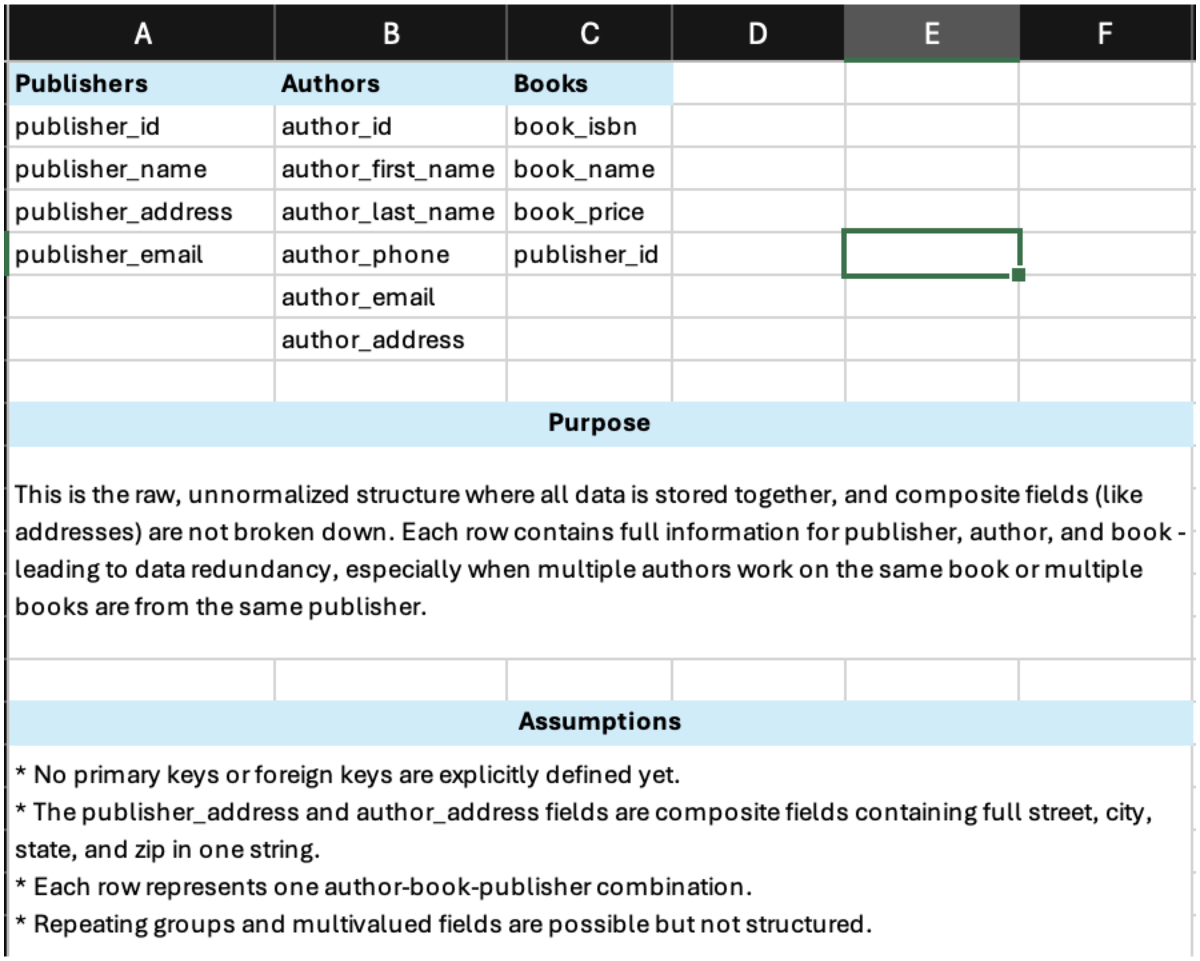
Module 3.2 Normalized Tables

SPRING 2025 CSD310 DATABASE DEVELOPMENT AND USE

Author: Brittaney Perry-Morgan
Date: Sunday, June 15th, 2025

Module 3.2 Normalized Tables

Raw Data



A	B	C	D	E	F
Publishers	**Authors**	**Books**			
publisher_id	author_id	book_isbn			
publisher_name	author_first_name	book_name			
publisher_address	author_last_name	book_price			
publisher_email	author_phone	publisher_id			
	author_email				
	author_address				

