RAMAIAH
Institute of Technology

M.S. Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of Computer Science and Engineering

Course Name: Database Systems

Course Code: CS52

Credits: 3:1:0

UNIT 4

Term: Oct 2021 – Feb 2022

Faculty:
Dr. Sini Anna Alex

3 Normal Forms Based on Primary Keys

Normalization of Relations

Practical Use of Normal Forms

Definitions of Keys and Attributes Participating in Keys

First Normal Form

Second Normal Form

Third Normal Form

Definitions of Keys and Attributes Participating in Keys

A **superkey** of a relation schema R = {A1, A2, ..., An} is a set of attributes S *subset-of* R with the property that no two tuples t1 and t2 in any legal relation state r of R will have t1[S] = t2[S]

A **key** K is a **superkey** with the *additional property* that removal of any attribute from K will cause K not to be a superkey any more.

Definitions of Keys and Attributes Participating in Keys

If a relation schema has more than one key, each is called a **candidate** key.

- One of the candidate keys is *arbitrarily* designated to be the **primary key**, and the others are called **secondary keys**.

A **Prime attribute** must be a member of *some* candidate key

A **Nonprime attribute** is not a prime attribute—that is, it is not a member of any candidate key.

Finding Candidate Keys and Super Keys of a Relation using FD set

The **set of attributes** whose attribute closure is set of all attributes of relation is called super key of relation.

Consider the following FD set. **{E-ID->E-NAME, E-ID->E-CITY, E-ID->E-STATE, E-CITY->E-STATE}**

Let us calculate attribute closure of different set of attributes: **EMPLOYEE(E-ID, E-NAME, E-CITY, E-STATE)**

(E-ID)+ = {E-ID, E-NAME,E-CITY,E-STATE}

(E-ID,E-NAME)+ = {E-ID, E-NAME,E-CITY,E-STATE}

(E-ID,E-CITY)+ = {E-ID, E-NAME,E-CITY,E-STATE}

(E-ID,E-STATE)+ = {E-ID, E-NAME,E-CITY,E-STATE}

(E-ID,E-CITY,E-STATE)+ = {E-ID, E-NAME,E-CITY,E-STATE}

(E-NAME)+ = {E-NAME}

(E-CITY)+ = {E-CITY,E-STATE}

Normalization of Relations

Normalization:

- The process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations.
- It is important that a database is **normalized to minimize redundancy (duplicate data) and to ensure only related data is stored in each table**.
- Normalization makes sure that all of your data looks and reads the same way across all records.

The benefits of normalization include: **Searching, sorting, and creating indexes is faster**

Normal form:

- Condition using keys and FDs of a relation to certify whether a relation schema is in a particular normal form

Normalization of Relations

2NF, 3NF, BCNF

- based on keys and FDs of a relation schema

4NF

- based on keys, multi-valued dependencies : MVDs;

5NF

- based on keys, join dependencies : JDs

Additional properties may be needed to ensure a good relational design (lossless join, dependency preservation)

Practical Use of Normal Forms

Normalization is carried out in practice so that the resulting designs are of high quality and meet the desirable properties

The practical utility of these normal forms becomes questionable when the constraints on which they are based are *hard to understand* or to *detect*

The database designers *need not* normalize to the highest possible normal form

- (usually up to 3NF and BCNF. 4NF rarely used in practice.)

Denormalization:

- The process of storing the join of higher normal form relations as a base relation—which is in a lower normal form

