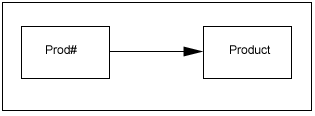
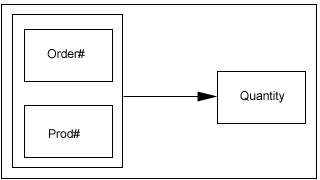
**3.6 FUNCTIONAL DEPENDENCY DIAGRAMS**

A set of Functional Dependencies for a data model can be documented in a **Functional Dependency Diagram** (also known as a **Determinancy Diagram**).

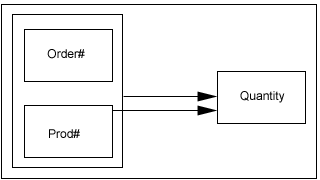
In a**Functional Dependency Diagram**each attribute is shown in a rectangle with an arrow indicating the direction of the dependency. The figure below illustrates the functional dependency **Prod# > Product**.



A Functional Dependency with Multiple Attributes is shown below, for the functional dependency **Order#, Prod# > Quantity**.

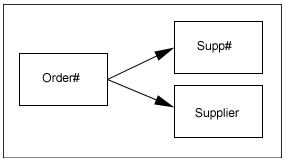


A derived Functional Dependency involving **Partial Key Dependency** is shown in the figure below.  
The arrow connected to the outer rectangle, which represents **Order#, Prod# > Product** can be deleted without loss of information.



A derived Functional Dependency involving ***Transitive Dependency*** is shown in the figure below.

The arrow which represents **Order# > Supplier** can be deleted without loss of information.

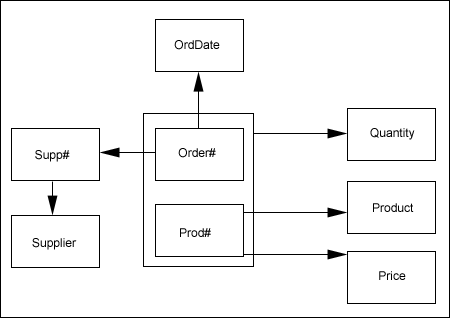


**Rules for Functional Dependency Diagrams**

The following rules apply to Functional Dependency Diagrams:

* each attribute appears only once on the Functional Dependency Diagram
* all the attributes of interest appear on the Functional Dependency Diagram
* no partial key dependencies appear on the Functional Dependency Diagram
* no transitive dependencies appear on the Functional Dependency Diagram

The complete Functional Dependency Diagram for the Purchase Order data model is shown below:



**Notice the way you read this FD above to create the tables from it – Read all table info starting from a PK field that has an arrow coming out of it, PK fields are marked with a pound sign #.**

**So Order# alone points to Orddate and Supp# - this means there will be a table with just Order#PK Orddate and Supp# in it and Supp# would be a FK**

**We also have the Supp# as a PK pointing to Supplier (which here is supplier name) and Supp# does not point to any other attributes so there would be a table with just Supp# and Supplier in it.**

**We have an outerbox around order# and prod# and they together point only to Quantity. This means order# and prod# together form a composite PK and they would exist with Quantity in one table.**

**We also have prod# alone pointing to Product and Price – so all three of these would be in one table**

[Continue to 2NF and Functional dependency](https://www.dlsweb.rmit.edu.au/Toolbox/ecommerce/dad_respak/dad_tutorial/html/dad_db3g_tut.htm)

**3.7 2NF AND FUNCTIONAL DEPENDENCY**

The requirements for 2NF can be restated as

* all non-key attributes must be ***Functionally Dependent*** on the whole of the Primary Key i.e. no ***partial key dependencies*** exist in the relation.

A **1NF** relation can be converted to a **2NF** by removing partial key dependencies from the relation.

The **1NF** relation  
**ORDERLINES1** ( ***Order#***, Prod#, Product, Price, Quantity )  
contains the Functional Dependencies

|  |  |  |
| --- | --- | --- |
| **Order#, Prod#** | **>** | **Product** |
| **Prod#** | **>** | **Product** |
| **Order#, Prod#** | **>** | **Price** |
| **Prod#** | **>** | **Price** |
| **Order#, Prod#** | **>** | **Quantity** |

which includes ***partial key dependencies***.

The partial key dependencies can be removed from  
**ORDERLINES1** ( ***Order#***, Prod#, Product, Price, Quantity )  
by decomposing it to the **2NF** relations  
**ORDERLINES2** ( ***Order#***, *Prod#*, Quantity )  
and  
**PRODUCTS2** ( Prod#, Product, Price ).

The **2NF** relation  
**ORDERLINES2** ( ***Order#***, ***Prod#***, Quantity )  
contains the Functional Dependency  
**Order#, Prod# > Quantity**  
which is not a ***partial key dependency***.

