

1. Normalize up to 3rd normal forms and give suitable name to each of the new relations.

PROD_SUPP_LIST (Product_no, Product_name, Product_typ_code, Product_typ_name, Supplier_no, Supplier_name, product price, description)

2. Fields in the original data table will be as follows:

SalesOrderNo, Date, CustomerNo, CustomerName, CustomerAdd, ClerkNo, ClerkName, ItemNo, Description, Qty, UnitPrice

Think of this as the baseline – one large table. Normalize up to 3rd normal forms and give suitable name to each of the new relations.

Normalization

Normalization is a method for organizing data elements in a database into tables.

Normalization Avoids

- Duplication of Data – The same data is listed in multiple lines of the database
- Insert Anomaly – A record about an entity cannot be inserted into the table without first inserting information about another entity – Cannot enter a customer without a sales order
- Delete Anomaly – A record cannot be deleted without deleting a record about a related entity. Cannot delete a sales order without deleting all of the customer's information.
- Update Anomaly – Cannot update information without changing information in many places. To update customer information, it must be updated for each sales order the customer has placed

Normalization is a three stage process – After the first stage, the data is said to be in first normal form, after the second, it is in second normal form, after the third, it is in third normal form

Before Normalization

1. Begin with a list of all of the fields that must appear in the database. Think of this as one big table.
2. Do not include computed fields
3. One place to begin getting this information is from a printed document used by the system.
4. Additional attributes besides those for the entities described on the document can be added to the database.

Before Normalization – Example

See Sales Order from below:

Sales Order

Fiction Company
202 N. Main
Mahattan, KS 66502

CustomerNumber: 1001
Customer Name: ABC Company
Customer Address: 100 Points
Manhattan, KS 66502

Sales Order Number: 405
Sales Order Date: 2/1/2000
Clerk Number: 210
Clerk Name: Martin Lawrence

Item Ordered	Description	Quantity	Unit Price	Total
800	widgit small	40	60.00	2,400.00
801	tingimajigger	20	20.00	400.00
805	thingibob	10	100.00	1,000.00
Order Total				3,800.00

Fields in the original data table will be as follows:

SalesOrderNo, Date, CustomerNo, CustomerName, CustomerAdd, ClerkNo, ClerkName,
ItemNo, Description, Qty, UnitPrice

Think of this as the baseline – one large table

Normalization: First Normal Form

- Separate Repeating Groups into New Tables.
- **Repeating Groups** Fields that may be repeated several times for one document/entity
- Create a new table containing the repeating data
- The primary key of the new table (repeating group) is always a composite key; Usually document number and a field uniquely describing the repeating line, like an item number.

First Normal Form Example

The new table is as follows:

SalesOrderNo, ItemNo, Description, Qty, UnitPrice

The repeating fields will be removed from the original data table, leaving the following.

SalesOrderNo, Date, CustomerNo, CustomerName, CustomerAdd, ClerkNo, ClerkName

These two tables are a database in first normal form

What if we did not Normalize the Database to First Normal Form?

Repetition of Data – SO Header data repeated for every line in sales order.

Normalization: Second Normal Form

- Remove Partial Dependencies.
- **Functional Dependency** The value of one attribute in a table is determined entirely by the value of another.
- **Partial Dependency** A type of functional dependency where an attribute is functionally dependent on only part of the primary key (primary key must be a composite key).
- Create separate table with the functionally dependent data and the part of the key on which it depends. Tables created at this step will usually contain descriptions of resources.

Second Normal Form Example

The new table will contain the following fields:

ItemNo, Description

All of these fields except the primary key will be removed from the original table. The primary key will be left in the original table to allow linking of data:

SalesOrderNo, ItemNo, Qty, UnitPrice

Never treat price as dependent on item. Price may be different for different sales orders (discounts, special customers, etc.)

Along with the unchanged table below, these tables make up a database in second normal form:

SalesOrderNo, Date, CustomerNo, CustomerName, CustomerAdd, ClerkNo, ClerkName

What if we did not Normalize the Database to Second Normal Form?

- Repetition of Data – Description would appear every time we had an order for the item
- Delete Anomalies – All information about inventory items is stored in the SalesOrderDetail table. Delete a sales order, delete the item.

- Insert Anomalies – To insert an inventory item, must insert sales order.
- Update Anomalies – To change the description, must change it on every SO.

Normalization: Third Normal Form

- Remove transitive dependencies.
- **Transitive Dependency** A type of functional dependency where an attribute is functionally dependent on an attribute other than the primary key. Thus its value is only indirectly determined by the primary key.
- Create a separate table containing the attribute and the fields that are functionally dependent on it. Tables created at this step will usually contain descriptions of either resources or agents. Keep a copy of the key attribute in the original file.

Third Normal Form Example

The new tables would be:

CustomerNo, CustomerName, CustomerAdd

ClerkNo, ClerkName

All of these fields except the primary key will be removed from the original table. The primary key will be left in the original table to allow linking of data as follows:

SalesOrderNo, Date, CustomerNo, ClerkNo

Together with the unchanged tables below, these tables make up the database in third normal form.

ItemNo, Description

SalesOrderNo, ItemNo, Qty, UnitPrice

What if we did not Normalize the Database to Third Normal Form?

- Repetition of Data – Detail for Cust/Clerk would appear on every SO
- Delete Anomalies – Delete a sales order, delete the customer/clerk
- Insert Anomalies – To insert a customer/clerk, must insert sales order.
- Update Anomalies – To change the name/address, etc, must change it on every SO.

Completed Tables in Third Normal Form

Customers: CustomerNo, CustomerName, CustomerAdd

Clerks: ClerkNo, ClerkName

Inventory Items: ItemNo, Description

Sales Orders: SalesOrderNo, Date, CustomerNo, ClerkNo

SalesOrderDetail: SalesOrderNo, ItemNo, Qty, UnitPrice