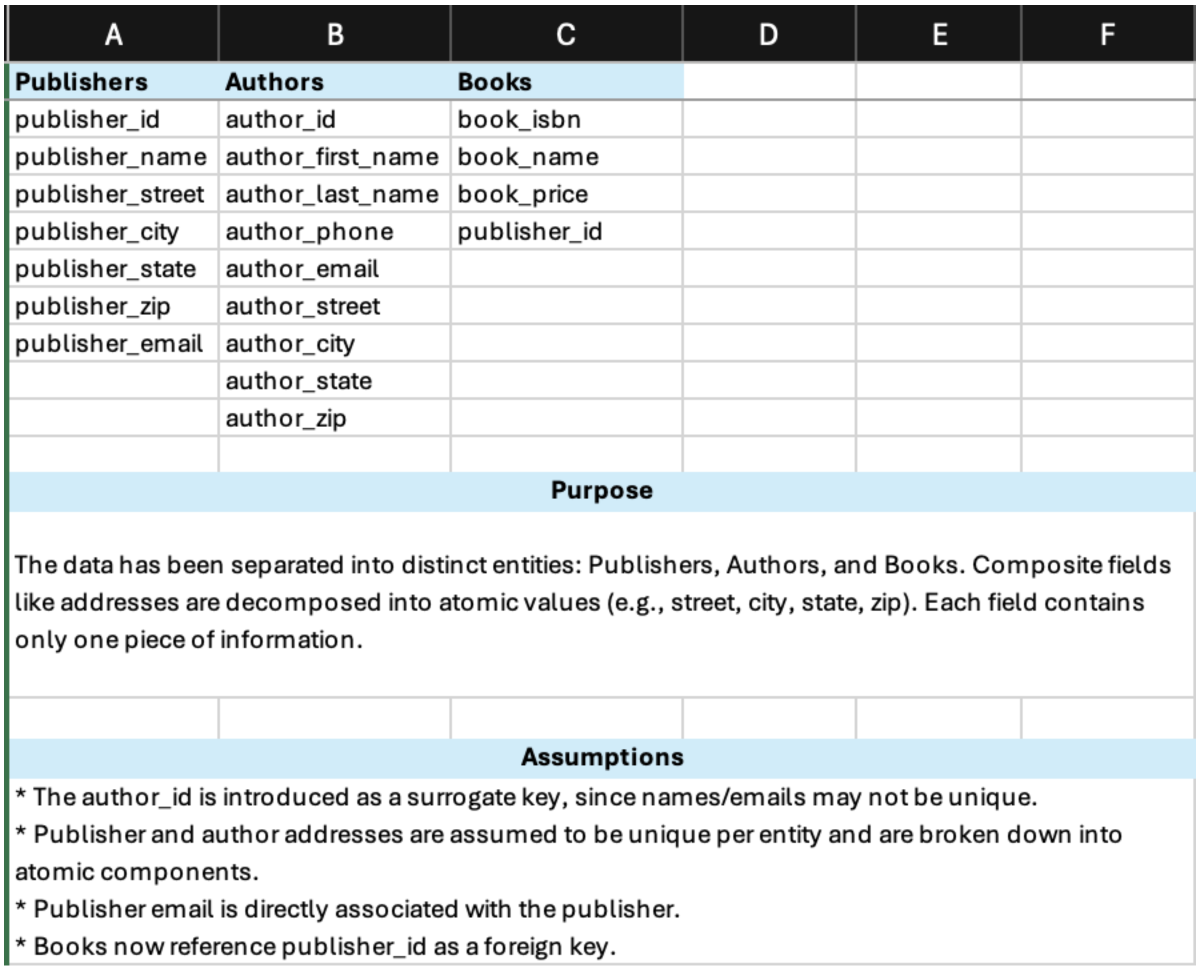
Purpose

This is the raw, unnormalized structure where all data is stored together, and composite fields (like addresses) are not broken down. Each row contains full information for publisher, author, and book - leading to data redundancy, especially when multiple authors work on the same book or multiple books are from the same publisher.

Assumptions

* No primary keys or foreign keys are explicitly defined yet.
* The publisher_address and author_address fields are composite fields containing full street, city, state, and zip in one string.
* Each row represents one author-book-publisher combination.
* Repeating groups and multivalued fields are possible but not structured.

1st Normal Form



A	B	C	D	E	F
Publishers	**Authors**	**Books**			
publisher_id	author_id	book_isbn			
publisher_name	author_first_name	book_name			
publisher_street	author_last_name	book_price			
publisher_city	author_phone	publisher_id			
publisher_state	author_email				
publisher_zip	author_street				
publisher_email	author_city				
	author_state				
	author_zip				

