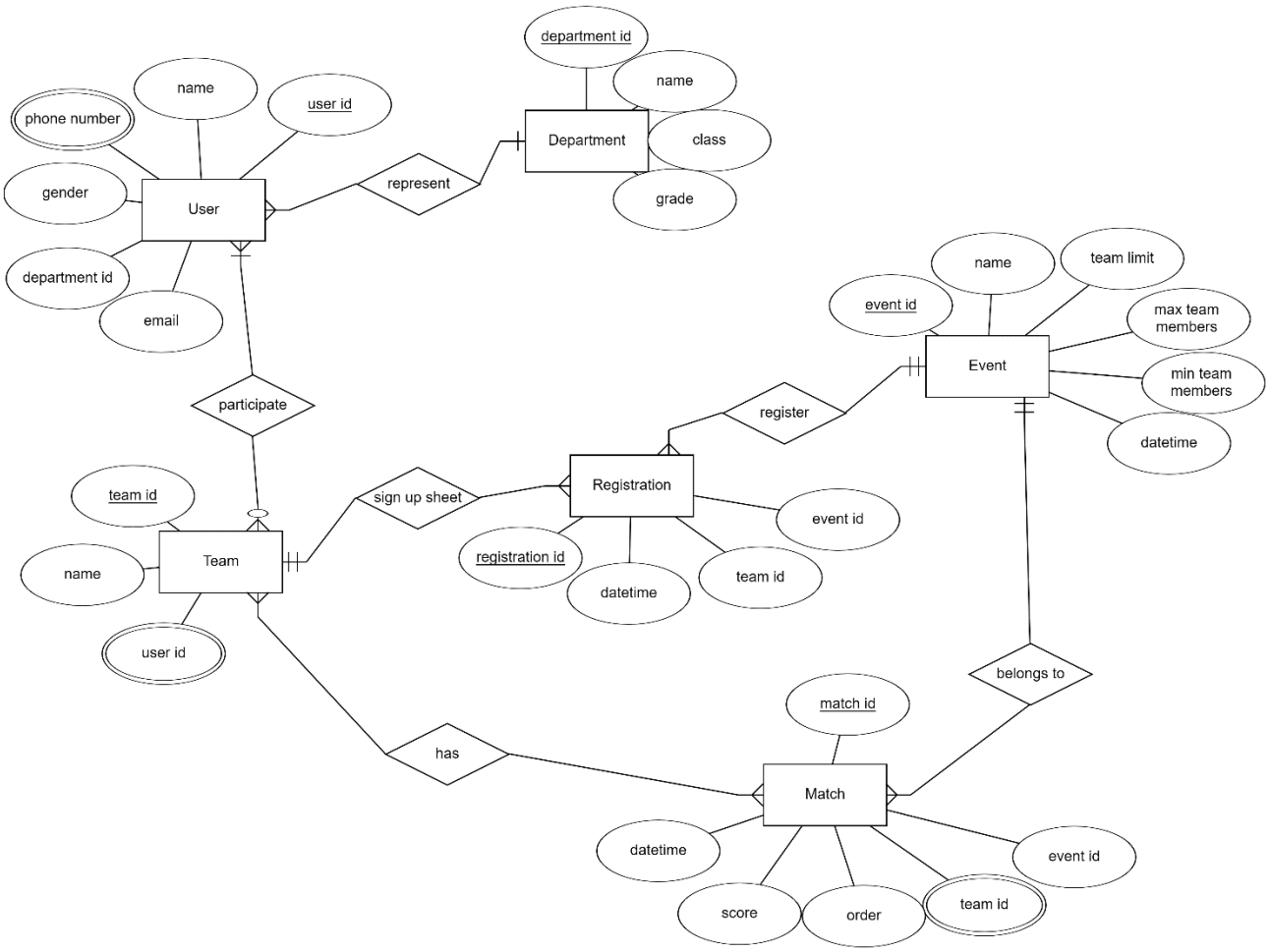
Introduction to Database Systems

Individual Homework 3: ER Model & Normalization

0716234 蕭彧

Part1



Relation A can be achieved by many(unspecified) to one(unspecified) relationship between entity “User” and Entity “Department”.

