

Chapter 4

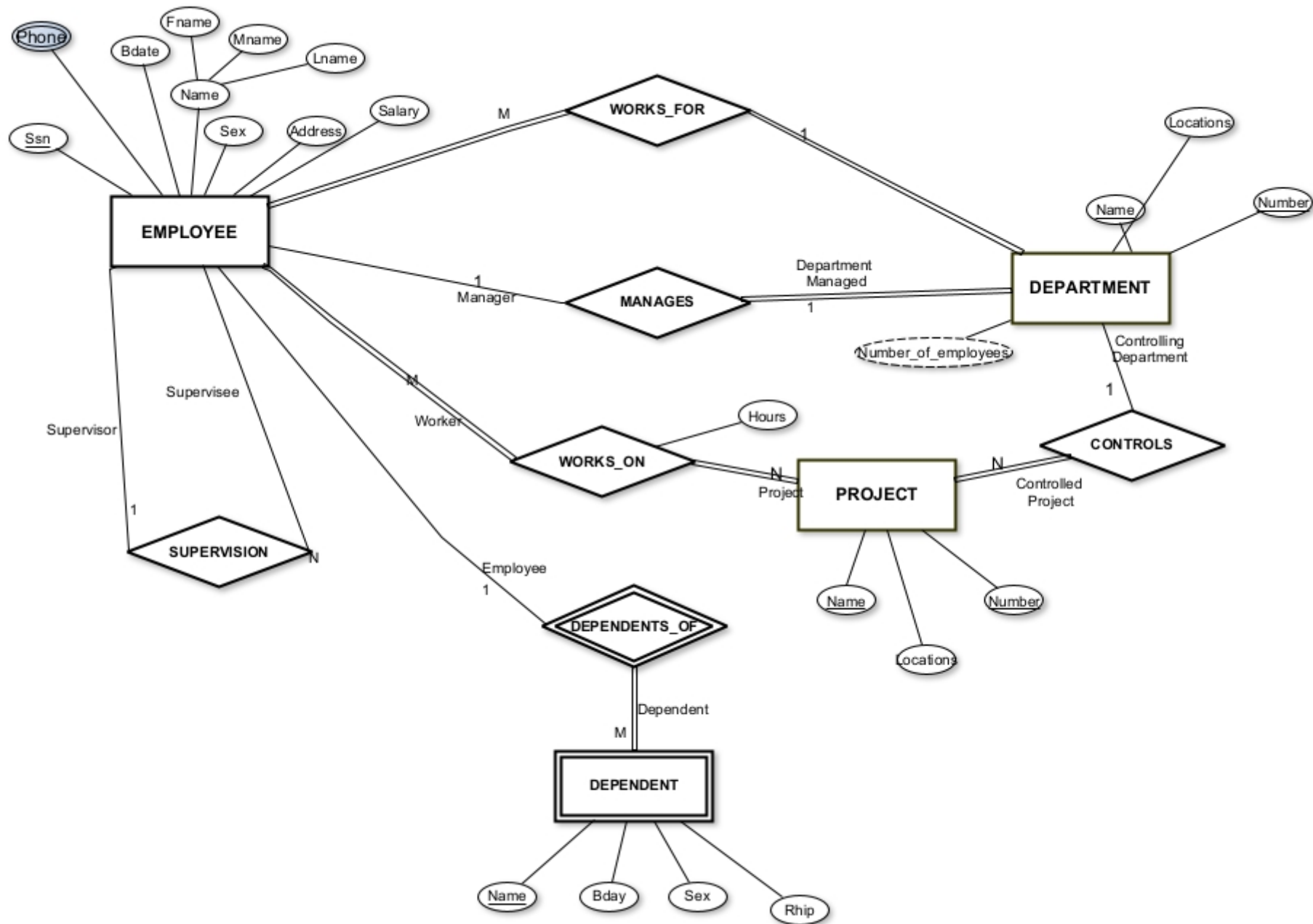
Data Normalization

Department of Health Informatics

Nov 2022

Steps in Data Normalization in RDBMS

- Step 1: Convert the ER diagram (conceptual schema) into logical schema (table)
- 3 basic Rules to convert conceptual schema into logical schema:
- Rule 1: Entity Names will automatically be table names
 - (e.g. the hypothetical COMPANY database will have 4 tables—EMPLOYEE, DEPARTMENT, PROJECT DEPENDENT)



Steps in Data Normalization...

