DATABASE SYSTEMS

CS – 355/ CE – 373

Instructor: Maria N. Samad

September 16th, 2024

FIRST NORMAL FORM (1NF)

- First normal form (1NF) states that the domain of an attribute must:
  - Include only atomic (simple, indivisible) values, and
  - The value of any attribute in a tuple must be a single value from the domain of that attribute
- Hence, 1NF disallows having a set of values, a tuple of values, or a combination of both as an attribute value for a single tuple
- In other words, 1NF disallows relations within relations or relations as attribute values within tuples
- The only attribute values permitted by 1NF are single atomic (or indivisible) values

FIRST NORMAL FORM (1NF)

- The basic rules are as follows:
  - For ***composite attributes***, each component becomes a separate attribute in the resultant 1NF
  - For ***multivalued attributes***, each item becomes a separate tuple, however, when defining the 1NF schema, the primary keys must be updated

EXAMPLE

- Consider the following relation (not in 1NF):

DEPARTMENT

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

- The following relation is normalized to 1NF:

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

FIRST-NORMAL FORM

- **Exercises – Composite attributes are mentioned in round brackets**:
  - *Cars (CarID, (License), Year, Cost)*
    - **1NF – *Cars (CarID, Letter, Digits, Year, Cost)***
  - *Cars (CarID, (License), (CarType), Year, Cost)*
    - **1NF – *Cars (CarID, Letter, Digits, Brand, Model, Year, Cost)***
  - *Book (BookNum, (BookDetails), YearPublished, Cost)*
    - **1NF – *Book (BookNum, Title, Author, ISBN, YearPublished, Cost)***
  - *Person(SSN, (Address), {PhoneNum})*
    - **1NF – *Person (SSN, Street, Zipcode, City, PhoneNum)***

SECOND-NORMAL FORM

- A relation schema *R* is in 2NF if

  (a) it is in 1NF, and

  (b) every nonprime attribute *A* in *R* is fully functionally dependent on the primary key of *R.*

- The test for 2NF involves testing for functional dependencies whose left-hand side attributes are part of the primary key.
- **If the primary key contains a single attribute, the test need not be applied at all, as that is considered as full dependency by default**

