SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE NAME : 23CS207 - DATABASE MANAGEMENT SYSTEMS

II YEAR / IV SEMESTER

Unit 3- Database Design

Topic 1, 2 : **Normalization of Database**

Normalization of Database

- Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, U
- It is a multi-step process that puts data into tabular form by removing duplicated data from the relation tables.pdate and Deletion Anomalies.

- Normalization is used for mainly two purpose,

1. Eliminating redundant(useless) data.
2. Ensuring data dependencies make sense i.e data is logically stored.

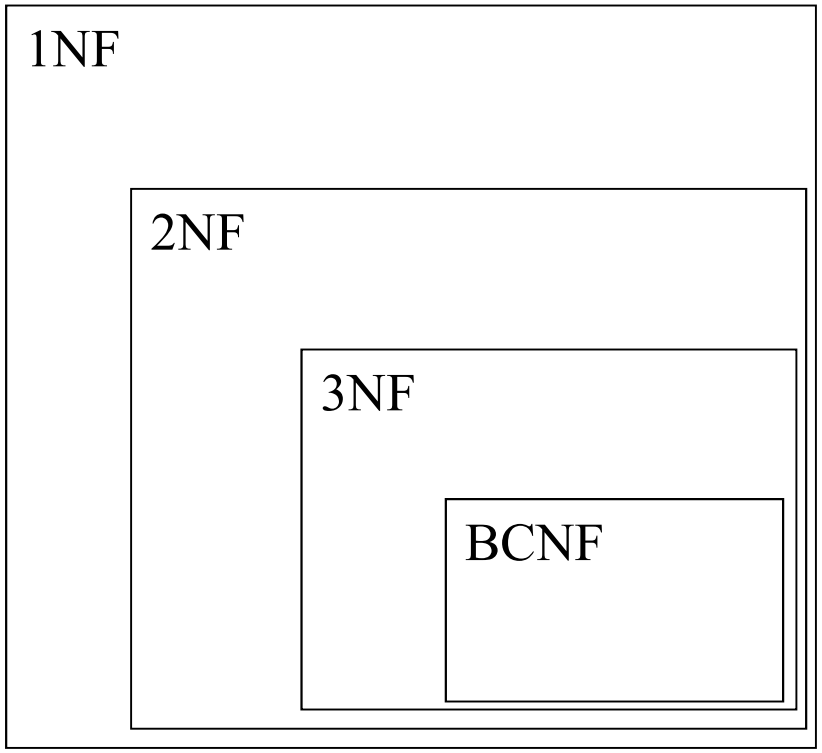
What is normalization?

Normalization is a process that "improves" a database design by generating relations that are of higher normal forms.

The *objective* of normalization:

"*to create relations where every dependency is on the key, the whole key, and nothing but the key*".

Normalization



a relation in BCNF, is also in 3NF

a relation in 3NF is also in 2NF

a relation in 2NF is also in 1NF

Problem Without Normalizatio...

- Without Normalization, it becomes difficult to handle and update the database, without facing data loss. Insertion, Updation and Deletion Anomalies are very frequent if Database is not Normalized. To understand these anomalies let us take an example of **Student** table.

S_id	S_Name	S_Address	Subject_opted
401	Adam	Noida	Bio
402	Alex	Panipat	Maths
403	Stuart	Jammu	Maths
404	Adam	Noida	Physics

- **Updation Anomaly :** To update address of a student who occurs twice or more than twice in a table, we will have to update **S_Address** column in all the rows, else data will become inconsistent.
- **Insertion Anamoly :** Suppose for a new admission, we have a Student id(S_id), name and address of a student but if student has not opted for any subjects yet then we have to insert **NULL** there, leading to Insertion Anamoly.
- **Deletion Anamoly :** If (S_id) 401 has only one subject and temporarily he drops it, when we delete that row, entire student record will be deleted along with it.

