Normalization is a process for determining what attributes go into what tables. In order to understand this process, we need to know certain definitions:

**Functional Dependence:**
Attribute B is functionally dependent on Attribute A, (or group of attributes A) if for each value of A there is only be one possible value of B.
We say $A \rightarrow B$ (A decides B) or (B is functionally dependent on A).

Note: What we consider as a more important thing: parent and children? What is dependant of such a relation in Database?

Consider the following tables - and list the functional dependencies

<table>
<thead>
<tr>
<th>Table</th>
<th>Functional Dependence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student (StNum, SocSecNum, StName, age)</td>
<td>$StNum \rightarrow$ SocSecNum, StName, age</td>
</tr>
<tr>
<td>Emp (EmpNum, SocSecNum, Name, age, start-date, CurrSalary)</td>
<td>$EmpNum \rightarrow$ SocSecNum, Name, age, start-date, CurrSalary</td>
</tr>
<tr>
<td>Registration (StNum, CNum, grade)</td>
<td>$StNum \rightarrow$ CNum, grade</td>
</tr>
</tbody>
</table>

We can now define **primary key**:
- A single attribute (or minimal group of attributes) that functionally determine all other attributes.
- If more than one attribute meets this condition, they are considered **candidate keys**, and one is chosen as the primary key.

the candidate keys

- for the Student table are
- for the Emp table
- for the registration table

For the tables with more than one candidate key, what do you think should be chosen for the primary key?
Determining functional dependencies

Consider the following table  \( \text{Emp}(\text{Enum}, \text{Ename}, \text{age}, \text{acctNum}, \text{AcctName},) \)
what are the functional dependencies? Can we be sure?

In order to determine the functional dependencies we need to know about the “real world model” the database is based on.
For example: does each employee work on each single account or can there be multiple accounts assigned? Can an account have more than one employee assigned to it? …
Suppose we know:  
  a) each Account is assigned to one employee,  
  b) an employee can work on more than one Account

Now we can determine the functional dependencies. They are:

Suppose we have this information for a new housing development
\( \text{Houses}(\text{LotNum}, \text{Address}, \text{builder}, \text{model}, \text{upGradeNum}, \text{upGradeName}, \text{squareFeet}) \)

What are the functional dependencies? We can’t determine them, unless we know more about the “real world situation”

Suppose we know:
1) Each builder has a list of models they offer, but each model is unique to a specific builder. For example:
   Builder A may offer 3 models: the Ashley, the Brigham, and the Cambridge
   Builder B may offer 2 models: the Axton and the Braxton
2) Each model always has the same amount of square feet
3) Each specific house comes with a list of upgrades
   (for example  
   house with LotNum 101 might have \( \text{upNum} 652 \) granite countertops \( \text{upNum} 639 \) hardwood floors \( \text{upNum} 622 \) finished basement
   house with LotNum 122 might have \( \text{upNum} 639 \) hardwood floors \( \text{upNum} 622 \) finished basement

Now what are the functional dependencies?
The process of normalization involves converting your tables to 1st then 2nd then 3rd Normal forms.

<table>
<thead>
<tr>
<th>1st Normal form</th>
</tr>
</thead>
<tbody>
<tr>
<td>According to Date's definition of 1NF, a table is in 1NF if and only if it is &quot;isomorphic to some relation&quot;, which means, specifically, that it satisfies the following five conditions:</td>
</tr>
<tr>
<td>1. There's no top-to-bottom ordering to the rows.</td>
</tr>
<tr>
<td>2. There's no left-to-right ordering to the columns.</td>
</tr>
<tr>
<td>3. There are no duplicate rows.</td>
</tr>
<tr>
<td>4. Every row-and-column intersection contains exactly one value from the applicable domain (and nothing else).</td>
</tr>
<tr>
<td>5. All columns are regular [i.e. rows have no hidden components such as row IDs, object IDs, or hidden timestamps].</td>
</tr>
</tbody>
</table>

Violation of any of these conditions would mean that the table is not strictly relational, and therefore that it is not in 1NF.

Simply, a table is in first normal form if there is no any duplicate, also called multiple values.

### Customer

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>First Name</th>
<th>Surname</th>
<th>Telephone Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>Robert</td>
<td>Ingram</td>
<td>555-861-2025</td>
</tr>
<tr>
<td>456</td>
<td>Jane</td>
<td>Wright</td>
<td>555-403-1659</td>
</tr>
<tr>
<td>789</td>
<td>Maria</td>
<td>Fernandez</td>
<td>555-808-9633</td>
</tr>
</tbody>
</table>

Repeating groups across rows (making multiple records!)

