

**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 |  |  | PK / FK | Bagel Order ID |  | PK |  |
|  |  |  | PK / FK | Bagel ID |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

* 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 LINE ITEM** | |  | **BAGEL** | |
| PK |  | |  | PK / FK | Bagel Order ID |  | PK |  |
| FK |  | |  | PK / FK | Bagel ID |  |  |  |
|  |  | |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  | | |  |  |  |  |  |  |
| PK |  | |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |

* 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 LINE ITEM** | | |  | **BAGEL** | |  |
| PK |  |  |  | PK / FK | bagel\_order\_id | INT |  | PK |  |  |
| FK |  |  |  | PK / FK | bagel\_id | CHAR(2) |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | | |  |  |  |  |  |  |  |  |
| PK |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

* 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.