Relation B can be achieved by many(mandatory) to many(optional) relationship between entity “User” and Entity “Team”.

Relation C can be achieved by many(mandatory) to many(optional) relationship between entity “User” and Entity “Team”.

Relation D can be achieved by one(mandatory) to many(unspecified) relationship between entity “Team” and Entity “Registration”, and many(unspecified) to one(mandatory) relationship between entity “Registration” and Entity “Event”.

Relation E can be achieved by many(unspecified) to one(mandatory) relationship between entity “Match” and Entity “Event”.
Relation F can be achieved by many(unspecified) to many(unspecified) relationship between entity “Team” and Entity “Match”.

Part2

A.

(1)

Functional dependency is a constraint between two sets of attributes in a relation from a database. For instance, let R be a relation schema and let x and y be nonempty sets of attributes in R. An instance(relation) r of R satisfies the FD x->y if the following holds for every pair of tuples t1 and t2 in r. If t1.x = t2.x then t1.y = t2.y.

For example, if we know the value of Employee number, we can obtain Employee Name, city, salary, etc. By this, we can say that the city, Employee Name, and salary are functionally depended on Employee number.

Employee number	Employee Name	Salary	City
1	Dana	50000	San Francisco
2	Francis	38000	London
3	Andrew	25000	Tokyo

(2)

A functional dependency is a property of the relation schema R, not of a particular legal relation state r of R. Therefore, an FD cannot be inferred automatically from a given relation extension r but must be defined explicitly by someone who knows the semantics of the attributes of R.

For example, the following figure shows a particular state of the TEACH relation schema. Although at first glance we may think that Text→Course, we cannot confirm this unless we know that it is true for all possible legal states of TEACH. It is, however, sufficient to demonstrate a single counter example to disprove a functional dependency. For example, because ‘Smith’ teaches both ‘DataStructures’ and ‘Data Management,’ we can conclude that Teacher does not functionally determine Course.

Teacher	Course	Text
Smith	Data Structures	Bartram
Smith	Data Management	Martin
Hall	Compilers	Hoffman
Brown	Data Structures	Horowitz

B.

(1)

A relation is said to be in Third Normal Form (3NF), if it is in 2NF and when no non key attribute is transitively dependent on the primary key i.e., there is no transitive dependency. Also, it should satisfy one of the below given conditions. For the function dependency C->D:

- C should be a super key
- D should be a prime attribute i.e, D should be a part of the candidate key.

3NF is used to reduce data duplication and to attain data integrity.

For the relation R(L, M, N, O, P) with functional dependencies as {L->M, MN->P, PO->L}:

```
The candidate keys will be : {LNO, MNO, NOP}
      as the closure of LNO = {L, M, N, O, P}
            closure of MNO = {L, M, N, O, P}
            closure of NOP = {L, M, N, O, P}
```

This relation is in 3NF as it is already in 2NF and has no transitive dependency. Also there is no non prime attribute that is deriving a non prime attribute.

(2)

BCNF stands for Boyce-Codd normal form and was made by R.F Boyce and E.F Codd in 1974.A functional dependency is said to be in BCNF if these properties hold:

- It should already be in 3NF.
- For a functional dependency say P->Q, P should be a super key.

BCNF is an extension of 3NF and it is has more strict rules than 3NF. Also, it is considered to be stronger than 3NF.

For the relation R(A, B, C, D) with functional dependencies as {A->B, A->C, C->D, C->A}:

```
The candidate keys will be : {A, C}
      as the closure of A = {A, B, C, D}
            closure of C = {A, B, C, D}
```

This relation is in BCNF as it is already in 3NF (there is no prime attribute deriving no prime attribute) and on the left hand side of the functional dependency there is a candidate key.