First Normal Form

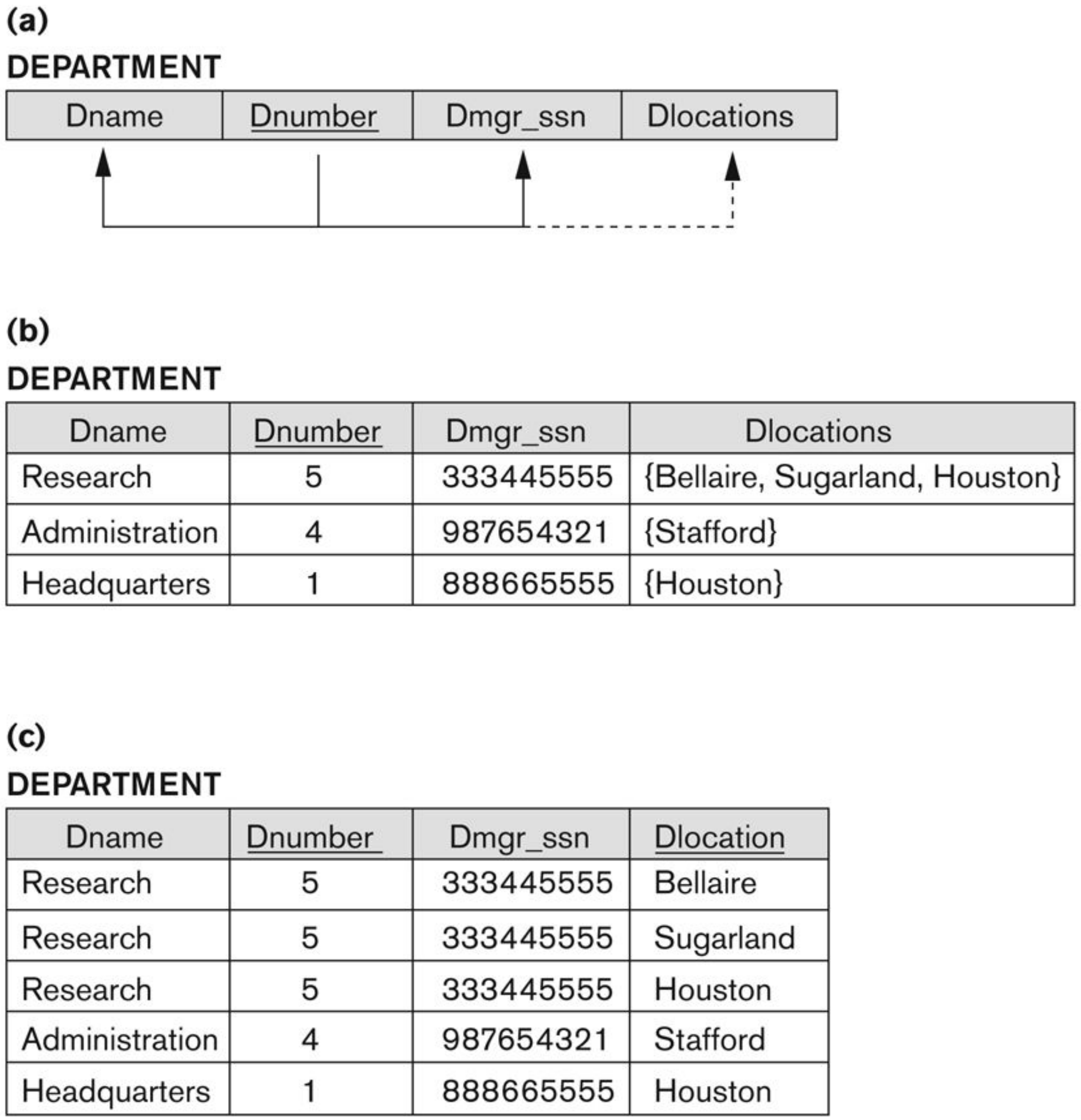
Disallows

- composite attributes
- multivalued attributes
- **nested relations**; attributes whose values for an *individual tuple* are non-atomic

Considered to be part of the definition of relation .

Most RDBMSs allow only those relations to be defined that are in First Normal Form