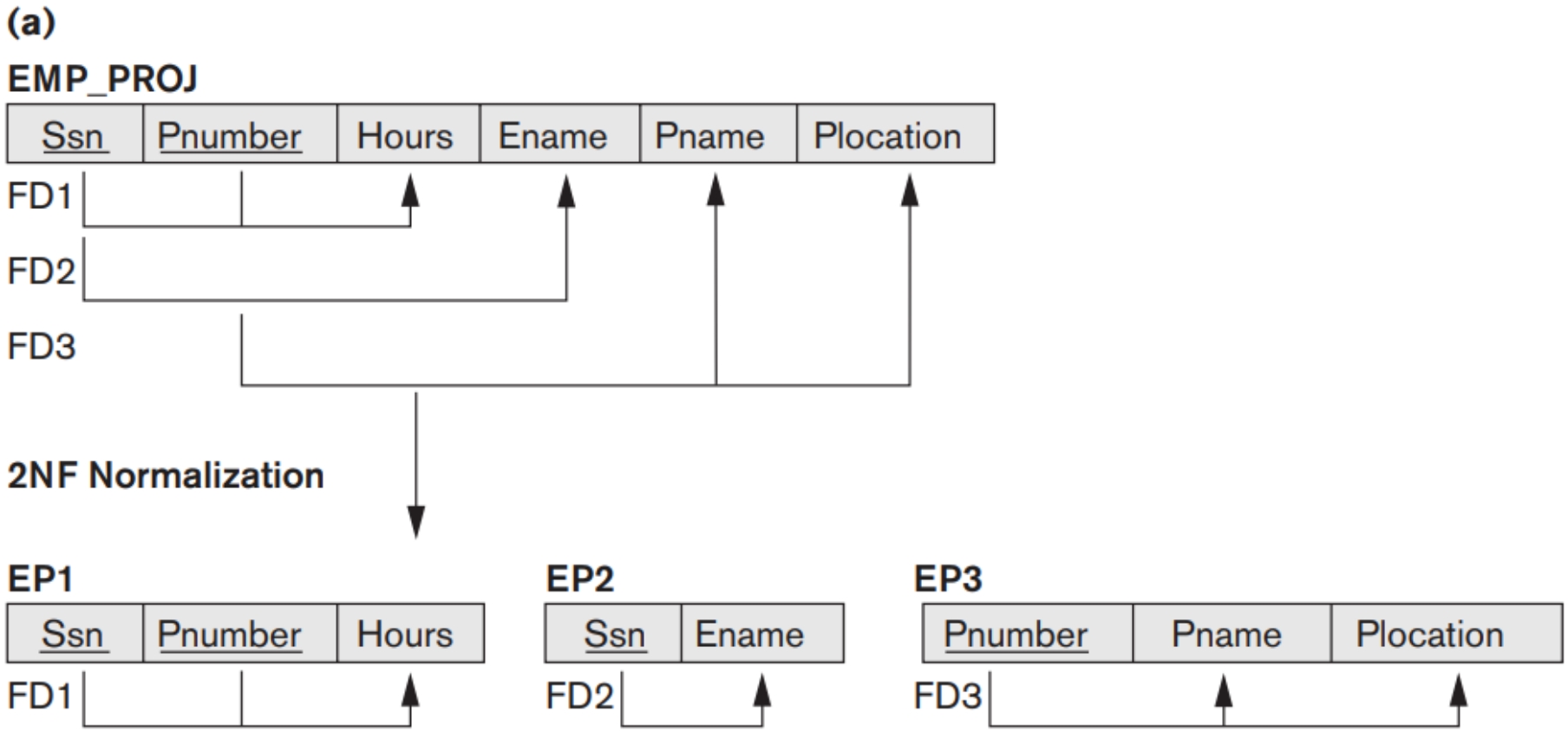
FULL vs PARTIAL FUNCTIONAL DEPENDENCY

- For example, consider the following schema:
  Emp_Proj (SSN, ProjectID, EmpName, WorkedHours, ProjectName, ProjectLocation)
- We first checked if the following are fully or partially dependent:
  - {SSN, ProjectID} $\rightarrow$ WorkedHours?
    - **FULL FD**
  - {SSN, ProjectID} $\rightarrow$ EmpName?
    - **PARTIAL FD**
  - {SSN, ProjectID} $\rightarrow$ ProjectName?
    - **PARTIAL FD**
  - {SSN, ProjectID} $\rightarrow$ ProjectLocation?
    - **PARTIAL FD**
  - Therefore, this relation is in 1NF because all atomic, single-valued attributes.
  - However, not every non-prime attribute is fully functional dependent on the prime attributes, so NOT in 2NF

NORMALIZATION TO 2NF

- If a relation schema is not in 2NF, it can be second normalized or 2NF normalized into a number of 2NF relations in which nonprime attributes are associated only with the part of the primary key on which they are fully functionally dependent
- Therefore, the functional dependencies in the previous example leads to the decomposition of ***Emp_Proj*** into the three relation schemas ***EP1***, ***EP2***, and ***EP3*** shown in the next slide, each of which is in 2NF

NORMALIZATION TO 2NF



SECOND-NORMAL FORM

- **Exercise 1**:
  - For the given relation schema:

    Lots (PropertyID, CountyName, LotNum, Area_in_meters, Price, TaxRate)
  - Given functional dependencies are:
    - PropertyID $\rightarrow$ LotNum
    - PropertyID $\rightarrow$ Area_in_meters
    - PropertyID $\rightarrow$ Price
  - Is this schema in 2NF? If not, decompose it to 2NF

SECOND-NORMAL FORM

- **Solution**:
  - First check if the given schema is in 1NF?
    - No composite nor multivalued attributes, so it is considered as 1NF
  - Then check for 2NF?
    - Check for functional dependencies
    - Individual dependencies are:
      - PropertyID $\rightarrow$ LotNum
      - PropertyID $\rightarrow$ Area_in_meters
      - PropertyID $\rightarrow$ Price

SECOND-NORMAL FORM

- **Solution**:
  - For combined attributes, check if any subset will fulfill the dependency requirement
  - {PropertyID, CountyName} $\rightarrow$ LotNum?
    - PropertyID $\rightarrow$ LotNum? Possible
    - Hence **PARTIAL FD**
  - {PropertyID, CountyName} $\rightarrow$ Area_in_meters
    - PropertyID $\rightarrow$ Area_in_meters? Possible
    - Hence **PARTIAL FD**

