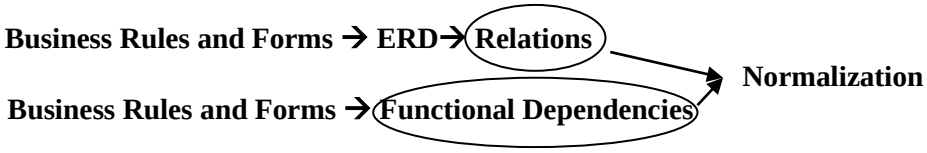
Top-Down Normalization Process:



Examples of Determining Functional Dependencies (FDs) using Business Rules (BRs):

BR1: The system will automatically generate a unique number for a wish list of a customer in order to trace all the wish lists. The system also stores the date that the wish list of the customer has been created.

FD1: custNo, ListNo→createDate

BR2: Within a wish list, the customer can add as many cameras as he or she likes. For each entry in a wish list, which is for one and only one camera, customer can make a brief note, for example, "this is my first choice".

FD2: custNo, ListNo, camearNo → note

BR3: A camera is described by a unique camera number, a camera name and a standard price.

FD3: camearNo→cameraName, standardPrice

Examples to Explain Top-Down Normalization:

After we convert ERD to relations, there is no guarantee that all relations are well structured and all anomalies are removed. Therefore we check the relations to make sure all anomalies are removed doing top-down normalization process (book-page 210).

1- **First normal form** (**1NF**):

- **First definition:**
  - All attributes are atomic: Attributes can't be divided into subparts are called Simple or Atomic attributes (they shouldn't be composite).
    - For instance in the following example:

      Enrollment *(StudentID, SubjectID, EnrollmentDate, StudentFirstName, StudentLastName, StudentAge, StudentGender, StudentDOB, StudentAddress, StudentPhone, SupervisorID, SupervisorFirstName, SupervisorLastName, SupervisorAddress, SupervisorPhone, Mark)*

      Address = (Unit Number, Street, Suburb, State) is non-atomic attribute. You can divide it to 4 atomic attributes: "Unit Number", "Street", "Suburb", and "State". So the correct sample should be:

      Enrollment *(StudentID, SubjectID, EnrollmentDate, StudentFirstName, StudentLastName, StudentAge, StudentGender, StudentDOB, StudentUnitNumber,*

StudentStreet, StudentSuburb, StudentState, StudentPhone, SupervisorID, SupervisorFirstName, SupervisorLastName, SupervisorUnitNumber, SupervisorStreet, SupervisorSuburb, SupervisorState, SupervisorPhone, Mark)

- There is no derived attribute
  - For instance "StudentAge" can be calculated using "StudentDOB", so it should be deleted.

Enrollment *(StudentID, SubjectID, EnrollmentDate, StudentFirstName, StudentLastName, StudentGender, StudentDOB, StudentUnitNumber, StudentStreet, StudentSuburb, StudentState, StudentPhone, SupervisorID, SupervisorFirstName, SupervisorLastName, SupervisorUnitNumber, SupervisorStreet, SupervisorSuburb, SupervisorState, SupervisorPhone, Mark)*

- **Second definition:**
  - There is no repeating group in the relation (there is no attribute in this relation that has more than one value for every value of PK. Or: All attributes have atomic value for every PK value)
    - For instance there are 2 values for Telephone Number for Customer ID= 456 in "Customer Telephone Number" table

Customer Telephone Number				
Customer ID	**Telephone ID**	**Telephone Type**	**Telephone State Code**	**Telephone Number**
123	1	Home	02	555-861-2025
456	2	Work	05	555-403-1659
456	3, 4	Home, Mobile	01, 02	555-776-4100, 555-776-4111
789	5	Mobile	04	555-808-9633

First you have to convert "Customer" table to the following table:

Customer Telephone Number				
Customer ID	**Telephone ID**	**Telephone Type**	**Telephone State Code**	**Telephone Number**
123	1	Home	02	555-861-2025
456	2	Work	05	555-403-1659
456	3	Home	01	555-776-4100
456	4	Mobile	02	555-776-4111
789	5	Mobile	04	555-808-9633

- A primary key has been defined that uniquely identifies each row.
  - Based on FD: Customer ID, Telephone ID → Telephone Number

    you have to consider "Customer ID" and "Telephone ID" together as PK.