Normalization into 1NF



(a)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations

(b)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

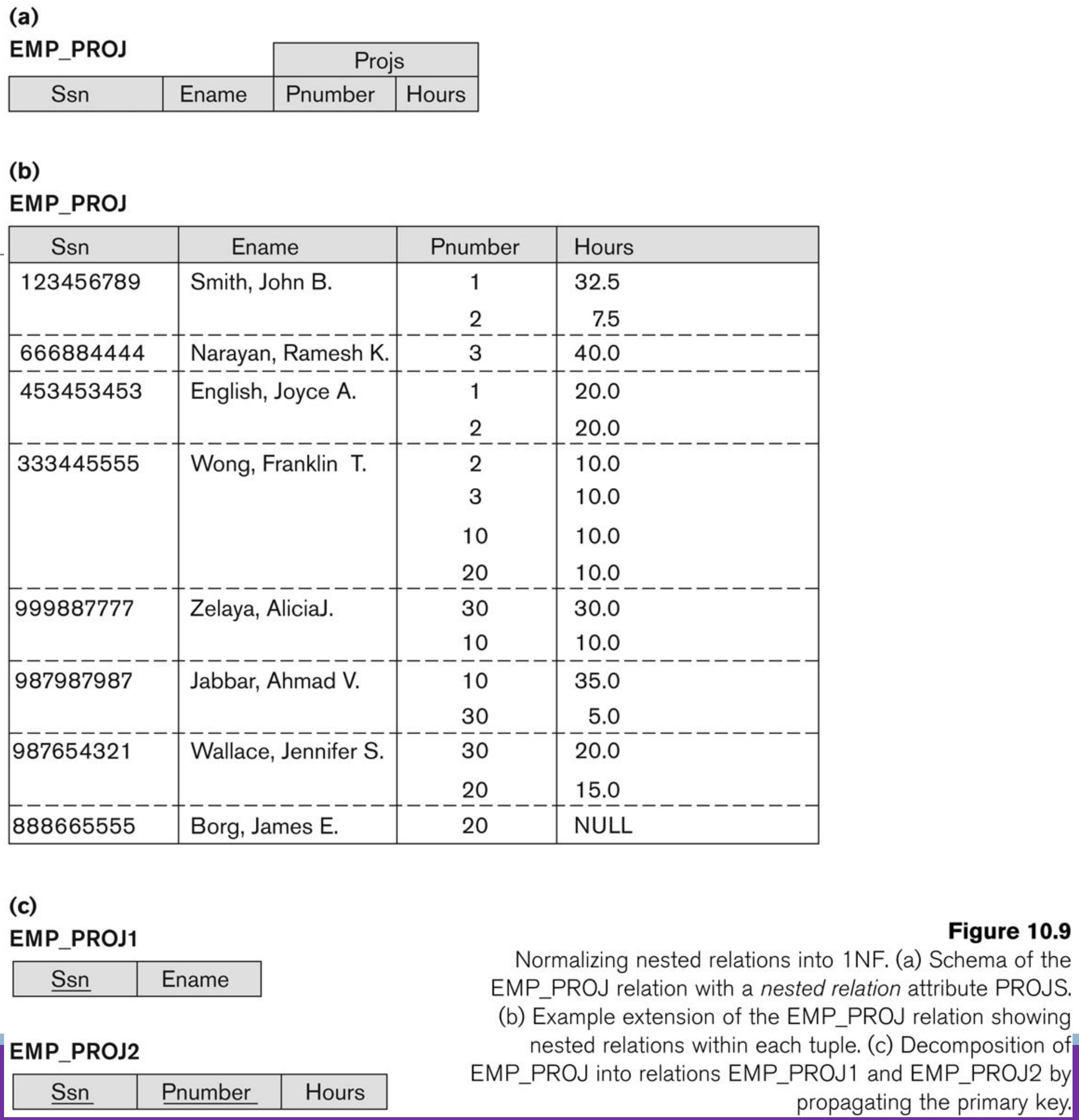
(c)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocation
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Figure 10.8
Normalization into 1NF. (a) A relation schema that is not in 1NF. (b) Example state of relation DEPARTMENT. (c) 1NF version of the same relation with redundancy.

Normalization nested relations into 1NF



(a)

EMP_PROJ

Ssn	Ename	Projs	
		Pnumber	Hours

(b)

EMP_PROJ

Ssn	Ename	Pnumber	Hours
123456789	Smith, John B.	1	32.5
		2	7.5
666884444	Narayan, Ramesh K.	3	40.0
453453453	English, Joyce A.	1	20.0
		2	20.0
333445555	Wong, Franklin T.	2	10.0
		3	10.0
		10	10.0
		20	10.0
999887777	Zelaya, AliciaJ.	30	30.0
		10	10.0
987987987	Jabbar, Ahmad V.	10	35.0
		30	5.0
987654321	Wallace, Jennifer S.	30	20.0
		20	15.0
888665555	Borg, James E.	20	NULL

(c)