BREAK

TYPES OF NORMALIZATION

- First Normal Form
- Second Normal Form
- Third Normal Form
- Boyce – codd Normal Form
- Fourth Normal Form
- Fifth Normal Form

First Normal Form

First Normal Form

We say a relation is in **1NF** if all values stored in the relation are single-valued and atomic.

1NF places restrictions on the structure of relations. Values must be simple.

First Normal Form

The following in **not** in 1NF

EmpNum	EmpPhone	EmpDegrees
123	233-9876	
333	233-1231	BA, BSc, PhD
679	233-1231	BSc, MSc

EmpDegrees is a multi-valued field:

employee 679 has two degrees: *BSc* and *MSc*

employee 333 has three degrees: *BA, BSc, PhD*

First Normal Form

EmpNum	EmpPhone	EmpDegrees
123	233-9876	
333	233-1231	BA, BSc, PhD
679	233-1231	BSc, MSc

To obtain 1NF relations we must, without loss of information, replace the above with two relations - see next slide

First Normal Form

Employee

EmpNum	EmpPhone
123	233-9876
333	233-1231
679	233-1231

EmployeeDegree

EmpNum	EmpDegree
333	BA
333	BSc
333	PhD
679	BSc
679	MSc

An outer join between Employee and EmployeeDegree will produce the information we saw before

Second Normal Form

Second Normal Form

A relation is in **2NF** if it is in 1NF, and every non-key attribute is fully dependent on each candidate key. (That is, we don't have any partial functional dependency.)

- 2NF (and 3NF) both involve the concepts of key and non-key attributes.
- A *key attribute* is any attribute that is part of a key; any attribute that is not a key attribute, is a *non-key attribute*.
- Relations that are not in BCNF have data redundancies
- A relation in 2NF will not have any partial dependencies

Second Normal Form

Consider this **InvLine** table (in 1NF)**:**

InvNum	LineNum	ProdNum	Qty	InvDate

InvNum, LineNum ⟶ ProdNum, Qty

There are two candidate keys.

Qty is the only non-key attribute, and it is dependent on InvNum

InvNum ⟶ InvDate

Since there is a determinant that is not a candidate key, InvLine is **not BCNF**

InvLine is **not 2NF** since there is a partial dependency of InvDate on InvNum

InvLine is only in **1NF**

Second Normal Form

InvLine

InvNum	LineNum	ProdNum	Qty	InvDate

The above relation has redundancies: the invoice date is repeated on each invoice line.

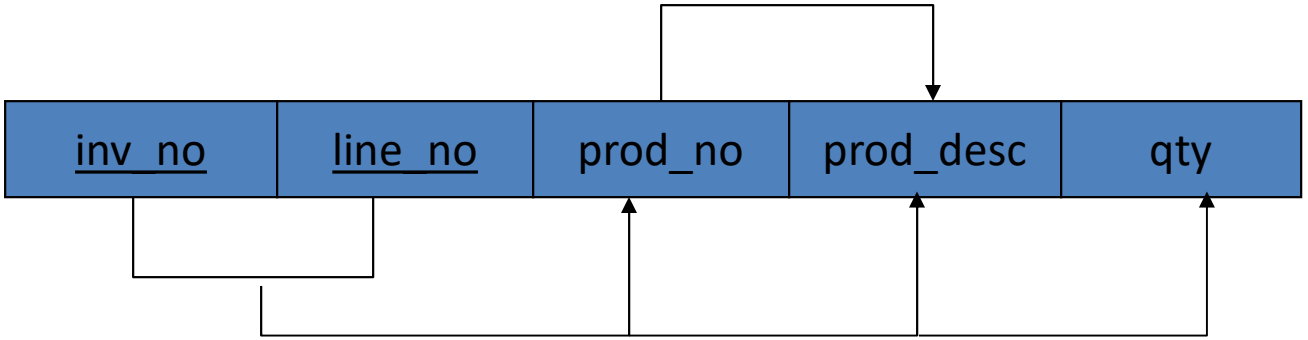
We can *improve* the database by decomposing the relation into two relations:

InvNum	LineNum	ProdNum	Qty

InvNum	InvDate

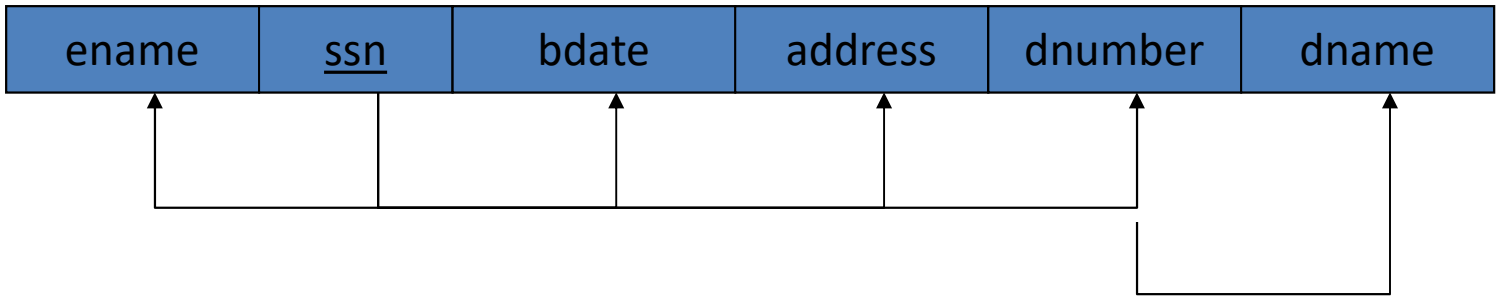
Question: What is the highest normal form for these relations? 2NF? 3NF? BCNF?

Is the following relation in 2NF?



2NF, but not in 3NF, nor in BCNF:

EmployeeDept



since dnumber is not a candidate key and we have:

dnumber → dname.

Third Normal Form

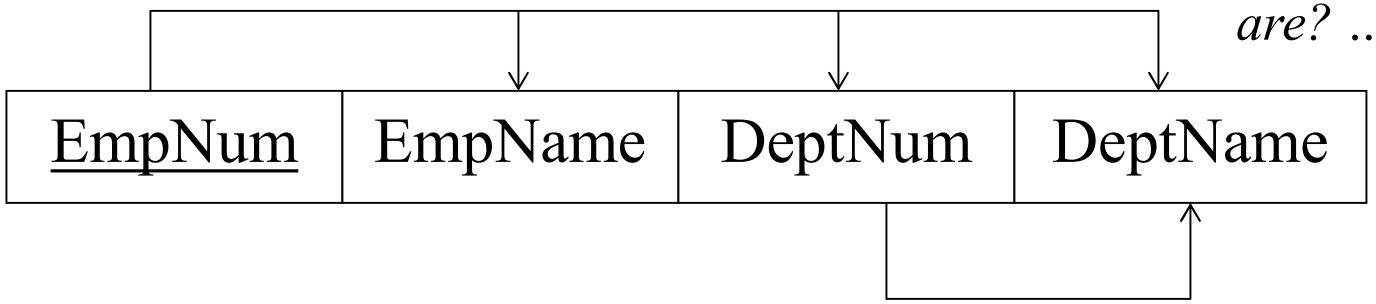
Third Normal Form

- A relation is in **3NF** if the relation is in 1NF and all determinants of *non-key* attributes are candidate keys
  That is, for any functional dependency: X → Y, where Y is a non-key attribute (or a set of non-key attributes), X is a candidate key.
- This definition of 3NF differs from BCNF only in the specification of non-key attributes - 3NF is weaker than BCNF. (BCNF requires all determinants to be candidate keys.)
- A relation in 3NF will not have any transitive dependencies of non-key attribute on a candidate key through another non-key attribute.

Third Normal Form

Consider this **Employee** relation

Candidate keys are? …



EmpName, DeptNum, and DeptName are non-key attributes.