2- Second normal form (2NF):

- **A relation is in second normal form (2NF) if:**

  - It is in 1st Normal Form.
  - All of its non-key attributes are dependent on all parts of the key (composite Key). It means there is no partial functional dependency.

    - For instance in the following sample:

      Enrollment *(StudentID, SubjectID, EnrollmentDate, StudentFirstName, StudentLastName, StudentGender, StudentDOB, StudentUnitNumber, StudentStreet, StudentSuburb, StudentState, StudentPhone, SupervisorID, SupervisorFirstName, SupervisorLastName, SupervisorUnitNumber, SupervisorStreet, SupervisorSuburb, SupervisorState, SupervisorPhone, Mark)*

      Related FDs are:

      StudentID → *StudentFirstName, StudentLastName, StudentGender, StudentDOB, StudentUnitNumber, StudentStreet, StudentSuburb, StudentState, StudentPhone, SupervisorID, SupervisorFirstName, SupervisorLastName, SupervisorUnitNumber, SupervisorStreet, SupervisorSuburb, SupervisorState, SupervisorPhone*

      StudentID, SubjectID, EnrollmentDate → *Mark*

      As we can see from FDs, StudentFirstName, StudentLastName, StudentGender, StudentDOB, StudentUnitNumber, StudentStreet, StudentSuburb, StudentState, StudentPhone, SupervisorID, SupervisorFirstName, SupervisorLastName, SupervisorUnitNumber, SupervisorStreet, SupervisorSuburb, SupervisorState and SupervisorPhone **are Just depended to** *"StudentID"*

      Just *Mark* **is depended to "**StudentID", "SubjectID" and "EnrollmentDate"

      To achieve 2Nf we have to put all attributes that are depended to one part of the composite key into the separate relation as follows:

      Student *(StudentID, StudentFirstName, StudentLastName, StudentGender, StudentDOB, StudentUnitNumber, StudentStreet, StudentSuburb, StudentState, StudentPhone, SupervisorID, SupervisorFirstName, SupervisorLastName, SupervisorUnitNumber, SupervisorStreet, SupervisorSuburb, SupervisorState, SupervisorPhone)*

      Enrollment *(StudentID, SubjectID, EnrollmentDate, Mark)*

- Another sample: for the following table that is in 1NF, and has “Customer ID” and “Telephone ID” as PK:

Customer Telephone Number				
Customer ID	**Telephone ID**	**Telephone Type**	**Telephone State Code**	**Telephone Number**
123	1	Home	02	555-861-2025
456	2	Work	05	555-403-1659
456	3	Home	01	555-776-4100
456	4	Mobile	02	555-776-4111
789	5	Mobile	04	555-808-9633

“Telephone Number” and “Telephone State Code” are depended to “Telephone ID”, not both “Customer ID” and “Telephone ID”. To achieve 2Nf, we have to split this table into two tables as follows:

Customer-Telephone		
Customer ID	**Telephone ID**	**Telephone Type**
123	1	Home
456	2	Work
456	3	Home
456	4	Mobile
789	5	Mobile

Telephone-Number		
Telephone ID	**Telephone State Code**	**Telephone Number**
1	02	555-861-2025
2	05	555-403-1659
3	01	555-776-4100
4	02	555-776-4111
5	04	555-808-9633

3- Third normal form (3NF):

- **A relation is in third normal form (3NF) if:**

  - It is in 2nd Normal Form.
  - There is no Transitive dependency in it. It means there is no non-key attribute in this relation that depends on another non-key

    - For instance in the following relation:

      Student *(StudentID, StudentFirstName, StudentLastName, StudentGender, StudentDOB, StudentUnitNumber, StudentStreet, StudentSuburb, StudentState, StudentPhone, SupervisorID, SupervisorFirstName, SupervisorLastName, SupervisorUnitNumber, SupervisorStreet, SupervisorSuburb, SupervisorState, SupervisorPhone)*

      The related Functional Dependencies (FDs) are:

      StudentID → *StudentFirstName, StudentLastName, StudentGender, StudentDOB, StudentUnitNumber, StudentStreet, StudentSuburb, StudentState, StudentPhone, SupervisorID, SupervisorFirstName, SupervisorLastName, SupervisorUnitNumber, SupervisorStreet, SupervisorSuburb, SupervisorState, SupervisorPhone*

      SupervisorID → *SupervisorFirstName, SupervisorLastName, SupervisorUnitNumber, SupervisorStreet, SupervisorSuburb, SupervisorState, SupervisorPhone*

      As we see from FDs, SupervisorFirstName, SupervisorLastName, SupervisorUnitNumber, SupervisorStreet, SupervisorSuburb, SupervisorState and SupervisorPhone are depended to "*SupervisorID*".

      To achieve 3Nf, all non-keys that are depended to "*SupervisorID*" should be removed from the first relation and create another relation. In addition, we have to keep a copy of "*SupervisorID*" in the first relation. It helps us to trace all data related to the Supervisor of the Student. Change the relation as follows:

      Student *(StudentID, StudentFirstName, StudentLastName, StudentGender, StudentDOB, StudentUnitNumber, StudentStreet, StudentSuburb, StudentState, StudentPhone, SupervisorID*)*

      Supervisor *(SupervisorID, SupervisorFirstName, SupervisorLastName, SupervisorUnitNumber, SupervisorStreet, SupervisorSuburb, SupervisorState, SupervisorPhone)*

- For instance in the following table:

Staff				
Staff-ID	**Staff-First-Name**	**Staff-Last-Name**	**Insurance-Company-ID**	**Insurance-Company-Name**
1	Johan	Walker	11	Allianz
2	Julia	Smith	20	Medibank
3	Jack	Wood	11	Allianz
4	Jenny	Brown	70	Sun Corp

"Insurance-Company-Name is depended to "Insurance-Company-ID", so you have to split this table into two tables as follows:

Staff			
Staff-ID	**Staff-First-Name**	**Staff-Last-Name**	**Insurance-Company-ID**
1	Johan	Walker	11
1	Johan	Walker	20
2	Jack	Wood	11
3	Jenny	Brown	70

Insurance-Company	
Insurance-Company-ID	**Insurance-Company-Name**
11	Allianz
20	Medibank
32	Sun Corp

As can be seen, "Insurance-Company-ID" is applied as a FK in the "Staff" table.

4- BCNF

- **A relation is in BCNF if:**
  - It is in 1nd Normal Form.
  - It is in 2nd Normal Form.
  - It is in 3nd Normal Form.
  - There are no FDs where the primary key of the relation is determined by non-key attributes of the same relation.

R(A,B,C,D,E)
FD1: A,B → C,D,E
FD2: D → A,B

A and B together are the (composite) primary key of R1. However, according to the FD2, they are both determined by D. This is a contradictory situation given the two FDs (Recall bill relation) Solution is to break R down into:

R1(A,B,C,E)
New born: R2(D,A,B)

Note: A Sample for your assignment when a relation is already in 3NF

Do not forget that you have to consider both "Relations" and "FDs" to do normalization.
For example:
Relation: DRUG (drugNo, drugDesc, drugDosage, drugMethod, drugName, drugUnitPrice)
FD: drugNo → drugDesc, drugDosage, drugMethod, drugName, drugUnitPrice
1NF – attributes are atomic and there is not derived attribute
2NF – no partial dependencies on the key drugNo (single attribute key)
3NF – no interdependencies between non-key attributes
Highest Normal Form = BCNF, determinant drugNo is the key

Note: Do not use “table sample” in your assignments, table samples in this document are just used for more clarification.