EMP_PROJ1

Ssn	Ename

EMP_PROJ2

Ssn	Pnumber	Hours

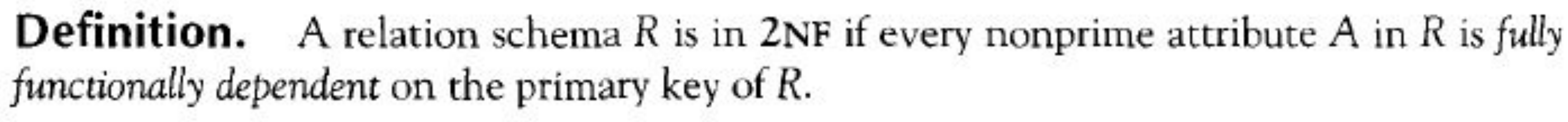
Figure 10.9

Normalizing nested relations into 1NF. (a) Schema of the EMP_PROJ relation with a *nested relation* attribute PROJS. (b) Example extension of the EMP_PROJ relation showing nested relations within each tuple. (c) Decomposition of EMP_PROJ into relations EMP_PROJ1 and EMP_PROJ2 by propagating the primary key.

Second Normal Form

Second normal form (2NF) is based on the concept of *full functional dependency.* A functional dependency X *Y* is a **full functional dependency** if removal of any attribute A from X means that the dependency does not hold any more.

Functional dependency X *Y* is a **partial dependency** if some attribute A ϵ X can be removed from X and the dependency still holds.



Definition. A relation schema *R* is in 2NF if every nonprime attribute *A* in *R* is *fully functionally dependent* on the primary key of *R*.

Second Normal Form contd………

A relation schema R is in **second normal form (2NF)** if every non-prime attribute A in R is fully functionally dependent on the primary key

R can be decomposed into 2NF relations via the process of 2NF normalization or “second normalization”

2NF Example

Key: SSN, PNUMBER

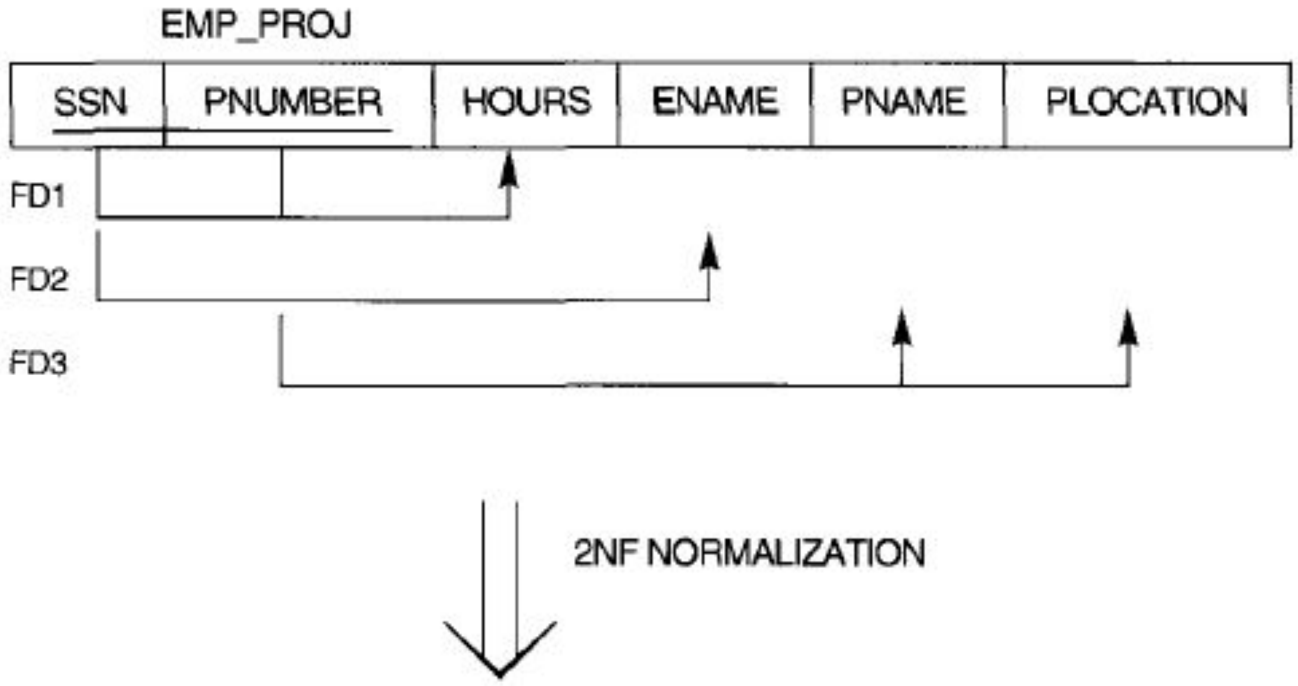
FDs:

SSN, PNUMBER-> HOURS

SSN->ENAME

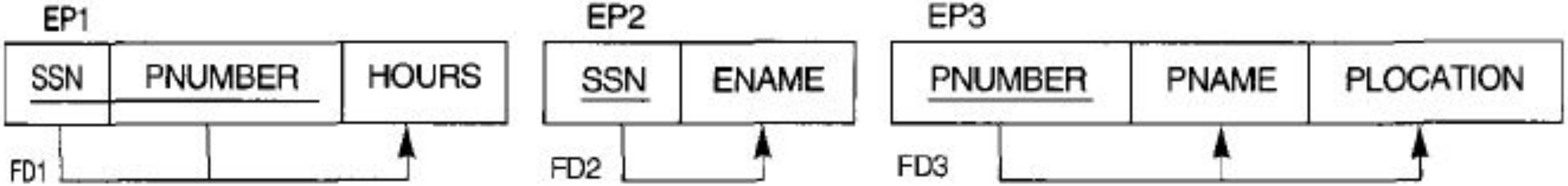
PNUMBER->PNAME

PNUMBER->PLOCATION



EMP_PROJ

SSN	PNUMBER	HOURS	ENAME	PNAME	PLOCATION



EP1

SSN	PNUMBER	HOURS

EP2

SSN	ENAME

EP3

PNUMBER	PNAME	PLOCATION

Tutorial 2: **Decompose a non-2NF relation to a 2NF relation**

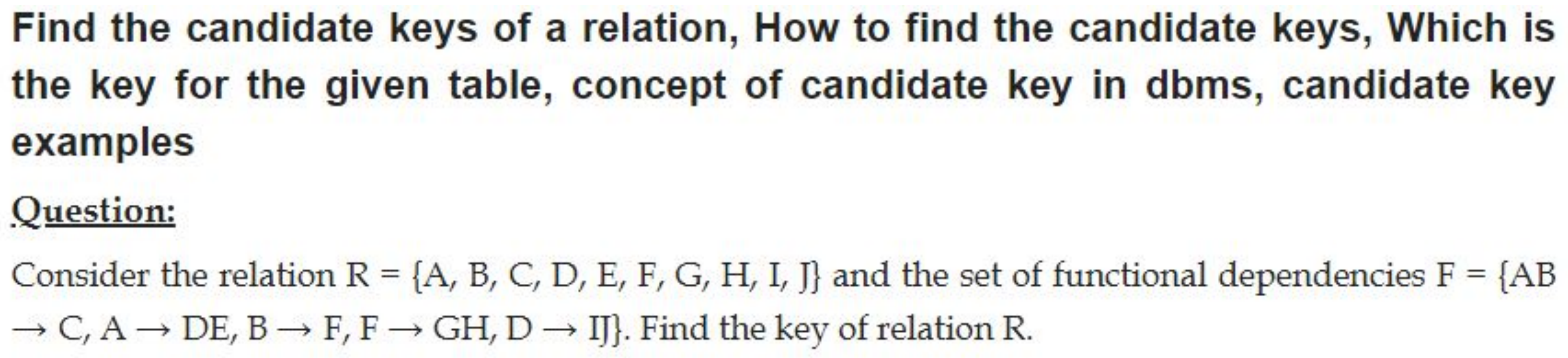
Assume a relation R (A, B, C, D, E) with the following set of functional dependencies;

F = {AB → C, B → D, E → D}

The key for this relation is ABE. Then, all three given FDs are partial dependencies, viz., AB → C, B → D, and E → D.
Step 1: separate tables for partial dependencies; hence, R1 (ABC), R2 (BD) and R3 (ED).

Step 2: remove RHS of these two partial FDs from R; hence, R4(A, B, E).