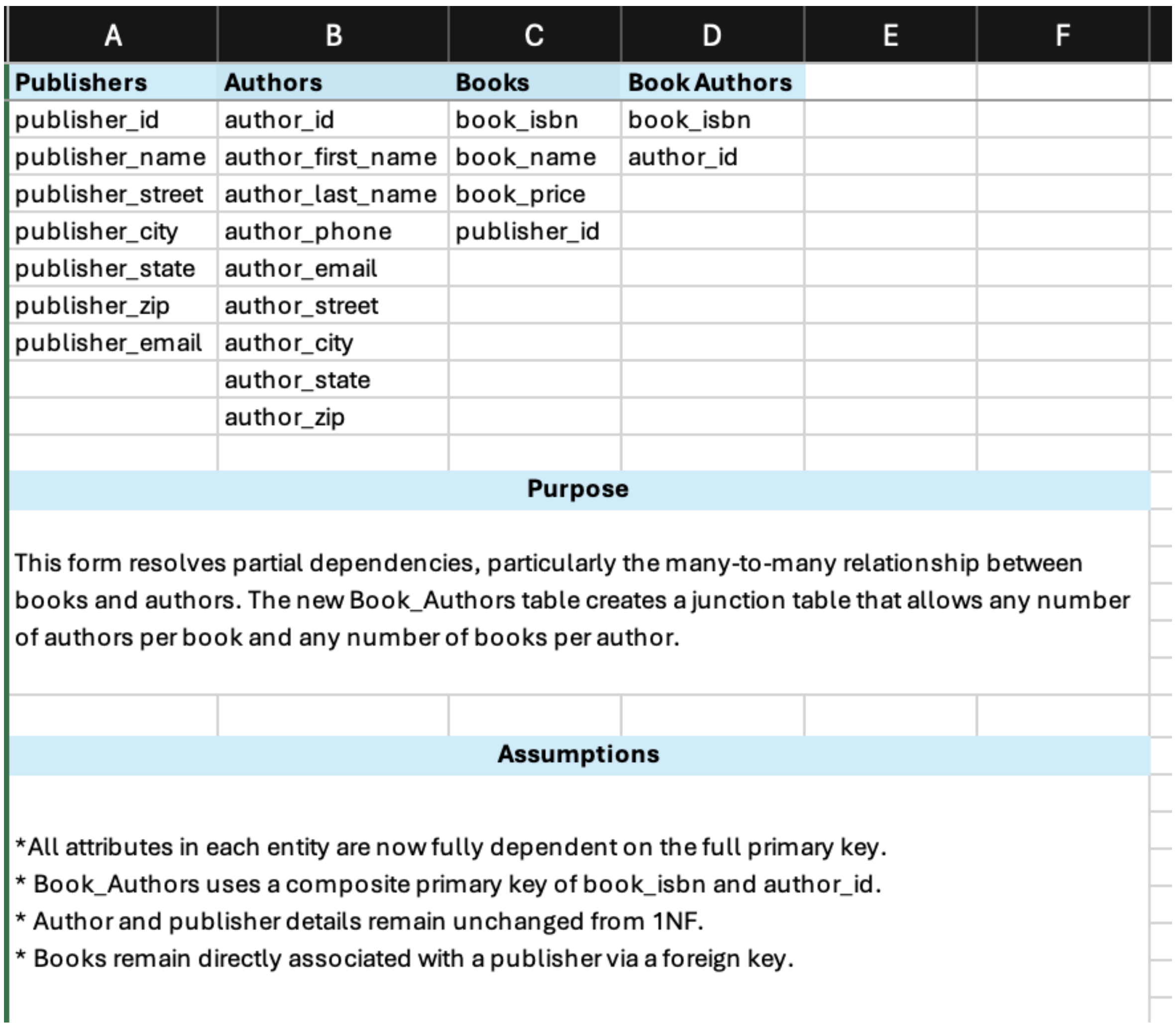
Purpose

The data has been separated into distinct entities: Publishers, Authors, and Books. Composite fields like addresses are decomposed into atomic values (e.g., street, city, state, zip). Each field contains only one piece of information.

Assumptions

* The author_id is introduced as a surrogate key, since names/emails may not be unique.
* Publisher and author addresses are assumed to be unique per entity and are broken down into atomic components.
* Publisher email is directly associated with the publisher.
* Books now reference publisher_id as a foreign key.

2nd Normal Form



A	B	C	D	E	F
Publishers	**Authors**	**Books**	**Book Authors**		
publisher_id	author_id	book_isbn	book_isbn		
publisher_name	author_first_name	book_name	author_id		
publisher_street	author_last_name	book_price			
publisher_city	author_phone	publisher_id			
publisher_state	author_email				
publisher_zip	author_street				
publisher_email	author_city				
	author_state				
	author_zip				

