Normalization Documentation: From Initial ERD to Final ERD

Goal

To transform the initial ERD into a fully normalized schema by applying the rules of **1NF**, **2NF**, and **3NF**, thereby reducing redundancy, improving data integrity, and enhancing maintainability.

Initial ERD Issues (Denormalized / Partially Normalized)

The initial model had:

- Redundant or loosely structured relationships (e.g., shopping cart and cart items separate without a refined link).
- Some transitive dependencies and room for data duplication.
- Missing or merged concepts like promotions embedded in products.
- Lack of normalization in terms of time-dependent data (e.g., discounts).

Normalization Steps Followed

Step 1: Ensured Atomicity (1NF)

Applied To: All tables
Changes Made:

- Ensured all attributes in every table are atomic (no multi-valued attributes).
- Example: Product_Name, Product_Description, Order_Status, Shipping_Method are stored as single, indivisible fields.

Step 2: Removed Partial Dependencies (2NF)

Applied To: Tables with composite keys like ShoppingCartItems, Reviews, etc.
Changes Made:

- Split composite-key dependent tables and ensured each non-key attribute depends on the full primary key.
- Separated CartItems (now **removed**) logic — the final ERD removed the ShoppingCartItems and Shopping_Cart tables, simplifying the model.
- Focus was moved to OrderItems, which more accurately represents finalized purchases, not transient cart data.

Step 3: Removed Transitive Dependencies (3NF)

Applied To: Products, Orders, Users, and Payments
Changes Made:

- **Product Promotions** were separated into a new Products_Promotions table:
    - Eliminates transitive dependency of time-bound discount data from the Products table.
    - Introduces a normalized way to handle discounts (Discount_Percentage, Start_Date, End_Date).
- Separated user information (Users) cleanly with only required fields and moved away from storing login-related metadata redundantly.
- Payment and review data now fully dependent only on their respective primary keys (no unnecessary coupling).

Step 4: Introduced New Entities to Support Normalization

Added Tables:

- Products_Promotions: Handles time-dependent, discount-related data (solving redundancy across products).
- OrderItems: Breaks many-to-many between Orders and Products into atomic rows.

These changes reflect:

- **Best Practices of 3NF and BCNF**.
- Enable tracking of promotional periods without duplicating discount logic in Products.

Step 5: Cleaned Foreign Keys and Referential Integrity

Changes Made:

- Foreign keys now strictly represent parent-child dependencies.
    - E.g., Orders.User_ID → Users.User_ID, OrderItems.Order_ID → Orders.Order_ID
- Indexes added for better performance (Payments__IDX, orders__IDXv1)

Final Result: Fully Normalized ERD

Normal Form	**Criteria Met**	**Description**
1NF	Yes	All attributes are atomic.
2NF	Yes	Removed partial dependencies and composite key issues.
3NF	Yes	No transitive dependencies; all non-key attributes depend on the key only.
BCNF+	Optional	Handled through added tables like Products_Promotions to resolve functional dependencies beyond 3NF.

Summary of Key Improvements

Improvement	**Initial ERD**	**Final ERD**
Promotion data handling	Stored or assumed inside Products	Separated as Products_Promotions
Cart data (Shopping_Cart)	Present but not normalized	Removed for clarity
Order items structure	ShoppingCartItems shared logic	Introduced OrderItems
Data types	Generic types like NUMBER and VARCHAR2	Introduced precision and timestamps
Redundancy	Moderate duplication (e.g., discount logic)	Eliminated through proper decomposition