NORMALIZATION EXAMPLE

1st Normal Form

- A relation is said to be in 1NF if it has
  - No repeating Group same information in multiple columns
  - A Primary Key has been defined to uniquely identify a record.
  - All Data Values are Atomic

  Consider the figure below of INVOICE data which contains the repeating groups.

Order_ID	Order_ Date	Customer_ ID	Customer_ Name	Customer_ Address	Product_ID	Product_ Description	Product_ Finish	Unit_ Price	Ordered_ Quantity
1006	10/24/2006	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
					5	Writer's Desk	Cherry	325.00	2
					4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2006	6	Furniture Gallery	Boulder, CO	11	4-Dr Dresser	Oak	500.00	4
					4	Entertainment Center	Natural Maple	650.00	3

- In last figure of INVOICE data you can easily view Repeating Group Data Such as Order id 1006 has 3 orders.

Order_ID	Order_ Date	Customer_ ID	Customer_ Name	Customer_ Address	Product_ID	Product_ Description	Product_ Finish	Unit_ Price	Ordered_ Quantity
1006	10/24/2006	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
1006	10/24/2006	2	Value Furniture	Plano, TX	5	Writer's Desk	Cherry	325.00	2
1006	10/24/2006	2	Value Furniture	Plano, TX	4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2006	6	Furniture Gallery	Boulder, CO	11	4–Dr Dresser	Oak	500.00	4
1007	10/25/2006	6	Furniture Gallery	Boulder, CO	4	Entertainment Center	Natural Maple	650.00	3

After Break Down!

Order_ID	Order_ Date	Customer_ ID
1006	10/24/2006	2
1006	10/24/2006	2
1006	10/24/2006	2
1007	10/25/2006	6
1007	10/25/2006	6

After Break Down!

Customer_ ID	Customer_ Name	Customer_ Address
2	Value Furniture	Plano, TX
2	Value Furniture	Plano, TX
2	Value Furniture	Plano, TX
6	Furniture Gallery	Boulder, CO
6	Furniture Gallery	Boulder, CO

After Break Down!

Product_ID	Product_ Description	Product_ Finish	Unit_ Price
7	Dining Table	Natural Ash	800.00
5	Writer's Desk	Cherry	325.00
4	Entertainment Center	Natural Maple	650.00
11	4-Dr Dresser	Oak	500.00
4	Entertainment Center	Natural Maple	650.00

- Identification of Keys : Those keys must be identified which can uniquely identify a row.

Product_ID → Product_Description, Product_Finish, Unit_Price
Order_ID, Product_ID → Ordered_Quantity

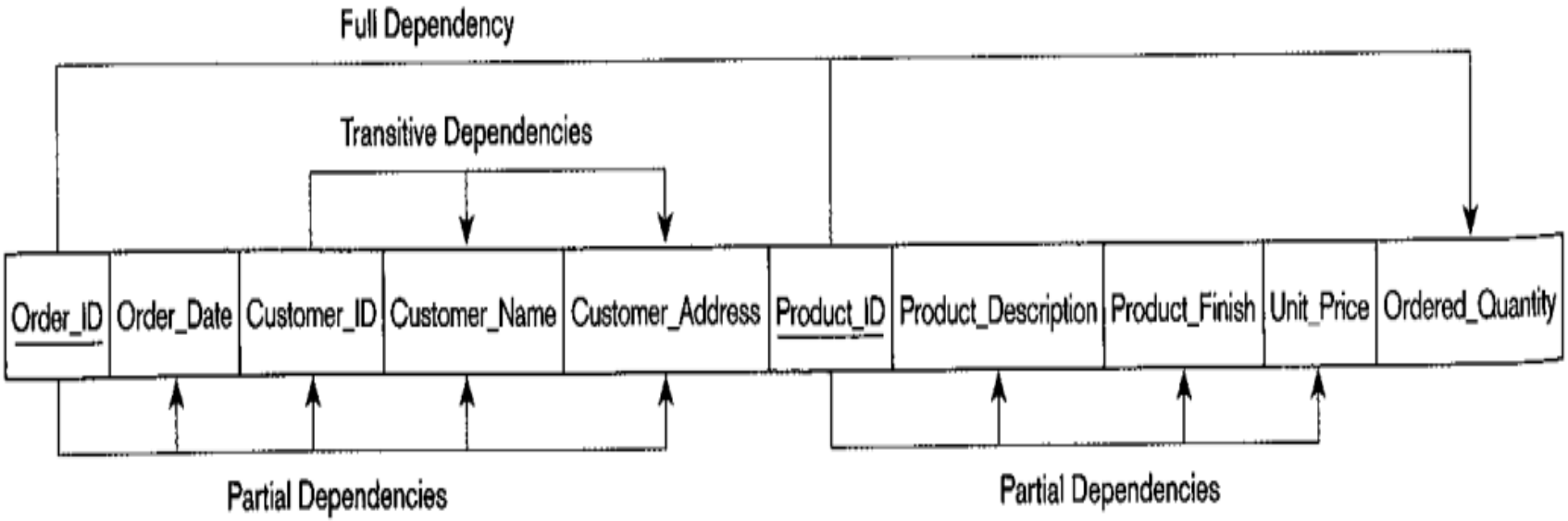
As you can see, the only candidate key for INVOICE is the composite key consisting of the attributes Order_ID and Product_ID (because there is only one row in the table for any combination of values for these attributes). Therefore, Order_ID and Product_ID are underlined in Figure 5-26, indicating that they comprise the primary key.

