BC, CD, AD can be obtained in the same way.

$ABC \to AD$ $ABC \to BD$ $ABC \to CD$
$ABC \to ABD$ $ABC \to ACD$ $ABC \to BCD$
$ABC \to ABCD$

For notrivial FDś whose left hand side is ABD, ACD, BCD can be obtained in the same way.

4.2. All the keys of R are:
AB, BC, CD, DA.

4.3. All Superkeys of R that are not keys:
ABC, ABD, ACD, BCD, and ABCD.

5. Let denote
$A_1 A_2 \dots A_m$ as A
$B_1 B_2 \dots B_n$ as B
$C_1 C_2 \dots C_j$ as C
$D_1 D_2 \dots D_k$ as D

we have $A \to B$ and $C \to D$.

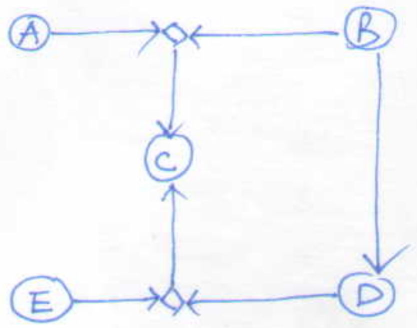
$A \to B$ then $AC \to BC$ (augmentation).

$C \to D$ then $BC \to BD$ (of augmentation)

$\Rightarrow AC \to BC$ and $BC \to BD$

$\Rightarrow AC \to BD$ (transitivity). Q.E.D.

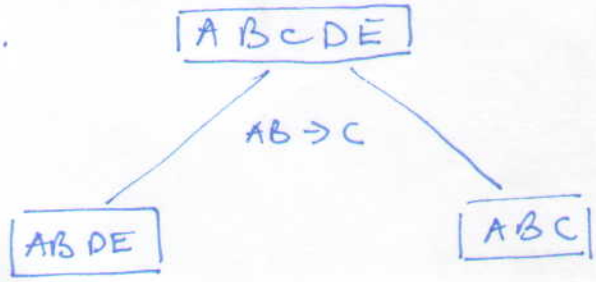
6.



$ABE \to AB \to C$
$ABE \to B \to D$
$\Rightarrow ABE \to ABCDE$
ABE is the (only) key.

6.1. All the FD's are notrivial
all the FD's have the left hand side not a Superkey
$\Rightarrow$ all the FD's violate BCNF.

6.2.

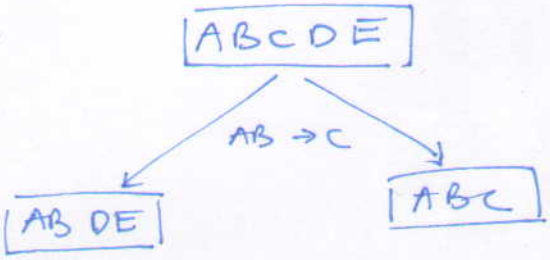


This decomposition is lossless join because the common attributes between ABDE and ABC is AB which is the key of ABC (because $AB \to C$).
This decomposition is not dependency preserving.

6.3. Because ABE is the key then

AB → C , DE → C, and B → D all violates 3NF.

6.4.



This decomposition is the same as the decomposition in 6.2, and it is lossless join but not dependency preserving.

Part 2 Design exercise.

In producing a relational schema supporting a database of course and student registration information, we will undergo through following steps:

- Conceptual schema ~~deg~~ design of the database to build ER diagram of the database
- Relational schema design to identify relations in the database as well as keys and other constraints on each relation.
- Normalization relational schema obtained from step 2.

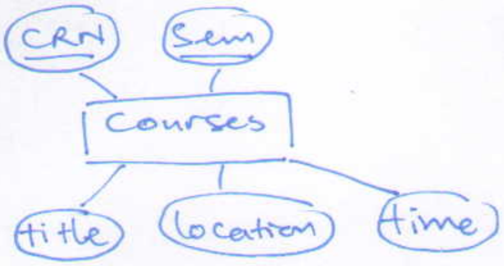
Step 1: Conceptual Scheme design

1.1. we need to identify which ~~entity~~ entities will be included in the database and relationship between entities

1.2. In real-world scenario, we have to meet will customers to obtain information for this step

1.3. In our example, we need the following entity sets: Students, Course, Instructor, Textbook

Attributes for each entity set will be



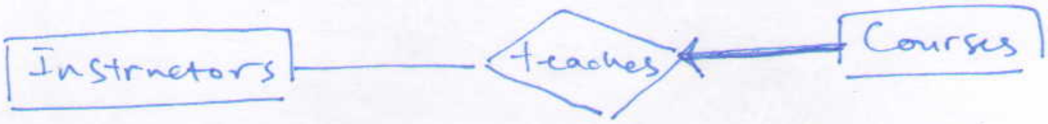
CRN = course number

Sem = semester

Instructors teach courses.