(3)

The difference between 3NF and BCNF:

	3NF	**BCNF**
1.	In 3NF there should be no transitive dependency that is no non prime attribute should be transitively dependent on the candidate key.	In BCNF for any relation A->B, A should be a super key of relation.
2.	It is less stronger than BCNF.	It is comparatively more stronger than 3NF.
3.	The redundancy is high in 3NF.	The redundancy is comparatively low in BCNF.
4.	In 3NF there is preservation of all functional dependencies.	In BCNF there may or may not be preservation of all functional dependencies.
5.	It is comparatively easier to achieve.	It is difficult to achieve.
6.	Lossless decomposition can be achieved by 3NF.	Lossless decomposition is hard to achieve in BCNF.

C.

(1)

F = {{A, B}→{C}, {A}→{D, E}, {B}→{F}, {F}→{G, H}, {D}→{I, J}}.

$\{A,B\}^+ = \{A,B,C\}$

Because {A}→{D, E}, {B}→{F}

$\{A,B\}^+ = \{A,B,C,D,E,F\}$

Because {F}→{G, H}, {D}→{I, J}}

$\{A,B\}^+ = \{A,B,C,D,E,F,G,H,I,J\} = R$

$\{A\}^+ = \{A, D, E\}$

Because {D}→{I, J}}

$\{A\}^+ = \{A, D, E, I, J\}$

$\{B\}^+ = \{B,F\}$

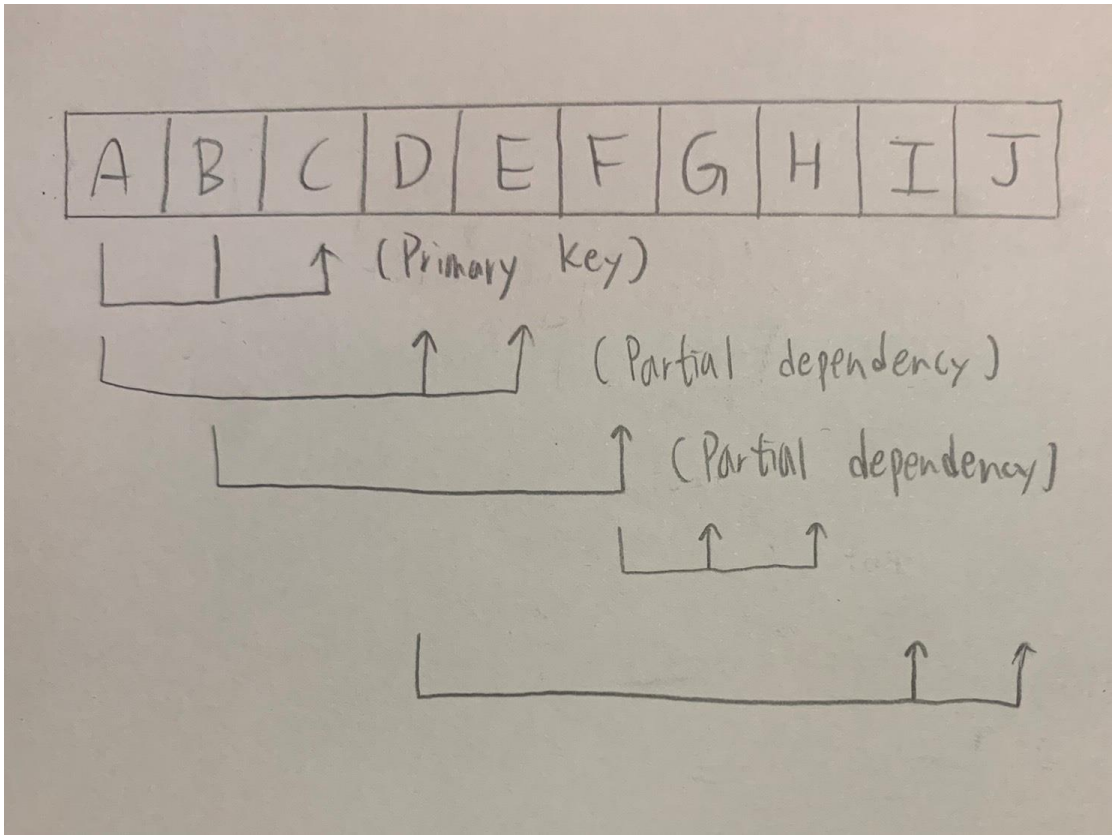
Because {F}→{G, H}

$\{B\}^+ = \{B,F,G,H\}$

$\{F\}^+ = \{F, G, H\}$

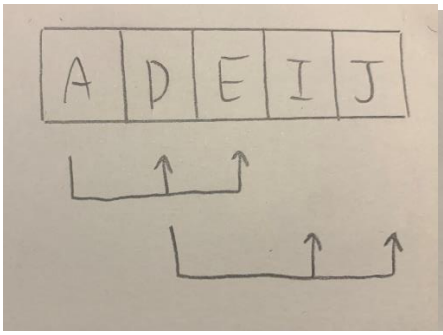
$\{D\}^+ = \{D, I, J\}\}$

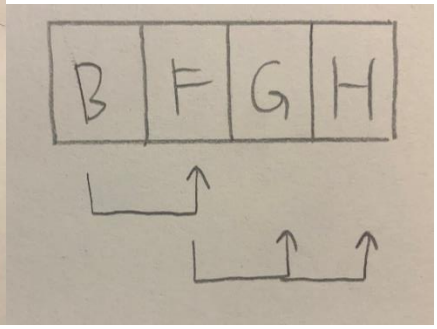
{A,B} form keys of R