- Rule 2: Mapping of attributes:
 - attributes will be columns of the respective tables.
 - Atomic or single-valued or derived or stored attributes will be columns
 - Composite attributes: the parent attribute will be dropped and the decomposed attributes (child attributes) will be columns of the table.
 - Multi-valued attributes will be mapped to a new table where the primary key of the main table will be posted for cross referencing.

Steps in Data Normalization...

Composite attr (Name)

EMPLOYEE

Ssn (PK)	Sex	Bdate	Fname	Mname	Lname	Address	Salary
-------------	-----	-------	-------	-------	-------	---------	--------

DEPARTMENT

Depname	Dlocations	DNumber (PK)	Noemployee
---------	------------	--------------	------------

PROJECT

Projname (PK)	Plocations	PnSumber (PK)
------------------	------------	---------------

DEPENDENT

Name (PK)	Bdate	sex	Rship
-----------	-------	-----	-------

TELEPHONE

Ssn (FK)	Phone
----------	-------

Associative table for multivalued attr of in EMPLOYEE entity set

Steps in Data Normalization in RDBMS

- Rule 3: Relationships are mapped by using a **foreign key** attribute.
 - FK is a primary or candidate key of one relation used to create association between tables.
 - For a relationship with 1:1 Cardinality post the PK or CK of one of the table into the other as a FK.
 - In cases where one entity is having **partial participation** on the relationship, it is recommended to post the CK of the partial participants to the total participant as a FK. For a relationship between EMPLOYEE and DEPARTMENT where employee MANAGES a department, the cardinality is one-to-one.
 - Here the PK of the EMPLOYEE can be posted to the DEPARTMENT or the PK of the Department can be posted to the EMPLOYEE.

Steps in Data Normalization in RDBMS

- But the EMPLOYEE is having partial participation on the relationship “MANAGES” as not all employees are managers of departments.
- Thus, even though both way is possible, it is recommended to post the PK of the EMPLOYEE to the DEPARTMENT table as a foreign key.

EMPLOYEE

Ssn (PK)	Sex	Bdate	Fname	Mname	Lname	Address	Salary	DNumber (FK)
----------	-----	-------	-------	-------	-------	---------	--------	--------------

DEPARTMENT

Depname	Dlocations	DNumber (PK)	Noemployee		
---------	------------	--------------	------------	--	--

MANAGES (1:1;
EMPLOYEE=partial partc, 1:M)

Steps in Data Normalization...

- For a relationship with 1:M cardinality, post the PK or CK from the —one side as a foreign key attribute to the —many side. E.g.: For a relationship called WORKS_FOR between EMPLOYEE (Many) and DEPARTMENT (One) the PK of the one side which is DEPARTMENT should be posted to the many side which is EMPLOYEE table.

EMPLOYEE

Ssn (PK)	Sex	Bdate	Fname	Mname	Lname	Address	Salary	Depname (FK)
-----------------	------------	--------------	--------------	--------------	--------------	----------------	---------------	---------------------

DEPARTMENT

Depname (PK)	Dlocations	DNumber (PK)	Noemployee	Ssn (FK)	
---------------------	-------------------	---------------------	-------------------	-----------------	--

WORKS_FOR (1:1;
EMPLOYEE=partial
partc. 1: M)

Steps in Data Normalization...

- For a relationship with M:N cardinality one has to create a new table (associative entity) and post PK from the participant entities as FK attributes in the new table along with some additional attributes (if applicable).

Steps in Data Normalization...