When forming a primary key, you must be careful not to include redundant (therefore unnecessary) attributes. Thus, although Customer_ID is a determinant in INVOICE, it is not included as part of the primary key because all of the nonkey attributes are identified by the combination of Order_ID and Product_ID. We will see the role of Customer_ID in the normalization process that follows.

A diagram that shows these functional dependencies for the INVOICE relation is shown in Figure 5-27. This diagram is a horizontal list of all the attributes in INVOICE, with the primary key attributes (Order_ID and Product_ID) underlined. Notice that the only attribute that depends on the full key is Ordered_Quantity. All of the other functional dependencies are either partial dependencies or transitive dependencies (both are defined below).

Functional Dependency Diagram

Figure 5-27
Functional dependency diagram for INVOICE



2nd Normal Form

We can remove many of the redundancies (and resulting anomalies) in the INVOICE relation by converting it to second normal form. A relation is in **second normal form (2NF)** if it is in first normal form and contains no partial dependencies. **A partial functional dependency** exists when a nonkey attribute is functionally dependent on part (but not all) of the primary key. As you can see, the following partial dependencies exist in Figure 5-27:

Order_ID → Order_Date, Customer_ID, Customer_Name, Customer_Address
Product_ID → Product_Description, Product_Finish, Unit_Price

To convert a relation with partial dependencies to second normal form, the following steps are required:

1. Create a new relation for each primary key attribute (or combination of attributes) that is a determinant in a partial dependency. That attribute is the primary key in the new relation.
2. Move the nonkey attributes that are dependent on this primary key attribute (or attributes) from the old relation to the new relation.

The results of performing these steps for the INVOICE relation are shown in Figure 5-28. Removal of the partial dependencies results in the formation of two new relations: PRODUCT and CUSTOMER_ORDER. The INVOICE relation is now left with just the primary key attributes (Order_ID and Product_ID) and Ordered_Quantity, which is functionally dependent on the whole key. We rename this relation ORDER_LINE, because each row in this table represents one line item on an order.

As indicated in Figure 5-28, the relations ORDER_LINE and PRODUCT are in third normal form. However, CUSTOMER_ORDER contains transitive dependencies and therefore (although in second normal form) is not yet in third normal form.

2NF (2)