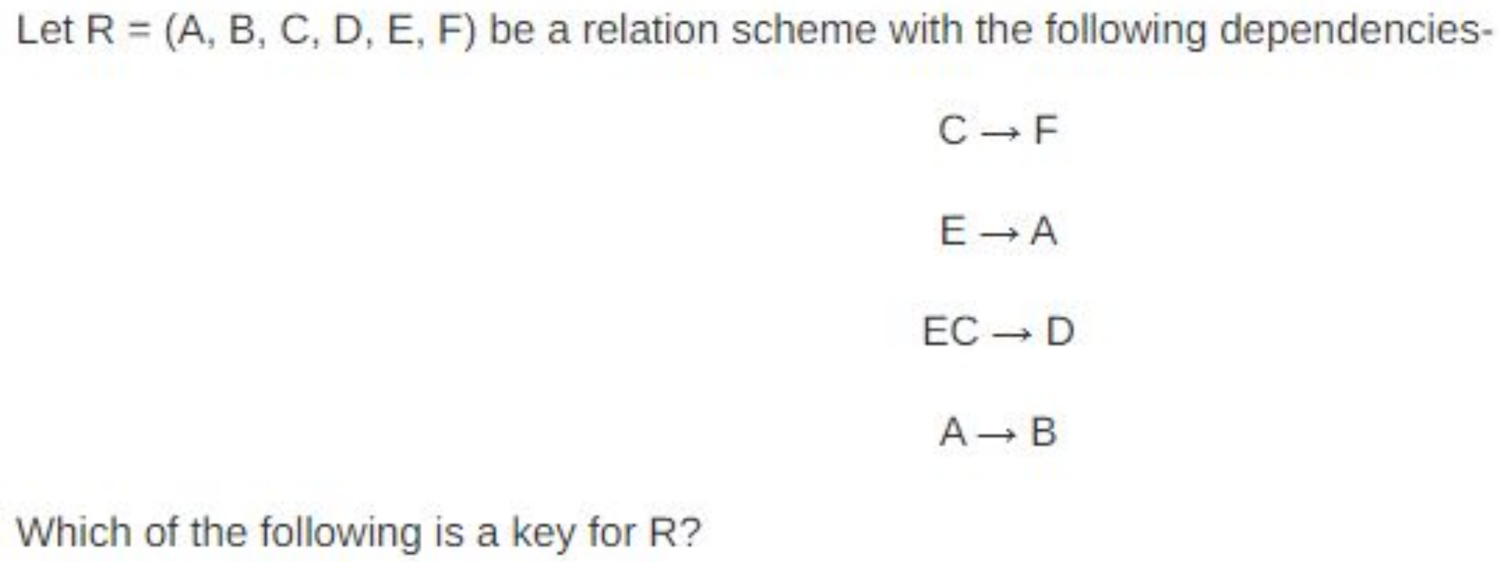
Thus, we have four tables ***R1 (ABC), R2 (BD), R3 (ED)*** and ***R4 (ABE)***.



Find the candidate keys of a relation, How to find the candidate keys, Which is the key for the given table, concept of candidate key in dbms, candidate key examples

Question:

Consider the relation R = {A, B, C, D, E, F, G, H, I, J} and the set of functional dependencies F = {AB → C, A → DE, B → F, F → GH, D → IJ}. Find the key of relation R.



Let R = (A, B, C, D, E, F) be a relation scheme with the following dependencies-

$$C \rightarrow F$$

$$E \rightarrow A$$

$$EC \rightarrow D$$

$$A \rightarrow B$$

Which of the following is a key for R?

Tutorial 4: Compute Minimal Cover

1. Find the minimal cover of the set of functional dependencies given;

{A → C, AB → C, C → DI, CD → I, EC → AB, EI → C}

2. F = { AB -> C, C -> A, BC -> D, ACD -> B, D -> E, D -> G, BE -> C, CG -> B, CG -> D, CE -> A, CE -> G}

Solution:

Q1: MC = {A → C, C → D, C → I, EC → A, EC → B, EI → C}

Q2:

Minimal Cover 1: {AB -> C, **C -> A**, BC -> D, **CD -> B**, D -> E, D -> G, BE -> C, **CG -> D**, CE -> G}

Minimal Cover 2 : {AB -> C, **C -> A**, **BC -> D**, D -> E, **D -> G**, BE -> C, **CG -> B**, CE -> G}

Thank you