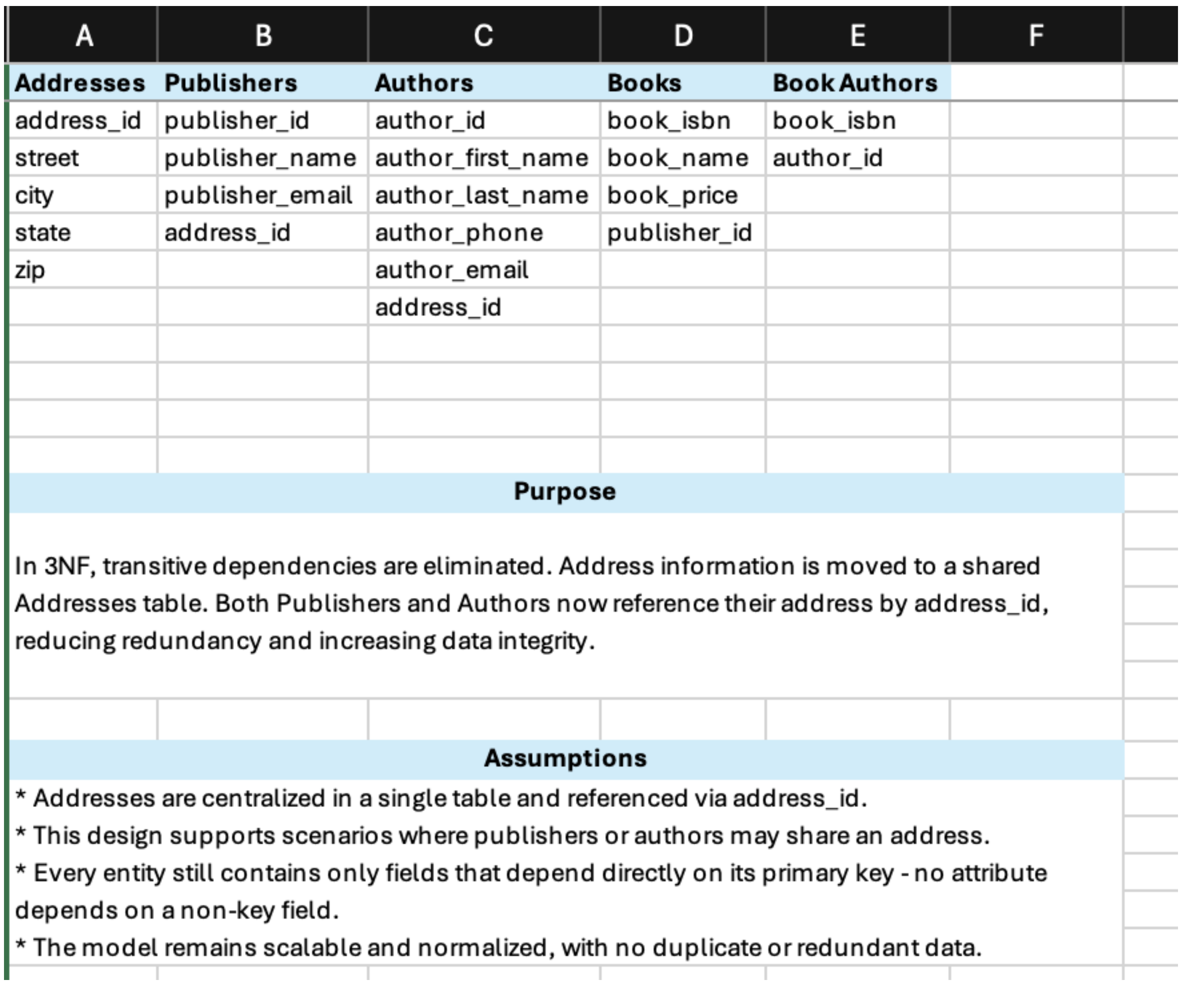
Purpose

This form resolves partial dependencies, particularly the many-to-many relationship between books and authors. The new Book_Authors table creates a junction table that allows any number of authors per book and any number of books per author.

Assumptions

*All attributes in each entity are now fully dependent on the full primary key.
* Book_Authors uses a composite primary key of book_isbn and author_id.
* Author and publisher details remain unchanged from 1NF.
* Books remain directly associated with a publisher via a foreign key.

3rd Normal Form



A	B	C	D	E	F
Addresses	**Publishers**	**Authors**	**Books**	**Book Authors**	
address_id	publisher_id	author_id	book_isbn	book_isbn	
street	publisher_name	author_first_name	book_name	author_id	
city	publisher_email	author_last_name	book_price		
state	address_id	author_phone	publisher_id		
zip		author_email			
		address_id			

Purpose

In 3NF, transitive dependencies are eliminated. Address information is moved to a shared Addresses table. Both Publishers and Authors now reference their address by address_id, reducing redundancy and increasing data integrity.

Assumptions

* Addresses are centralized in a single table and referenced via address_id.
* This design supports scenarios where publishers or authors may share an address.
* Every entity still contains only fields that depend directly on its primary key - no attribute depends on a non-key field.
* The model remains scalable and normalized, with no duplicate or redundant data.