

**Nora’s Bagel Bin Database Blueprints**

**First Normal Form (1NF)**

|  |  |
| --- | --- |
| **BAGEL ORDER** | |
| PK | Bagel Order ID |
| PK | Bagel ID |
|  | Order Date |
|  | First Name |
|  | Last Name |
|  | Address 1 |
|  | Address 2 |
|  | City |
|  | State |
|  | Zip |
|  | Mobile Phone |
|  | Delivery Fee |
|  | Bagel Name |
|  | Bagel Description |
|  | Bagel Price |
|  | Bagel Quantity |
|  | Special Notes |

This 1NF table achieves the following:

* captures all necessary data fields from the unnormalized “Nora’s Bagel Bin Catering Order” form
* excludes calculated fields like subtotal and sales tax as it is not necessary to store these values
* establishes primary keys
* does not contain any repeated groups (data that depends on both pieces of the composite primary key)
* contains only atomic data (no attributes will contain more than one piece of data)

***NOTE:*** The attributes given above (e.g., address lines, phone number) do *not* need to be processed any further into subcomponents during the normalization process.

**Nora’s Bagel Bin Database Blueprints *(continued)***

**Second Normal Form (2NF)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **BAGEL ORDER** | |  | **BAGEL ORDER LINE ITEM** | |  | **BAGEL** | |
| PK | Bagel Order ID |  | FK | Bagel Order ID |  | PK | Bagel ID |
|  | First Name | 1:M | FK | Bagel ID | M:1 |  | Bagel Name |
|  | Last Name |  |  | Bagel Quantity |  |  | Bagel Description |
|  | Address 1 |  |  |  |  |  | Bagel Price |
|  | Address 2 |  |  |  |  |  |  |
|  | City |  |  |  |  |  |  |
|  | State |  |  |  |  |  |  |
|  | Zip |  |  |  |  |  |  |
|  | Mobile Phone |  |  |  |  |  |  |
|  | Delivery Fee |  |  |  |  |  |  |
|  | Special Notes |  |  |  |  |  |  |
|  | Order Date |  |  |  |  |  |  |

Explanation:

=> Bagel-specific attributes are assigned to the BAGEL table first.

=> Our join table, “BAGEL ORDER LINE ITEM” needs 2 foreign keys (bagel\_order\_id and bagel\_id) to join the other two tables and bagel quantity to calculate the total price.

=> And order-specific attributes are assigned to the “BAGEL ORDER” table.

=> Cardinality: “BAGEL ORDER LINE ITEM” table’s rows can have multiple bagle\_order\_ID and bagel\_ID, But One bagel\_order\_id or bagel\_id corresponds to only one unique row (bagel\_order\_id and bagel\_id) in both “BAGEL ORDER” and “BAGEl” tables. So the relationship between “BAGEL ORDER” and “BAGEL ORDER LINE ITEM” tables is “one to many”, at the same time “BAGEL” and “BAGEL ORDER LINE ITEM” tables is “many to one”

* Separate the data that depend on just one of the two parts of the primary key into separate tables.
* Keep any columns in the original table (now named "Bagel Order Line Item") that still depend on both parts of the original primary key.
* Fill in each shaded cell within the tables with an attribute from the 1NF blueprint in the previous section.
* Fill in the dotted cells between the tables with the cardinality of the relationship between those two tables: one-to-one (1:1), one-to-many (1:M), many-to-one (M:1), or many-to-many (M:M).

**Nora’s Bagel Bin Database Blueprints *(continued)***

**Third Normal Form (3NF)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BAGEL ORDER** | | |  | **BAGEL ORDER LINE ITEM** | |  | **BAGEL** | |
| PK | Bagel Order ID | |  | FK | Bagel Order ID |  | PK | Bagel ID |
| FK | Customer ID | | 1:M | FK | Bagel ID | M:1 |  | Bagel Name |
|  | Delivery Fee | |  |  | Bagel Quantity |  |  | Bagel Description |
|  | Special Notes | |  |  |  |  |  | Bagel Price |
|  | Order Date | |  |  |  |  |  |  |
|  | M:1 |  |  |  |  |  |  |  |
| **CUSTOMER** | | |  |  |  |  |  |  |
| PK | Customer ID | |  |  |  |  |  |  |
|  | First Name | |  |  |  |  |  |  |
|  | Last Name | |  |  |  |  |  |  |
|  | Address 1 | |  |  |  |  |  |  |
|  | Address 2 | |  |  |  |  |  |  |
|  | City | |  |  |  |  |  |  |
|  | State | |  |  |  |  |  |  |
|  | Zip | |  |  |  |  |  |  |
|  | Mobile Phone | |  |  |  |  |  |  |

**Explanation:**

=> Bagel-specific attributes are assigned to the BAGEL table first.