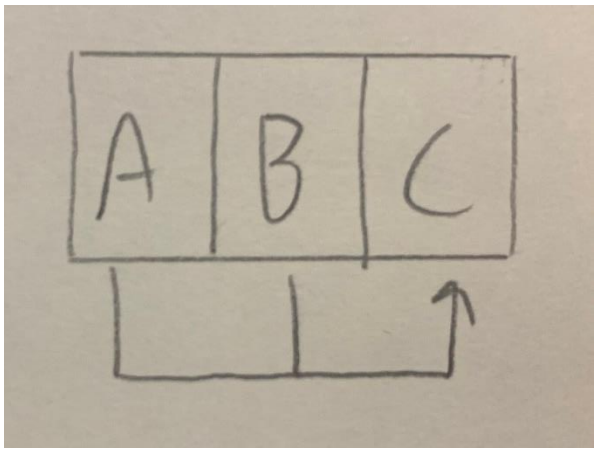


(2)

There are two partial dependency in R, so we need to split them apart from R to meet 2NF. I create a new relation R2 {A,D,E,I,J} (I, J is because transitive dependency {D}→{I, J}), R3 {B,F,G,H} (GH is because transitive dependency {F}→{G, H}), and R1 {A,B,C} (the remain opponents of R).







D.

BOOK (Book_title, Author_name, Book_type, List_price, Author_affil, Publisher)
Book_title → Publisher, Book_type
Book_type → List_price
Author_name → Author_affil

Substitute (Book_title, Author_name, Book_type, List_price, Author_affil, Publisher) to (A, B, C, D, E, F)
R = (A, B, C, D, E, F)
F = { A → CF, C → D, B → E }
Compute $(AB)^+$
$(A,B)^+$ ={A,B}
Because A→ CF, B→ E
$(A,B)^+$ ={A,B,C,E,F}
Because C→ D
$(A,B)^+$ ={A,B,C,D,E,F} = R

Compute $(A)^+$	Compute $(B)^+$
$(A)^+$= {A,C,F} Because C→ D $(A)^+$= {A,C,D,F}	$(B)^+$={B,E}

{ A, B } form the key of R

The primary key of this relation is { Book_title , Author_name }, since {Book_title}+, {Book_type}+, {Author_name}+ are all not BOOK, only { Book_title , Author_name }+ = BOOK.

So, there are two partial dependency in this relation, which are Book_title →

Publisher, Book_type and Author_name → Author_affil. It means that this relation is in 1NF since it violates 2NF rules.