SECOND-NORMAL FORM

- **Solution**:
  - For combined attributes, check if any subset will fulfill the dependency requirement
  - {PropertyID , CountyName} $\rightarrow$ Price
    - PropertyID $\rightarrow$ Price? Possible
    - Hence **PARTIAL FD**
  - {PropertyID, CountyName} $\rightarrow$ TaxRate
    - PropertyID $\rightarrow$ TaxRate? Not Possible
    - CountyName $\rightarrow$ TaxRate? Not Possible
    - Hence **FULL FD**
  - As there are partial dependencies present in this schema, this schema is not in 2NF, and must be decomposed into multiple 2NFs

SECOND-NORMAL FORM

- **Solution**:
  - *Lots1 (PropertyID, LotNum, Area_in_meters, Price)*
  - *Lots2 (PropertyID, CountyName, TaxRate)*

- Now these two schemas are fully functional dependent, hence in 2NF

SECOND-NORMAL FORM

- **Exercise 2**:
  - For the given relation schema:
    Event_Venue(event_id, venue_id, event_name, venue_location, event_date)
  - Given functional dependencies are:
    - event_id $\rightarrow$ event_name
    - event_id $\rightarrow$ event_date
    - venue_id $\rightarrow$ venue_location
  - Is this schema in 2NF? If not, decompose it to 2NF

SECOND-NORMAL FORM

- **Solution**:
  - First check if the given schema is in 1NF?
    - No composite nor multivalued attributes, so it is considered as 1NF
  - Then check for 2NF?
    - Check for functional dependencies
    - Individual dependencies are:
      - event_id $\rightarrow$ event_name
      - event_id $\rightarrow$ event_date
      - venue_id $\rightarrow$ venue_location

SECOND-NORMAL FORM

- **Solution**:
  - For combined attributes, check if any subset will fulfill the dependency requirement
  - {event_id, venue_id} $\rightarrow$ event_name?
    - event_id $\rightarrow$ event_name? Possible
    - Hence **PARTIAL FD**
  - {event_id, venue_id} $\rightarrow$ event_date?
    - event_id $\rightarrow$ event_date? Possible
    - Hence **PARTIAL FD**
  - {event_id, venue_id} $\rightarrow$ venue_location?
    - venue_id $\rightarrow$ venue_location? Possible
    - Hence **PARTIAL FD**
  - As there are partial dependencies present in this schema, this schema is not in 2NF, and must be decomposed into multiple 2NFs

SECOND-NORMAL FORM

- **Solution**:
  - *Event (event_id, event_name, event_date)*
  - *Venue (venue_id, venue_location)*
  - *Event_Venue (event_id, venue_id)*

THIRD-NORMAL FORM

- Third normal form (3NF) is based on the concept of transitive dependency.
- A functional dependency $X \rightarrow Y$ in a relation schema R is a transitive dependency
  - If there exists a set of attributes Z in R that is neither a candidate key nor a subset of any key of R, and
  - both $X \rightarrow Z$ and $Z \rightarrow Y$ hold
- A relation schema R is in 3NF if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key

THIRD-NORMAL FORM

- ***Theorem:***
  - For every Functional Dependency given in the schema, check $X \rightarrow A$
  - If one of the following conditions are satisfied, then the relation schema is considered to be in 3NF
    1. X is a superkey of R
    2. A is a prime attribute of R
  - For example, in the given relation schemas:
    Lots1 (PropertyID, LotNum, Area_in_meters, Price)
    Lots2 (PropertyID, CountyName, TaxRate)
    We may have correct dependencies, but we also know that we can find out the price of any lot if its area is given.
  - Now check whether both these schemas are in 3NF or not?
  - Lots2 has no transitive dependency, so that is in 3NF
  - What about Lots1?