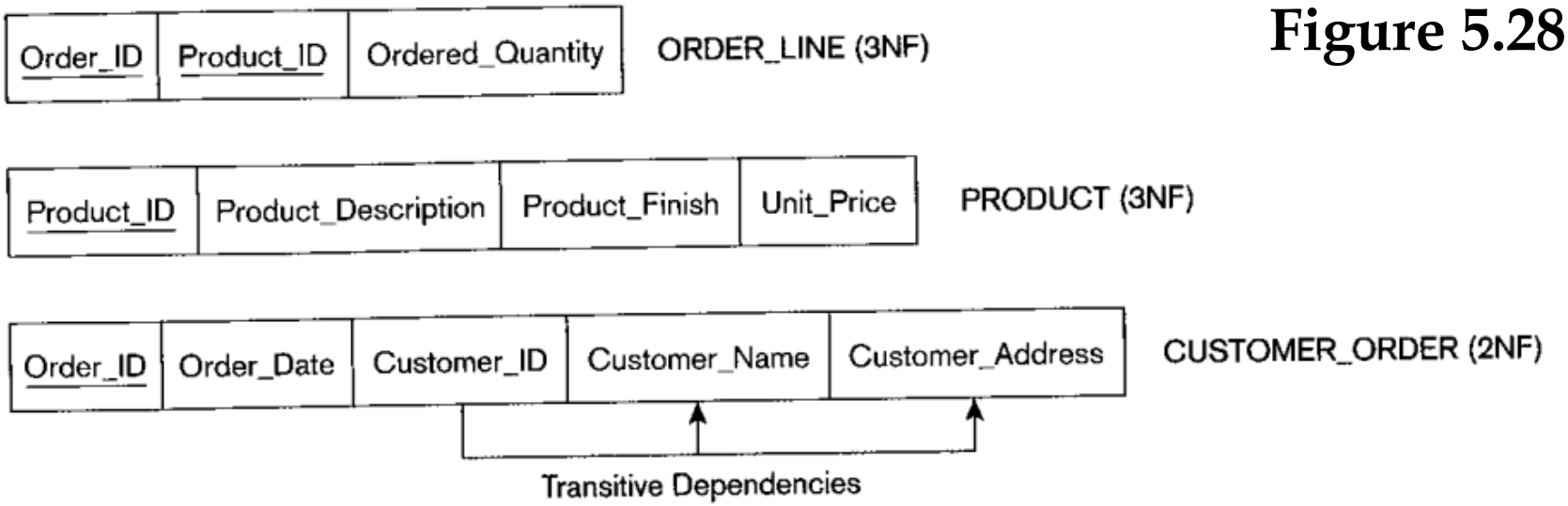


Figure 5.28

3rd Normal Form (1)

A relation is in **third normal form** (3NF) if it is in second normal form and no transitive dependencies exist. A **transitive dependency** in a relation is a functional dependency between two (or more) nonkey attributes. For example, there are two transitive dependencies in the CUST_ORDER relation shown in Figure 5-28:

Customer_ID → Customer_Name, and
Customer_ID → Customer_Address.

In other words, both customer name and address are uniquely identified by the Customer_ID, but Customer_ID is not part of the primary key (as we noted earlier).

Transitive dependencies create unnecessary redundancy that may lead to the type of anomalies discussed earlier. For example, the transitive dependency in CUSTOMER_ORDER (Figure 5-28) requires that a customer's name and address be reentered every time a customer submits a new order, regardless of how many times they have been entered previously. You have no doubt experienced this type of annoying requirement when ordering merchandise online, visiting a doctor's office, or any number of similar activities.

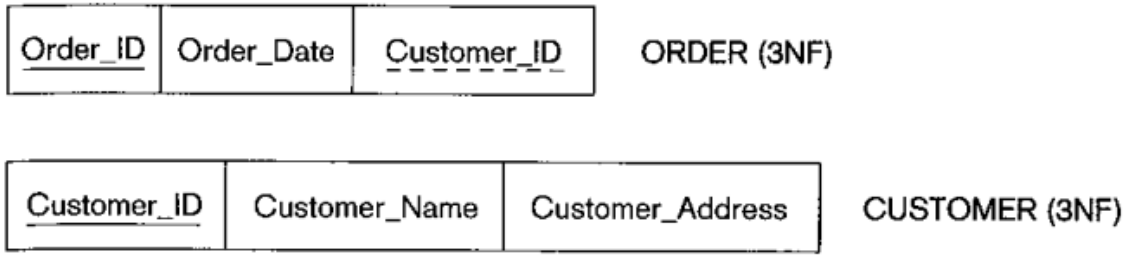
3rd Normal Form (2)

Removing Transitive Dependencies You can easily remove transitive dependencies from a relation by means of a three-step procedure:

1. For each nonkey attribute (or set of attributes) that is a determinant in a relation, create a new relation. That attribute (or set of attributes) becomes the primary key of the new relation.
2. Move all of the attributes that are functionally dependent on the attribute from the old to the new relation.
3. Leave the attribute (which serves as a primary key in the new relation) in the old relation to serve as a foreign key that allows you to associate the two relations.

3rd Normal Form (3)

Fig 5.29



The results of applying these steps to the relation CUSTOMER_ORDER are shown in Figure 5-29. A new relation named CUSTOMER has been created to receive the components of the transitive dependency. The determinant Customer_ID becomes the primary key of this relation, and the attributes Customer_Name and Customer_Address are moved to the relation. CUST_ORDER is renamed ORDER, and the attribute Customer_ID remains as a foreign key in that relation. This allows us to associate an order with the customer who submitted the order. As indicated in Figure 5-29, these relations are now in third normal form.