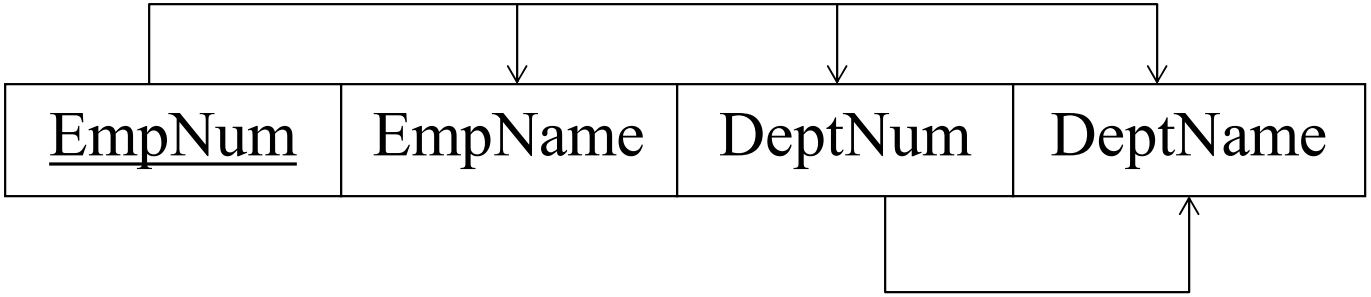
DeptNum determines DeptName, a non-key attribute, and DeptNum is not a candidate key.

Is the relation in 3NF? … no

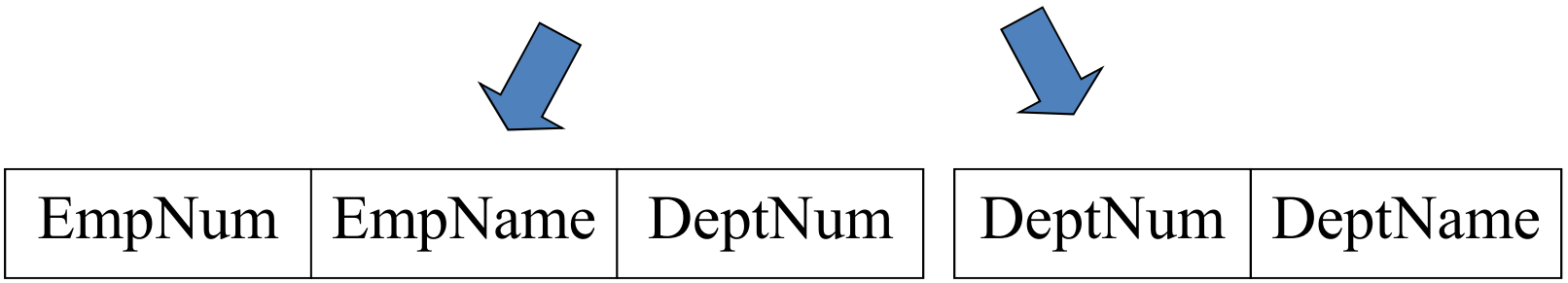
Is the relation in 2NF? … yes

Is the relation in BCNF? … no

Third Normal Form

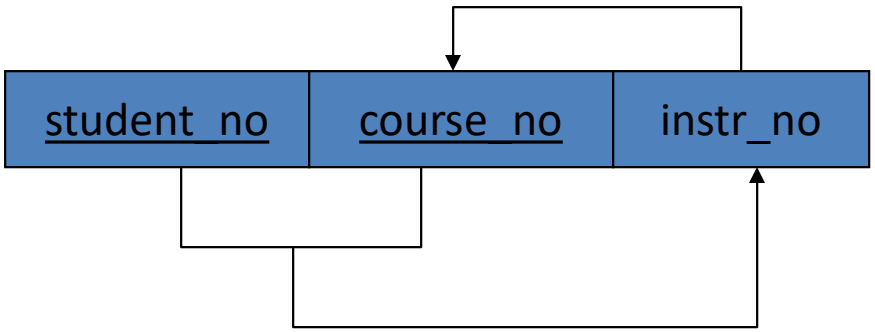


We correct the situation by decomposing the original relation into two 3NF relations. Note the decomposition is *lossless*.