THIRD-NORMAL FORM

- *Lots1 (PropertyID, LotNum, Area_in_meters, Price)*
  - Check for all the transitive dependencies from Prime attributes to Non-prime attributes
  - PropertyID $\rightarrow$ LotNum? No transitive dependency here, as there is no other attribute, Z in the schema that will give PropertyID $\rightarrow$ Z and Z $\rightarrow$ LotNum
    - Then no need to check if determinant is superkey or not, or dependent is prime attribute or not
  - PropertyID $\rightarrow$ Area_in_meters? No transitive dependency here, as there is no other attribute, Z in the schema that will give PropertyID $\rightarrow$ Z and Z $\rightarrow$ Area_in_meters
    - Then no need to check if determinant is superkey or not, or dependent is prime attribute or not

THIRD-NORMAL FORM

- *Lots1 (PropertyID, LotNum, Area_in_meters, Price)*
  - Check for all the transitive dependencies from Prime attributes to Non-prime attributes
  - PropertyID $\rightarrow$ Price? There is transitive dependency here, as there is Area attribute in the schema that can give PropertyID $\rightarrow$ Area_in_meters and Area_in_meters $\rightarrow$ Price
    - Then check if determinant is superkey or not?
      - For PropertyID $\rightarrow$ Area_in_meters? PropertyID is a primary key, so this condition holds then not violating the rules of 3NF
      - For Area_in_meters $\rightarrow$ Price? Area_in_meters is not a superkey, so check the dependent then. Price is not a prime attribute either, so this transitive dependency causes this schema to be **NOT in 3NF**

NORMALIZATION TO 3NF

- If a relation schema is not in 3NF, it can be decomposed by removing the attribute from schema that is causing the violation and placing it in another relation
- Therefore, the transitive dependency in the previous example leads to the decomposition of ***Lots1*** into the two relation schemas ***Lots1a*** and ***Lots1b*** shown the next slide, each of which is in 3NF.
- Solution:
  - ***Lots1a (PropertyID, LotNum, Area_in_meters)***
  - ***Lots1b (Area_in_meters, Price)***

THIRD-NORMAL FORM

- **Exercise**:
  - For the given relation schema:

    Emp_Dept (SSN, EmpName, Bdate, Address, DeptNum, DeptName, DeptManagerSSN)
  - Birth date and address need not be divided into sub-parts
  - The possible functional dependencies are:
    - SSN $\rightarrow$ EmpName
    - SSN $\rightarrow$ Bdate
    - SSN $\rightarrow$ Address
    - SSN $\rightarrow$ DeptNum
    - SSN $\rightarrow$ DeptName
    - DeptNum $\rightarrow$ DeptName
    - DeptNum $\rightarrow$ DeptManagerSSN
  - Is this schema in 3NF? If not, decompose it to 3NF

THIRD-NORMAL FORM

- **Solution**:
  - First check if it is in 1NF?
    - As given in the question, birth date and address are not treated as composite attributes in this schema, which means there are no composite or multivalued attributes in the schema
    - Thus, it is already in 1NF
  - Then check if it is in 2NF?
    - As there is no combined primary key, so there won't be any partial dependencies possible, hence it is considered in 2NF already

THIRD-NORMAL FORM

- **Solution**:
  - Finally check if it is in 3NF or not?
    - For that, we need to find any transitive dependencies from prime attributes to non-prime attributes
  - SSN $\rightarrow$ EmpName? No transitive dependency here, as there is no other attribute, Z in the schema that will give SSN $\rightarrow$ Z and Z $\rightarrow$ EmpName
    - Then no need to check if determinant is superkey or not, or dependent is prime attribute or not
  - SSN $\rightarrow$ Bdate? No transitive dependency here, as there is no other attribute, Z in the schema that will give SSN $\rightarrow$ Z and Z $\rightarrow$ Bdate
    - Then no need to check if determinant is superkey or not, or dependent is prime attribute or not

THIRD-NORMAL FORM

- **Solution**:
  - SSN $\rightarrow$ Address? No transitive dependency here, as there is no other attribute, Z in the schema that will give SSN $\rightarrow$ Z and Z $\rightarrow$ Address
    - Then no need to check if determinant is superkey or not, or dependent is prime attribute or not
  - SSN $\rightarrow$ DeptNum? No transitive dependency here, as there is no other attribute, Z in the schema that will give SSN $\rightarrow$ Z and Z $\rightarrow$ EmpName
    - Then no need to check if determinant is superkey or not, or dependent is prime attribute or not