EMPLOYEE

Ssn (PK)	Sex	Bdate	Fname	Mname	Lname	Address	Salary
-------------	-----	-------	-------	-------	-------	---------	--------

PROJECT

Projname (PK)	Plocations	PnSumber (PK)
------------------	------------	---------------

ASSOCTAB

Ssn (PK)	Projname (FK)
----------	------------------

Steps in Data Normalization in RDBMS

- Now, the original four tables will become 6 tables (EMPLOYEE, DEPARTMENT, PROJECT, DEPENDENT+TELEPHONE (composite attr), and ASSOCTAB (M:N cardinality of PROJECT and EMPLOYEE))

Steps in Data Normalization in RDBMS

- To summarize the conversion of ER diagram into logical schema (table)
 - A) The entity types become table names
 - B) Atomic and derived attributes of each table become column names in each tables
 - C) Composite attributes- drop the parent attribute and put its child attributes as columns in the tables

Steps in Data Normalization in RDBMS

- To summarize the conversion of ER diagram into logical schema (table)...
- d) Multivalued attributes- create a new associate table that contains the multivalued attribute and the PK of the table that consists the multi-valued attribute. The PK now becomes a FK in the associate table
- e) To represent partial and full participating relationships, preferably, put the partial participating table PK as a FK in the fully participating table

Steps in Data Normalization in RDBMS

- To summarize the conversion of ER diagram into logical schema (table)...
- f) 1: M cardinality: put the 1-side table's PK as a FK in the M sides table
- g) M: N cardinality: create associate table that contains the PKs of the tables in the relationship. The PKs now become FKs in the associate table

Exercise

- Convert the COMPANY ER diagram into logical table. Show the PKs and FKs of related tables as PK and FK in parenthesis
- (note: there should be 6 tables)

Steps in Data Normalization...

TELEPHONE

EMPLOYEE				Ssn (FK)		Phone		
Ssn (PK)	Sex	Bdate	Fname	Mname	Lname	Address	Salary	Dnumber (FK)

DEPARTMENT				
Depname	Dlocations	DNumber (PK)	Noemployee	Ssn (FK)

Projname (PK)	Plocations	PnSumber (PK)
---------------	------------	---------------

DEPENDENT			
Name (PK)	Bdate	sex	Rship

ASSOCTAB	
Ssn (PK)	Projname (FK)

EMPLOYEE:
DEPARTMENT, 1:M,
MANAGES
relship

EMPLOYEE:
DEPARTMENT, M:1,
WORKS_FOR
R relship, full
participation
both

Steps in Data Normalization...

- Now, the normalization process starts
- First normal forms (1NF): a table is in 1NF if and only if:
 1. all the attributed have atomic values (a table should not have multi-valued and composite attributes)
 2. All values in a column have the same data types
 3. The columns have unique names
 4. No order (top-to-bottom or left-to-right) is applied

NOTE: If the ER diagram is correctly mapped into logical tables, it will be in 1NF.

Steps in Data Normalization...

- Now, the normalization process starts
- Second normal forms (2NF): a table is in 2NF if and only if:
 1. If it is in 1NF and
 2. **If all non-key attributes are dependent on the entire primary key. i.e. no partial dependency.** This implies that proper sub-sets of any candidate key (which are the prime attrs of the relationship) should not determine non-prime attributes

NB: Testing for 2NF is done only if the PK is composite. if the PK is simple (only one attribute), then a table that is in 1NF is by default in a 2NF

Steps in Data Normalization...

e.g., If the candidate Key is $\{ADE\}$ (which is a composite PK) in $R(A,B,C,D,E,F)$, then any subset of ADE should not determine the non-prime attributes in the relationship (which are B,C, F)

e.g.: $R(A,B,C,D,E,F)$ and FD: $\{A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow E\}$

Then is this relationship in 2NF?

Steps in Data Normalization...