Verify these two relations are in 3NF.

In 3NF, but not in BCNF:

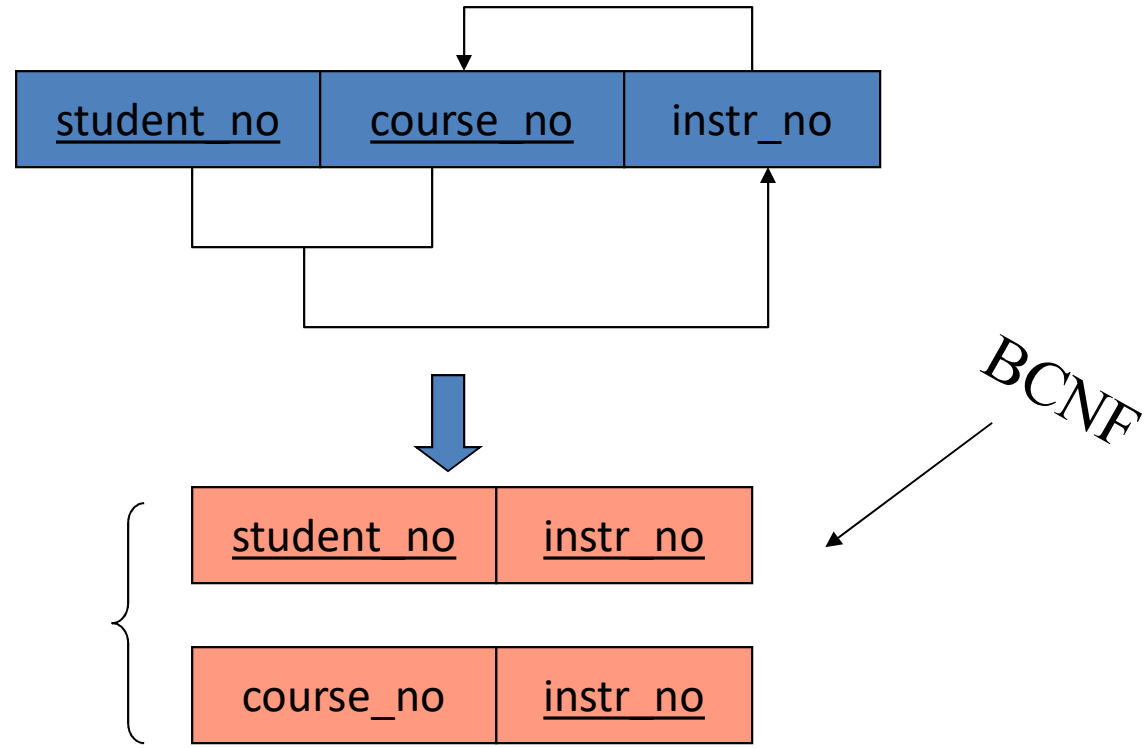


Instructor teaches one course only.

Student takes a course and has one instructor.

{student_no, course_no} → instr_no
instr_no → course_no

since we have instr_no → course-no, but instr_no is not a Candidate key.



$\{student\_no, instr\_no\} \rightarrow student\_no$

$\{student\_no, instr\_no\} \rightarrow instr\_no$

$instr\_no \rightarrow course\_no$

Evaluation

- Types of normalization.

a)_____

b)_____

c)_____

d)_____

e)_____

Evaluation

- Types of normalization.
- First Normal Form
- Second Normal Form
- Third Normal Form
- Boyce – codd Normal Form
- Fourth Normal Form
- Fifth Normal Form

THANK YOU