We will follow a systematic approach to categorizing the provided data fields into tables and ensuring they are in 3NF. The initial step involves identifying the entities represented by the data fields and their relationships. In this case, the entities are **Books**, **Authors**, and **Publishers**.

*1: Identify Entities and Attributes*

1. **Books**
   * book\_isbn (Primary Key)
   * book\_name
   * book\_price
   * publisher\_ID (Foreign Key)
2. **Authors**
   * author\_id (Primary Key)
   * author\_first\_name
   * author\_last\_name
   * author\_phone
   * author\_email
   * author\_address
3. **Publishers**
   * publisher\_ID (Primary Key)
   * publisher\_name
   * publisher\_address
   * publisher\_email
4. **Book\_Authors** (Junction Table for many-to-many relationship between Books and Authors)
   * book\_isbn (Foreign Key)
   * author\_id (Foreign Key)

*2: Create Tables*

Now, we will create the tables based on the identified entities and their attributes.

Books Table

|  |  |  |  |
| --- | --- | --- | --- |
| **book\_isbn** | **book\_name** | **book\_price** | **publisher\_ID** |
| 978-1234567890 | Les Misérables | 20.00 | 1 |
| 978-1234567891 | The Stranger | 15.00 | 2 |
| 978-1234567892 | Anna Karenina | 25.00 | 1 |
| 978-1234567893 | The Lord of the Rings | 30.00 | 3 |
| 978-1234567894 | Lolita | 18.00 | 4 |
| 978-1234567895 | War and Peace | 22.00 | 1 |
| 978-1234567896 | The Odyssey | 19.00 | 5 |
| 978-1234567897 | The Iliad | 21.00 | 5 |

Authors Table

| **author\_id** | **author\_first\_name** | **author\_last\_name** | **author\_phone** | **author\_email** | **author\_address** |
| --- | --- | --- | --- | --- | --- |
| 1 | Victor | Hugo | 123-456-7890 | victor.hugo@example.com | 123 Victor St, Paris |
| 2 | Albert | Camus | 234-567-8901 | albert.camus@example.com | 234 Albert St, Paris |
| 3 | Leo | Tolstoy | 345-678-9012 | leo.tolstoy@example.com | 345 Leo St, Moscow |
| 4 | Vladimir | Nabokov | 456-789-0123 | vladimir.nabokov@example.com | 456 Vladimir St, St. Petersburg |
| 5 | Homer | N/A | 567-890-1234 | homer@example.com | 567 Homer St, Greece |

Publishers Table

| **publisher\_ID** | **publisher\_name** | **publisher\_address** | **publisher\_email** |
| --- | --- | --- | --- |
| 1 | Penguin Random House | 1 Publisher Rd, NY | contact@penguin.com |
| 2 | HarperCollins | 2 Publisher Rd, NY | contact@harpercollins.com |
| 3 | Houghton Mifflin | 3 Publisher Rd, NY | contact@hmh.com |
| 4 | Vintage Books | 4 Publisher Rd, NY | contact@vintage.com |
| 5 | Oxford University Press | 5 Publisher Rd, NY | contact@oup.com |

Book\_Authors Table (Junction Table)

| **book\_isbn** | **author\_id** |
| --- | --- |
| 978-1234567890 | 1 |
| 978-1234567891 | 2 |
| 978-1234567892 | 3 |
| 978-1234567893 | 3 |
| 978-1234567894 | 4 |
| 978-1234567895 | 3 |
| 978-1234567896 | 5 |
| 978-1234567897 | 5 |

*3: Ensure 3NF Compliance*

To ensure that the tables are in 3NF, we must confirm that:

1. Each table has a primary key.
2. All non-key attributes are fully functionally dependent on the primary key.
3. There are no transitive dependencies.

In our design:

* The **Books** table has a primary key of book\_isbn, and all other attributes depend solely on this key.
* The **Authors** table has a primary key of author\_id, with all attributes dependent on it.
* The **Publishers** table has a primary key of publisher\_ID, with all attributes dependent on it.
* The **Book\_Authors** junction table effectively resolves the many-to-many relationship between books and authors without introducing redundancy.

The tables have been structured to meet the requirements of 3NF, ensuring data integrity and minimizing redundancy. A junction table allows for the representation of complex relationships while maintaining clarity and organization within the database schema. This structured approach enhances the efficiency of data retrieval and supports the database's scalability for future expansions.