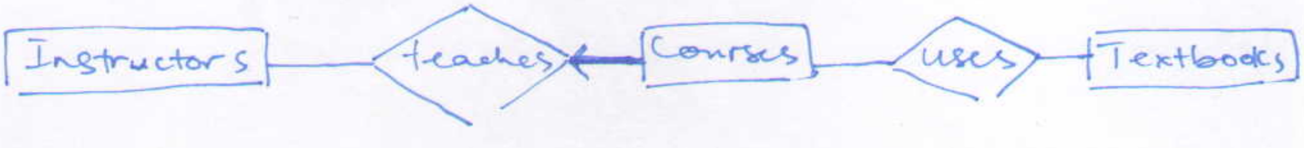
A course must be taught by at least 1 instructor (total participation)

Furthermore, we assume a course is taught by at most 1 instructor (key constraint)

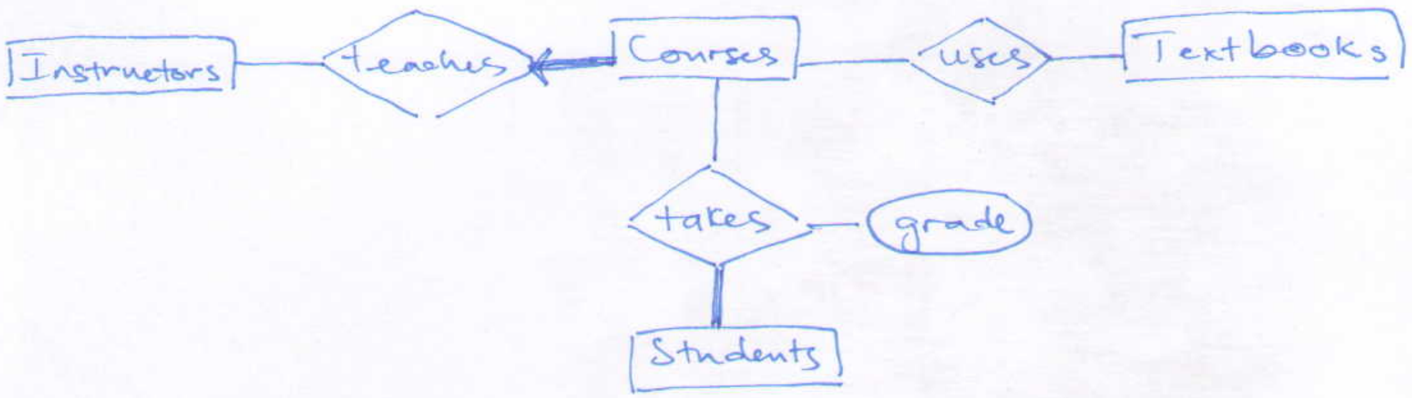


Attributes of entity sets are not drawn to make the E-R diagram simple.

A course will use textbooks.



Students take courses. Every student takes at least 1 course (total participation)



Step2: Relational schema design

2.1.} we obtain the information from the E-R
2.2.} diagram above.

2.3. In our database, we have the following relations (and their associated constraints)

Instructors (<u>iid</u>: string , iname : string)

Students (<u>sid</u> : string, sname : string)

Textbooks (<u>tbid</u> : string, title ; string, author : string, year : integer)

Courses (<u>CRN</u>: string, <u>Sem</u> : string, title : string, location : string, time : string)

Teaches (<u>CRN</u> : string, <u>Sem</u> : string, iid : string)

Foreign key : (CRN, Sem) referencing to Courses.
iid referencing to Instructors

Uses (CRN : String , sem : String , tbid : string)

Foreign key :(CRN, Sem) referencing to Courses
tbid referencing to Textbooks

Takes (<u>sid</u>: string, <u>CRN</u>: string, <u>sem</u>: string, grade : real)
Foreign keys: sid referencing to Students
(CRN, sem) referencing to Courses

Total participation constraint: Students.

<u>Step 3</u>: Normalization

3.1. }
3.2. } The information needed for normalization is the relational schema from step 2 and the functional dependencies.

3.3. We observe that in the relations described in step 2, all primary key constraints are FDs that satisfy BCNF and 3NF.
Besides primary key constraints, we may have the following functional dependencies which are also candidate key constraints:

In Textbooks: (title, year) can be a candidate key

In Courses: (title, sem) can be a candidate key.

Therefore, the above relational schema design is already in BCNF and 3NF. No further normalization is needed.