The **2NF** relation  
**PRODUCTS2** ( Prod#, Product, Price )  
contains the Functional Dependencies  
**Prod# > Product  
Prod# > Price**  
which do not include ***partial key dependencies***.

[Continue to 3NF - Third Normal Form](https://www.dlsweb.rmit.edu.au/Toolbox/ecommerce/dad_respak/dad_tutorial/html/dad_db3h_tut.htm)

**3.8 3NF - THIRD NORMAL FORM**

A relation is in **3NF** if

* it is in **2NF**
* no ***non-key attribute*** is **Functionally Dependent** on any other ***non-key attribute*** (i.e. no transitive dependencies exist in the relation).

This requirement means that each **2NF** relation with a single non-key attribute is automatically in **3NF**.  
For the Purchase Order data model, the following **2NF** relation is in **3NF**:  
**ORDERLINES2** ( ***Order#***, ***Prod#***, Quantity )

The other **2NF** relations with multiple non-key attributes must be examined to determine if any non-key attribute is identified uniquely by one or more of the other non-key attributes which make up the relation.

The **2NF** relation  
**ORDERS2**( Order#, OrdDate, Supp#, Supplier )  
contains the Functional Dependencies  
**Order# > OrdDate  
Order# > Supp#  
Order# > Supplier  
Supp# > Supplier**  
which includes the transitive dependency  
**Order# > Supp# > Supplier**

The transitive dependency can be removed from  
**ORDERS2**( Order#, OrdDate, Supp#, Supplier )  
by decomposing it to the **3NF** relations  
**ORDERS3**( Order#, OrdDate, Supp# )  
and  
**SUPPLIERS3**( Supp#, Supplier ).

The **2NF** relation  
**PRODUCTS2** ( Prod#, Product, Price ).  
contains the Functional Dependencies  
**Prod# > Product  
Prod# > Price**  
but NOT  
**Product > Price**  
so it is in **3NF**.

The complete set of **3NF** relations for the Purchase Order data model is:  
**ORDERS3**( Order#, OrdDate,**Supp#** )  
**SUPPLIERS3**( Supp#, Supplier ).  
**ORDERLINES3** ( ***Order#***, ***Prod#***, Quantity )  
**PRODUCTS3** ( Prod#, Product, Price ).

The ***foreign key*** **ORDERS3.Supp#** relates to the ***primary key*** **SUPPLIERS3.Supp#**  
The ***foreign key* ORDERLINES3.Order#** relates to the ***primary key* ORDERS3.Order#**  
The ***foreign key*** **ORDERLINES3.Prod#** relates to the ***primary key*** **PRODUCTS3.Prod#**

The complete set of **3NF** relations for the Purchase Order data model is illustrated in the tables below:

**ORDERS3**( Order#, OrdDate, **Supp#** )

|  |  |  |
| --- | --- | --- |
| **Order#** | **OrdDate** | **Supp#** |
| 95010 | 1.1.95 | 102 |
| 95011 | 2.1.95 | 111 |
| 95012 | 2.1.95 | 144 |
| 95013 | 3.1.95 | 102 |

**SUPPLIERS3**( Supp#, Supplier )

|  |  |
| --- | --- |
| **Supp#** | **Supplier** |
| 102 | PARKER |
| 111 | HUNTER |
| 144 | PARKER |

**ORDERLINES3** ( ***Order***#, ***Prod#***, Quantity )

|  |  |  |
| --- | --- | --- |
| ***Order#*** | ***Prod#*** | **Quantity** |
| 95010 | A30 | 30 |
| 95010 | S10 | 10 |
| 95010 | C20 | 20 |
| 95011 | A30 | 20 |
| 95011 | P20 | 10 |
| 95012 | A31 | 5 |
| 95012 | P21 | 5 |
| 95013 | S10 | 20 |

**PRODUCTS3** ( Prod#, Product, Price )

|  |  |  |
| --- | --- | --- |
| **Prod#** | **Product** | **Price** |
| A30 | APPLES | 1 |
| S10 | STRAWBERRIES | 2 |
| C20 | CHERRIES | 1.5 |
| P20 | PEARS | 3 |
| A31 | APPLES | 1.5 |
| P21 | PEARS | 3.5 |

Converting the **2NF** relations to **3NF** relations has resolved the following problems:

* no duplication of data remains in **ORDERS3**
* a new supplier can be inserted without an order
* deleting Order# 95011 does not delete all reference to 111 | HUNTER
* updating 102 | PARKER requires changing only one record

[Continue to Top-down data modelling using ER Diagrams](https://www.dlsweb.rmit.edu.au/Toolbox/ecommerce/dad_respak/dad_tutorial/html/dad_db3i_tut.htm)