THIRD-NORMAL FORM

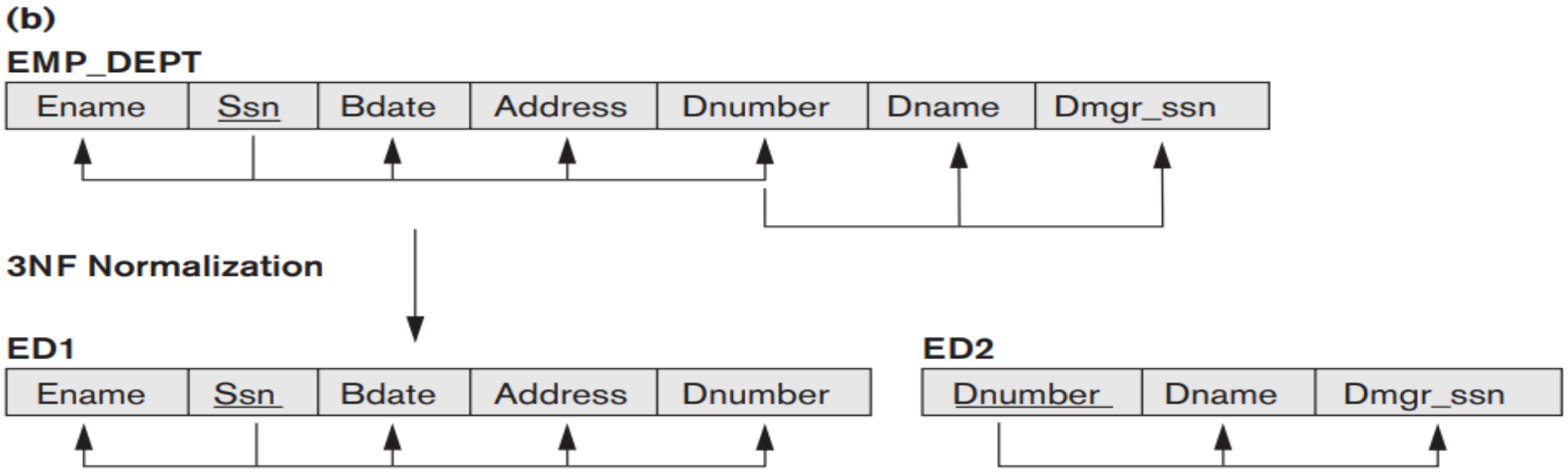
- **Solution**:
  - SSN $\rightarrow$ DeptName? There is transitive dependency here, as there is DeptNum attribute in the schema that will give SSN $\rightarrow$ DeptNum and DeptNum $\rightarrow$ DeptName
    - Then check if determinant is superkey or not?
      - For SSN $\rightarrow$ DeptNum? SSN is a primary key, so this condition holds then not violating the rules of 3NF
      - For DeptNum $\rightarrow$ DeptName? DeptNum is not a superkey, so check the dependent then. DeptName is not a prime attribute either, so this transitive dependency causes this schema to be **NOT in 3NF**

THIRD-NORMAL FORM

- **Solution**:
  - SSN $\rightarrow$ DeptManagerSSN? There is transitive dependency here, as there is DeptNum attribute in the schema that will give SSN $\rightarrow$ DeptNum and DeptNum $\rightarrow$ DeptManagerSSN
    - Then check if determinant is superkey or not?
      - For SSN $\rightarrow$ DeptNum? SSN is a primary key, so this condition holds then not violating the rules of 3NF
      - For DeptNum $\rightarrow$ DeptManagerSSN? DeptNum is not a superkey, so check the dependent then. DeptManagerSSN is not a prime attribute either, so this transitive dependency causes this schema to be **NOT in 3NF**

THIRD-NORMAL FORM

- **Solution**:
  - The resultant schemas are in 3NF:
    - ***ED1 (SSN, EmpName, Bdate, Address, DeptNum)***
    - ***ED2 (DeptNum, DeptName, DeptManagerSSN)***



GENERALIZATION OF NORMAL FORMS DEFINITIONS

- Here we give the general definitions of 2NF and 3NF that take all candidate keys of a relation into account.
- Notice that this does not affect the definition of 1NF since it is independent of keys and functional dependencies.
- We define **prime attribute**, as an attribute that is part of any candidate key.
- Partial and full functional dependencies and transitive dependencies for 2NF and 3NF will be presented with respect to all candidate keys.

NORMALIZATION

- Activity Sheet

NORMALIZATION

- Activity Sheet Solution
  - Normalization Solution