This will violate the policy of keys!

Repeating groups across columns

The designer might attempt to get around this restriction by defining multiple Telephone Number columns:

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>First Name</th>
<th>Surname</th>
<th>Tel. No. 1</th>
<th>Tel. No. 2</th>
<th>Tel. No. 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>Robert</td>
<td>Ingram</td>
<td>555-861-2025</td>
<td></td>
<td></td>
</tr>
<tr>
<td>456</td>
<td>Jane</td>
<td>Wright</td>
<td>555-403-1659</td>
<td>555-776-4100</td>
<td>555-403-1659</td>
</tr>
<tr>
<td>789</td>
<td>Maria</td>
<td>Fernandez</td>
<td>555-808-9633</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Difficulty in querying the table. Answering such questions as "Which customers have telephone number \( x \)?" and "Which pairs of customers share a telephone number?" is awkward.
- Inability to enforce uniqueness of Customer-to-Telephone Number links through the RDBMS. Customer 789 might mistakenly be given a Tel. No. 2 value that is exactly the same as her Tel. No. 1 value.
• Restriction of the number of telephone numbers per customer to three. If a customer with four telephone numbers comes along, we are constrained to record only three and leave the fourth unrecorded. This means that the database design is imposing constraints on the business process, rather than (as should ideally be the case) vice-versa.

Repeating groups within a composite value

The designer might, alternatively, retain the single Telephone Number column but alter its domain, making it a string of sufficient length to accommodate multiple telephone numbers:

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>First Name</th>
<th>Surname</th>
<th>Telephone Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>Robert</td>
<td>Ingram</td>
<td>555-861-2025</td>
</tr>
<tr>
<td>456</td>
<td>Jane</td>
<td>Wright</td>
<td>555-403-1659, 555-776-4100</td>
</tr>
<tr>
<td>789</td>
<td>Maria</td>
<td>Fernandez</td>
<td>555-808-9633</td>
</tr>
</tbody>
</table>

The Telephone Number heading becomes semantically woolly, as it can now represent either a telephone number, a list of telephone numbers, or indeed anything at all. A query such as "Which pairs of customers share a telephone number?" is more difficult to formulate, given the necessity to cater for lists of telephone numbers as well as individual telephone numbers.

To ensure/create a first normal form:

a) determine the primary key - (without considering the repetitions)
b) **Remove** each repeating group from the original table to a new one. Then, copy its relevant key (either the entire key or partial one, or even single key column) in this new table.
c) **Switch** the key role and make the repeating groups the key in new table.

Thus, the **Customer table** would be broken up into two tables. They would be

<table>
<thead>
<tr>
<th>Customer Name()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Phone()</td>
</tr>
</tbody>
</table>

a) what is the primary key of each table?

b) what is the relationship between the two tables? Which tables have **foreign keys** pointing to the primary key of another table (a foreign key is an attribute in one table after switching which matches (points to) the primary key in another table.)
For each of the following tables convert to a group of tables in first normal form
Show the primary key of each table
Show the foreign key of each table (and what table it points to)
Sketch the relationships between the tables

Student(StNum, StName, SocSecNum, age, clubs, awards)

Faculty(FacNum, SocSecNum, degree, rank, committees, papers-written)
2NF was originally defined by E.F. Codd in 1971.[1] A table that is in first normal form (1NF) must meet additional criteria if it is to qualify for second normal form. Specifically: a 1NF table is in 2NF if and only if, given any candidate key K and any attribute A that is not a constituent of a candidate key, A depends upon the whole of K rather than just a part of it.

A 1NF table is in 2NF if and only if all its non-prime attributes are functionally dependent on the whole of every candidate key. In slightly more formal terms: a 1NF table is in 2NF if and only if all its non-prime attributes are functionally dependent on the whole of every candidate key.

Note that when a 1NF table has no composite key (candidate keys consisting of more than one attribute), the table is automatically in 2NF.

Consider the following table - Where each project has many employees and each employee works on many projects. (some sample data is shown)

<table>
<thead>
<tr>
<th>ProjectInfo(PrjNum, PrjName, budget, EmpNum, EmpName, HrsWorked)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P22   Cyclone   50000  E1001   Joe     12</td>
</tr>
<tr>
<td>P22   Cyclone   50000  E2002   Pat     50</td>
</tr>
<tr>
<td>P21   IMB       20000  E3003   Ed      40</td>
</tr>
<tr>
<td>P21   IMB       20000  E2002   Pat     30</td>
</tr>
<tr>
<td>P21   IMB       20000  E1001   Joe     70</td>
</tr>
</tbody>
</table>

a)What problems do you see keeping all this data in one table?

a)what are the functional dependencies?

b)what is the key? (minimum group of attributes that determine all other attributes)
c) what are the partial dependencies? Is it in 2\textsuperscript{nd} N.F.?

