

Functional Dependency

Formal Definition: Attribute **B** is functionally dependant upon attribute **A** (*or a collection of attributes*) if a value of **A** determines a single value of attribute **B** at any one time.

Formal Notation: $A \rightarrow B$ This should be read as ‘A determines B’ or ‘B is functionally dependant on A’. **A** is called the *determinant* and **B** is called the *object of the determinant*.

Example:

staffNo	job	dept	dname
SL10	Salesman	10	Sales
SA51	Manager	20	Accounts
DS40	Clerk	20	Accounts
OS45	Clerk	30	Operations

Functional Dependencies

$\text{staffNo} \rightarrow \text{job}$

$\text{staffNo} \rightarrow \text{dept}$

$\text{staffNo} \rightarrow \text{dname}$

$\text{dept} \rightarrow \text{dname}$

Partial Functional Dependency

Partial Functional Dependency: This is the situation that exists if it is necessary to only use a subset of the attributes of the composite determinant to identify its object uniquely.

Example:

Student#	Crs#	CrsName	grade
----------	------	---------	-------

Full Functional Dependencies
 $(\text{Student\#}, \text{Crs\#}) \rightarrow \text{grade}$

Partial Functional Dependencies
 $\text{Crs\#} \rightarrow \text{CrsName}$

Full Functional Dependency

Full Functional Dependency: Only of relevance with composite determinants. This is the situation when it is necessary to use all the attributes of the composite determinant to identify its object uniquely.

Example:

order#	Pd#	qty	price
A001	001	10	200
A002	001	20	400
A002	002	20	800
A004	001	15	300

Full Functional Dependencies

$(\text{Order\#}, \text{Pd\#}) \rightarrow \text{qty}$

$(\text{Order\#}, \text{Pd\#}) \rightarrow \text{price}$

Unnormalised Normal Form (UNF)

ORDER

Customer No: 001964
Name: Mark Campbell
Address: 1 The House
Leytonstone
E11 9ZZ

Order Number: 00012345
Order Date: 14-Feb-2002

Product Number	Product Description	Unit Price	Order Quantity	Line Total
T5060	Hook	5.00	5	25.00
PT42	Bolt	2.50	10	20.50
QZE48	Spanner	20.00	1	20.00

Order Total: 65.50

ORDER (order-no, order-date, cust-no, cust-name, cust-add,
(*prod-no, prod-desc, unit-price, ord-qty, line-total*)*, order-total

First Normal Form (1NF)

Definition: A relation is in 1NF if, and only if, all its underlying attributes contain atomic values only.

Remove repeating groups into a new relation

A repeating group is shown by a pair of brackets within the relational schema.

ORDER (order-no, order-date, cust-no, cust-name, cust-add,
*(prod-no, prod-desc, unit-price, ord-qty, line-total)**, order-total

Steps from UNF to 1NF:

- Remove the outermost repeating group (and any nested repeated groups it may contain) and create a new relation to contain it.
- Add to this relation a copy of the PK of the relation immediately enclosing it.
- Name the new entity (*appending the number 1 to indicate 1NF*)
- Determine the PK of the new entity
- Repeat steps until no more repeating groups.

Example - UNF to 1NF


ORDER (order-no, order-date, cust-no, cust-name, cust-add,
(*prod-no, prod-desc, unit-price, ord-qty, line-total*)*, order-total

1. Remove the outermost repeating group (and any nested repeated groups it may contain) and create a new relation to contain it. (*rename original to indicate 1NF*)

ORDER-1 (order-no, order-date, cust-no, cust-name, cust-add, order-total
(prod-no, prod-desc, unit-price, ord-qty, line-total))

2. Add to this relation a copy of the PK of the relation immediately enclosing it.

ORDER-1 (order-no, order-date, cust-no, cust-name, cust-add, order-total
(order-no, prod-no, prod-desc, unit-price, ord-qty, line-total))



3. Name the new entity (*appending the number 1 to indicate 1NF*)

ORDER-LINE-1 (order-no, prod-no, prod-desc, unit-price, ord-qty, line-total)

4. Determine the PK of the new entity

ORDER-LINE-1 (order-no, prod-no, prod-desc, unit-price, ord-qty, line-total)

Second Normal Form (2NF)

Definition: A relation is in 2NF if, and only if, it is in 1NF and every non-key attribute is fully dependent on the primary key.

Remove partial functional dependencies into a new relation

Steps from 1NF to 2NF:

- Remove the offending attributes that are only partially functionally dependent on the composite key, and place them in a new relation.
- Add to this relation a copy of the attribute(s) which are the determinants of these offending attributes. These will automatically become the primary key of this new relation.
- Name the new entity (*appending the number 2 to indicate 2NF*)
- Rename the original entity (*ending with a 2 to indicate 2NF*)

Example - 1NF to 2NF

ORDER-LINE-1 (order-no, prod-no, prod-desc, unit-price, ord-qty, line-total)

1. Remove the offending attributes that are only partially functionally dependent on the composite key, and place them in a new relation.