=> Our join table, “BAGEL ORDER LINE ITEM” needs 2 foreign keys (bagel\_order\_id and bagel\_id) to join the other two tables and bagel quantity to calculate the total price.

=>And order-specific attributes are assigned to the “BAGEL ORDER” table.

=> Customer-specific attributes are assigned to the new “CUSTOMER” table.

=> Cardinality: One customer can have multiple bagle\_order\_id (Bagel Order) but one bagel\_order\_id corresponds to only one “customer\_id”, so top-to-bottom the cardinality here is “Many to One”. Also “BAGEL ORDER LINE ITEM” table’s rows can have multiple bagle\_order\_ID and bagel\_ID, But One bagel\_order\_id or bagel\_id corresponds to only one unique row (bagel\_order\_id and bagel\_id) in both “BAGEL ORDER” and “BAGEl” tables. So the relationship between “BAGEL ORDER” and “BAGEL ORDER LINE ITEM” tables is “one to many”, at the same time “BAGEL” and “BAGEL ORDER LINE ITEM” tables is “many to one”

* Look for remaining data that are or could be repeated within each table but do not depend on the primary key.
* Move that repeated data into their own table by filling in the shaded cells with attributes from your 2NF diagram.
* Create a new attribute to be the primary key for this new table and also use it as the foreign key linking to this new table; fill in the appropriate shaded cells with this new attribute.
* Fill in the dotted cell between the two new tables with the cardinality of the relationship between those two tables: one-to-one (1:1), one-to-many (1:M), many-to-one (M:1), or many-to-many (M:M).
* Give the two new tables appropriate names.
* Fill in the remaining shaded cells from any unchanged tables using the information from your 2NF diagram.

**Nora’s Bagel Bin Database Blueprints *(continued)***

**Final Physical Database Model**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BAGEL ORDER** | | |  | **BAGEL ORDER LINE ITEM** | | |  | **BAGEL** | |  |
| PK | bagel\_order\_id | INT |  | FK | bagel\_order\_id | INT |  | PK | bagel\_id | INT |
| FK | customer\_id | CHAR() | 1:M | FK | bagel\_id | INT | M:1 |  | bagel\_name | VARCHAR() |
|  | delivery\_fee | NUMERIC() |  |  | bagel\_quantity | NUMERIC() |  |  | bagel\_description | VARCHAR() |
|  | special\_notes | VARCHAR() |  |  |  |  |  |  | bagel\_price | NUMERIC() |
|  | order\_date | TIMESTAMP |  |  |  |  |  |  |  |  |
|  | M:1 |  |  |  |  |  |  |  |  |  |
| **CUSTOMER** | | |  |  |  |  |  |  |  |  |
| PK | customer\_id | CHAR() |  |  |  |  |  |  |  |  |
|  | first\_name | VARCHAR() |  |  |  |  |  |  |  |  |
|  | last\_name | VARCHAR() |  |  |  |  |  |  |  |  |
|  | address\_1 | VARCHAR() |  |  |  |  |  |  |  |  |
|  | address 2 | VARCHAR() |  |  |  |  |  |  |  |  |
|  | city | VARCHAR() |  |  |  |  |  |  |  |  |
|  | state | VARCHAR() |  |  |  |  |  |  |  |  |
|  | zip | NUMERIC() |  |  |  |  |  |  |  |  |
|  | mobile\_phone | NUMERIC() |  |  |  |  |  |  |  |  |

* Fill in the table names, attribute name, and table relationship cardinalities using the values from your completed 3NF diagram.
  + Rename any fields that have unusable database characters, like spaces (e.g., change “First Name” to first\_name or firstname.)
* Assign one of the following five data types to each attribute: CHAR(), VARCHAR(), TIMESTAMP, INTEGER, or NUMERIC().
  + Each of the five data types must be used at least once in your database model.

***NOTE:*** To make your final project report as professional as possible, you should remove the shading from the cells of your final diagrams and take a screenshot (cropped) of the tables or cut and paste them into your final project report without the accompanying instructions in this file.