Since there are partial dependencies this table is NOT in 2\textsuperscript{nd} N.F.
To convert to 2\textsuperscript{nd} N.F

a) **Remove** each partial dependency out of the original table to a new table: Copy the partial relationship to a new table, with its relevant key columns. Then, remove that non-key column from the original table.

b) Leave only the attributes that depend on the WHOLE key in the original table.

In the above example, we would **create** 2 new tables, from the partial dependencies

\begin{align*}
\text{Project}(\text{PrjNum},) & \quad \text{Emp}(\text{EmpNum})
\end{align*}

And the original table would include the key and all attributes dependent on the whole key

\begin{align*}
\text{Assignments}(\text{PrjNum}, \text{EmpNum},)
\end{align*}

c) What are the primary keys for each of the tables listed above
What are the relationships between the tables? What are the foreign keys?

---

**Normalization Exercise 2:** Consider the following table- With some sample data shown (assume each order can contain various numbers of different products, but is placed by one customer on one date)

\begin{verbatim}
Order (OrdNum, date, CustNum, prodNum, num-ordered, unit-price, total-price)
 O23  1/1/08  C22    P21      5    7.00 35.00
 O23  1/1/08  C22    P25      3    21.00 63.00
 O24  1/1/08  C22    P23      3    7.00 21.00
 O25  2/1/08  C44    P21     11    7.00 77.00
 O25  2/1/08  C44    P28      4    99.00 396.00
 O25  2/1/08  C44    P25      1    21.00 21.00
\end{verbatim}
1) determine the functional dependencies?
2) determine the primary key?
3) determine if there are any partial dependencies?
4) convert to 2nd N.F. Show all keys (primary and foreign)

Exercise 3: Consider the following table involving course info and the grades given in those courses

Courses (Cnum, Cname, credits, stuNum, stName, stAge, semester, grade)

a) List the functional dependencies

b) Determine the primary key

c) List any partial dependencies

d) Convert to 2nd N.F. Show all keys (primary and foreign) and sketch the relationships between the tables

Exercise 4:
Given the follow table T (A, B, C, D, E, F, G)

Suppose we have the following dependencies: A, B --> C; A, B --> F; A --> D; A --> E; B --> G

a) What is the primary key? (the minimum attributes that determine all the other attributes)

b) What are the partial dependencies?

c) Convert to set of tables in 2nd N.F. Indicate primary and foreign keys sketch the relationship between the tables
**3rd Normal Form**

The **third normal form (3NF)** is a normal form used in database normalization. 3NF was originally defined by E.F. Codd in 1971.[1] Codd's definition states that a table is in 3NF if and only if both of the following conditions hold:

- The relation R (table) is in **second normal form (2NF)**
- Every non-prime attribute of R is non-transitively dependent (i.e. directly dependent) on every candidate key of R.

A non-prime attribute of R is an attribute that does not belong to any candidate key of R. A transitive dependency is a functional dependency in which \( X \rightarrow Z \) (\( X \) determines \( Z \)) indirectly, by virtue of \( X \rightarrow Y \) and \( Y \rightarrow Z \) (where it is not the case that \( Y \rightarrow X \)).[2]

Simply saying, a table is in 3rd N.F if it is in 2nd N.F and the only dependencies involve the candidate keys.

Consider the following table: \( \text{Student} \ (\text{SocSecNum, StNum, name, age}) \)

what are the dependencies?

Since the only dependencies involved candidate keys then the Student table is in 3rd N.F

Consider the following table: \( \text{Emp} \ (\text{empNum, SocSecNum, age, deptNum, deptName}) \)

( where each employee works in one department)

what are the dependencies? what is the primary key?

Is it in 2nd N.form? explain

Is it in 3rd N. Form? explain

What problems do you see using the employee table above?

Thus even tables in 2nd Normal form create problems- so we must carry the normalization process further

Here is an example of a table that violates 3rd N.F \( \text{T} (A, B, C, D, E, F, G) \)

if \( A \) is the primary key then we know \( A \rightarrow B, C, D, E, F, G \)

if there are other dependencies such as \( D \rightarrow E, F \) \( C \rightarrow G \)

that don’t involve the key then the table is not in 3rd N.F

To convert to 3rd N.F
For each non-key dependency (a dependency that doesn’t involve a candidate key)

1) **Copy** this partial dependency in the original table to a new table, i.e., the attribute(s) that depend on something other than the primary key.