ORDER-LINE-1 (order-no, prod-no, ord-qty, line-total)

(prod-desc, unit-price)

2. Add to this relation a copy of the attribute(s) which determines these offending attributes. These will automatically become the primary key of this new relation..

ORDER-LINE-1 (order-no, prod-no, ord-qty, line-total)

(prod-no, prod-desc, unit-price)



3. Name the new entity (*appending the number 2 to indicate 2NF*)

PRODUCT-2 (prod-no, prod-desc, unit-price)

4. Rename the original entity (*ending with a 2 to indicate 2NF*)

ORDER-LINE-2 (order-no, prod-no, ord-qty, line-total)

Third Normal Form (3NF)

Definition: A relation is in 3NF if, and only if, it is in 2NF and every non-key attribute is non-transitively dependent on the primary key.

Remove transitive dependencies into a new relation

Steps from 2NF to 3NF:

- Remove the offending attributes that are transitively dependent on non-key attribute(s), and place them in a new relation.
- Add to this relation a copy of the attribute(s) which are the determinants of these offending attributes. These will automatically become the primary key of this new relation.
- Name the new entity (*appending the number 3 to indicate 3NF*)
- Rename the original entity (*ending with a 3 to indicate 3NF*)

Example - 2NF to 3NF

ORDER-2 (order-no, order-date, cust-no, cust-name, cust-add, order-total)

1. Remove the offending attributes that are transitively dependent on non-key attributes, and place them in a new relation.


ORDER-2 (order-no, order-date, cust-no, order-total)

(cust-name, cust-add)

2. Add to this relation a copy of the attribute(s) which determines these offending attributes. These will automatically become the primary key of this new relation..

ORDER-2 (order-no, order-date, cust-no, order-total)

(cust-no, cust-name, cust-add)



3. Name the new entity (*appending the number 3 to indicate 3NF*)

CUSTOMER-3 (cust-no, cust-name, cust-add)

4. Rename the original entity (*ending with a 3 to indicate 3NF*)

ORDER-3 (order-no, order-date, cust-no, order-total)

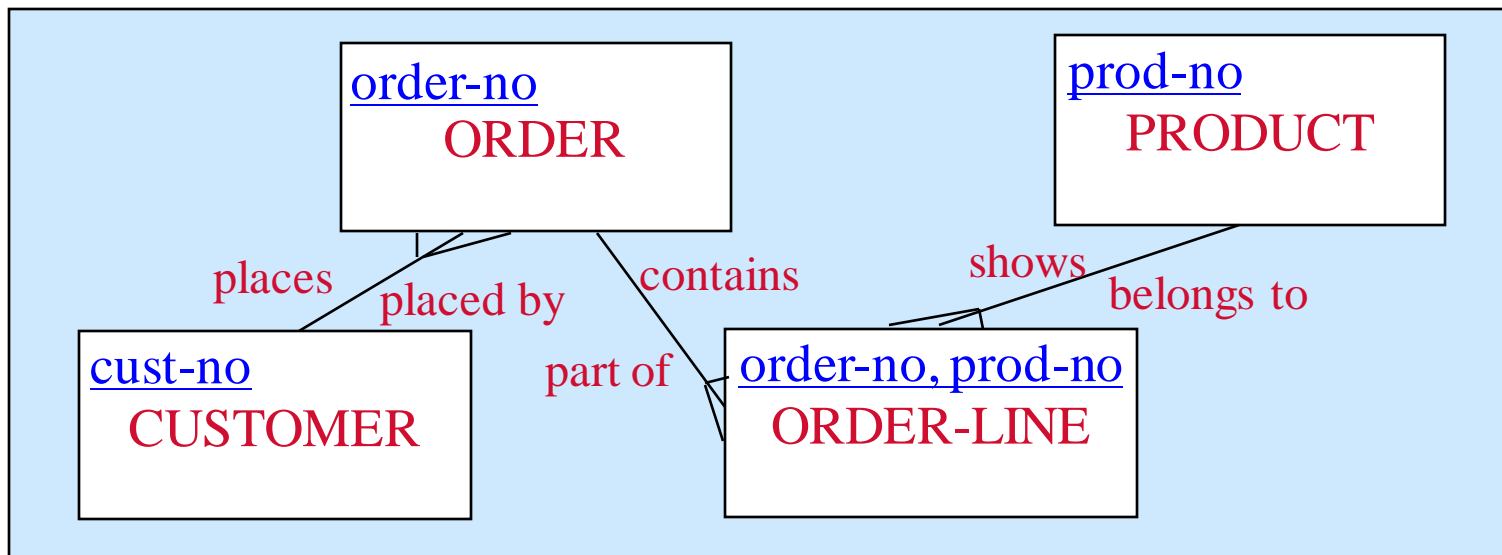
Example - Relations in 3NF

ORDER-3 (order-no, order-date, cust-no, order-total)

CUSTOMER-3 (cust-no, cust-name, cust-add)

PRODUCT-2 (prod-no, prod-desc, unit-price)

ORDER-LINE-2 (order-no, prod-no, ord-qty, line-total)



Stages of Normalisation