3rd Normal Form (4)

Normalizing the data in the INVOICE view has resulted in the creation of four relations in third normal form: CUSTOMER, PRODUCT, ORDER, and ORDER_LINE. A relational schema showing these four relations and their associations (developed using Microsoft Visio) is shown in Figure 5-30. Note that Customer_ID is a foreign key in ORDER and Order_ID and Product_ID are foreign keys in ORDER_LINE. (Foreign keys are shown in Visio for logical, but not conceptual, data models.) Also note that minimum cardinalities are shown on the relationships even though the normalized relations provide no evidence of what the minimum cardinalities should be. Sample data for the relations might include, for example, a customer with no orders, thus providing evidence of the optional cardinality for the relationship Places. However, even if there were an order for every customer in a sample data set, this would not prove mandatory cardinality. Minimum cardinalities must be determined from business rules not illustrations of reports, screens, and transactions. The same statement is true for specific maximum cardinalities (for example, a business rule that no order may contain more than ten line items).

3rd Normal Form (4)

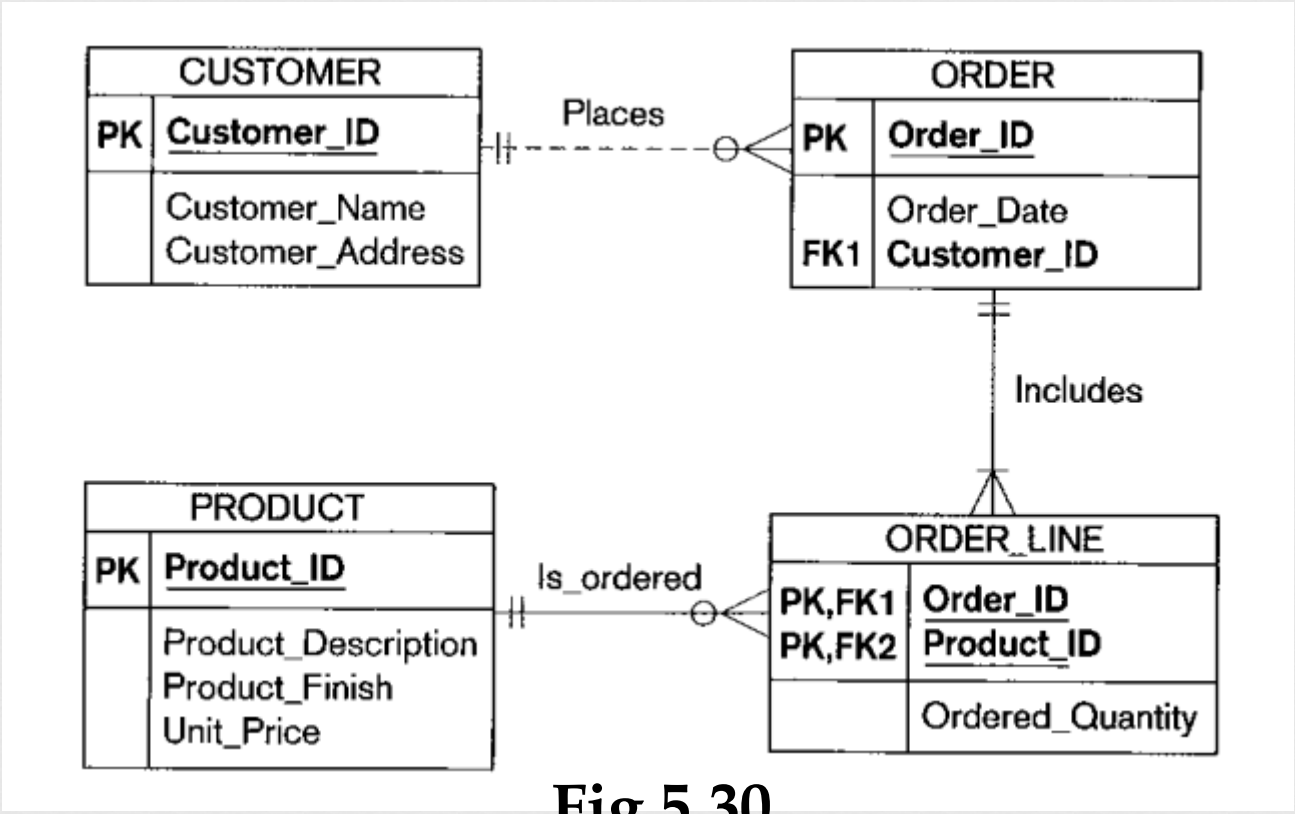


Fig 5.30