Using table T above we would have:

*Copy the dependencies* $D \rightarrow E,F$  $C \rightarrow G$ *to two new tables.*

$T_1(D,E,F)$  $T_2(C,G)$

we now have 3 tables -with the following primary and foreign keys

$T(A,B,C,D,E,F)$  $T_1(D,E,F)$  $T_2(C,G)$

fkey $D \rightarrow T_1$

fkey $C \rightarrow T_2$

In the above example: $Emp(empNum, SocSecNum, age, deptNum, deptName)$

the candidate keys are ________________

and the dependency ________________ doesn’t involve these keys

We create a table from the dependency that doesn’t involve the candidate keys

$Dept(______________)$

What are the keys of the Dept and Emp tables above (primary and foreign)?

What is the relationship between these two tables?

---

**Exercise 5:** Consider the following table (with some possible data)

(each order goes to a single customer and shipped from one warehouse)

<table>
<thead>
<tr>
<th>Order(ordNum, date, warehouseNum, warehouseLoc, custNum, custName)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Or55 1/1/07 w5 NY C55 Acme</td>
</tr>
<tr>
<td>Or66 1/1/07 w3 WC C66 IBM</td>
</tr>
<tr>
<td>Or77 4/4/07 w5 NY C77 Intel</td>
</tr>
<tr>
<td>Or88 4/12/07 w3 WC C55 Acme</td>
</tr>
</tbody>
</table>

We can see that this is not a “good table” because it is about more than one “entity”

It will also have redundant information about customers and warehouses.

a) What are the functional dependencies of the order table? What is the primary key?
b) why must it be in 2nd Normal Form?

c) Is it in 3rd Normal Form? Explain (any dependencies not involving candidate keys?)

d) Convert to 3rd N.Form. Indicate primary and foreign keys.
Indicate the relationships among these tables?

Exercise 6: Consider the following table with the given dependencies.
\[ T(A, B, C, D, E, F, G) \quad E \rightarrow G \quad B \rightarrow C, A \quad D \rightarrow A, B, C, E, F, G \]
a) What is the primary key? ______________________
b) Why must it be in 2nd N.Form? (why can't there be any partial dependencies)

c) Is it in 3rd N.Form? explain (any dependencies involving attributes that are not candidate keys?)

d) Convert to a set of tables in 3rd N.Form. Indicate primary and foreign keys.
Indicate the relationships among these tables?

Exercise 7, Consider the following table and dependencies
\[ T(A, B, C, D, E, F, G, H) \quad D+E \rightarrow A, B, C \quad D \rightarrow F \quad E \rightarrow G, H \quad H \rightarrow G \]
a) what is the key?____________________
b) are there any partial dependencies? List them?
c) Is T1 2\textsuperscript{nd} Normal form? Convert to a group of tables in 2\textsuperscript{nd} N.F


d) Are the tables in part C in 3\textsuperscript{rd} N.Form? Explain


e) Convert to 3\textsuperscript{rd} N. Form.
For each of your final tables indicate the primary key of each table and any foreign keys
Indicate what table each foreign key refers to. Show the relationships between these tables.


Exercise 8
Consider the table involving a chain of bookstores, books and publishers
(BranchNum, BranchAddr, BkNum, Tittle, PubNum, PubName, List-Price, InStock, list-Price, Selling-Price)

Suppose we know the following
1) each branch can sell a book at whatever Selling-price they want
2) each book has a set list price
3) each book is published by a single publisher

a) what are the functional dependencies?


b) if we keep everything in this one table what is the primary key? _______________
(remember the key is the minimum group of attributes that determine all other attributes)


c) Is the table in 2\textsuperscript{nd} N.F.? __________
(remember if we have any partial dependencies it is not in 2\textsuperscript{nd} N.F)
List any dependencies which involve part of the primary key (partial dependencies)?


d) Convert to a set of tables in 2\textsuperscript{nd} N.F. (show the primary key of each table)
(remember we do this by creating new tables from each partial dependency)
e) Is the table in 3\textsuperscript{rd} N.F? _______

(3\textsuperscript{rd} N.F: must be in 2\textsuperscript{nd} N.F and have all dependencies must involve candidate keys)

List any dependencies that don't involve the candidate keys

f) Convert to a set of tables in 3\textsuperscript{rd} N.F. Show all primary and foreign keys

For each foreign key indicate what table it points to.