- The candidate key is $\{A\} \rightarrow$ it is a simple PK. Therefore, the relation (table) is in 2NF given it is in 1NF
- For clarification:
 - If proper subsets of attribute A determine the any of the non-prime attrs (B,C,D,E or F), then no 2NF is met.
 - Since A has not proper subset, there is not partial dependency and therefore, the above relation in in 2NF

Steps in Data Normalization...

Third NF (3NF)

- A relation that is in First and Second Normal Form and in which no non-primary-key attribute is transitively dependent on the primary key, then it is in Third Normal Form (3NF).

Steps in Data Normalization...

Example of 3NF:

Assumption: Students of same batch live in one building or dormitory

StudID (PK)	Stud_F_Name	Stud_L_Name	Dept	Year	Dormitory
125/97	Abebe	Mekuria	Info Sc	1	401
654/95	Lemma	Alemu	Geog	3	403
842/95	Chane	Kebede	CompSc	3	403
165/97	Alem	Kebede	InfoSc	1	401
985/95	Almaz	Belay	Geog	3	403

Steps in Data Normalization...

- This schema is in its 1NF and 2NF since the primary key is a single attribute and there are no repeating groups (multi valued attributes).
- Let's take *StudID*, *Year* and *Dormitory* and see the dependencies.
- *StudID* → *Year* AND *Year* → *Dormitory*
- *And Year cannot determine StudID and Dormitory cannot determine StudID*
- *Then transitively StudID* → *Dormitory*
- *Therefore, Dormitory, the non-PK attr is transitively dependent on the PK (StudID). So, it is not in its 3NF*

Steps in Data Normalization...

- To convert it to a 3NF we need to remove all **transitive dependencies** of non key attributes on another non-key attribute.
- The non-primary key attributes, dependent on each other will be moved to another table and linked with the main table using Candidate Key- Foreign Key relationship.
- STUDENT DORM

StudID (PK)	Stud_F_Name	Stud_L_Name	StudID (FK)	Year	Dormitory
-------------	-------------	-------------	-------------	------	-----------

Steps in Data Normalization...

- Although there are more normalizations, a dataset is said to be normalized if it reaches 3NF
- A database with all tables in 3NF is said to be a normalized database

Steps in Data Normalization...

- with all tables in 3NF is said to be a normalized database
- Mnemonic for remembering the rationale of normalization up to 3NF, use the following:
 1. There should be no **repeating field** in a relation (table)
 2. The fields **depend upon the key**: the table should solely depend of the PK (candidate key)
 3. There should be no partial key dependency
 4. No inter-data dependency

Summary of NFs Based on PK and Corresponding Normalization

Normal Form	Test	Remedy (Normalization)
1NF	Relation (table) should have unique attribute names, same data type in the same attribute, no composite attributes	•Form new relations for each multivalued attribute. •Put each child attributes of a multivalued attributes as their own attributes

Summary....

Normal Form	Test	Remedy (Normalization)
2NF	When the PK is a simple attribute, and the table is already in 1NF, then no test is needed	No remedy
2NF	When the PK is composite, then there should not be partial dependency. i.e., the prime attributes should not determine the non-prime attributes. E.g. if {AB} is the PK R(A,B, C, D), Then, either A or B independently determine C or D	Decompose and set up a new relation for each partial key with its dependent attribute(s). Make sure to keep a relation with the original PK and any attributes that are fully functionally dependent on it.

Summary....

Normal Form	Test	Remedy (Normalization)
3NF	Relation should not have a nonkey attribute functionally determined by another nonkey attribute (or by a set of nonkey attributes). That is, there should be no transitive dependency of a nonkey attribute on the PK. E.g., if a table has four non-key columns (let's say, A,B,C,D) and a PK (day X), Then there should not any functional dependency between the nonkey attributes (A,B,C,D,)	Decompose and set up a relation that includes the nonkey attribute(s) that functionally determine(s) other nonkey attribute(s). E.g, if in the above table $C \rightarrow D$, then decompose the table into two. Table 1 will include the {PK, A, B}; table 2 will include the {PK, C, D}

Exercise

- Is the COMPANY database a normalized database?