Database system and database design

Database Normalization PART 1

Agenda

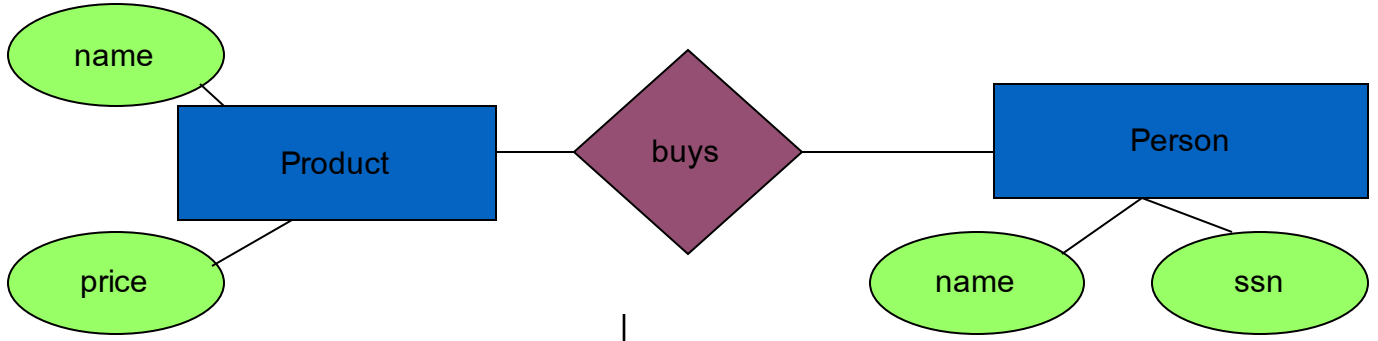
- Normal Form
- Midterm review

The benefits of normalization

- Retrieval more efficient
- Insert, update, and delete operation more efficient
- Reduce data redundancy
- Simplifies maintenance and reduces storage

Relational Schema Design

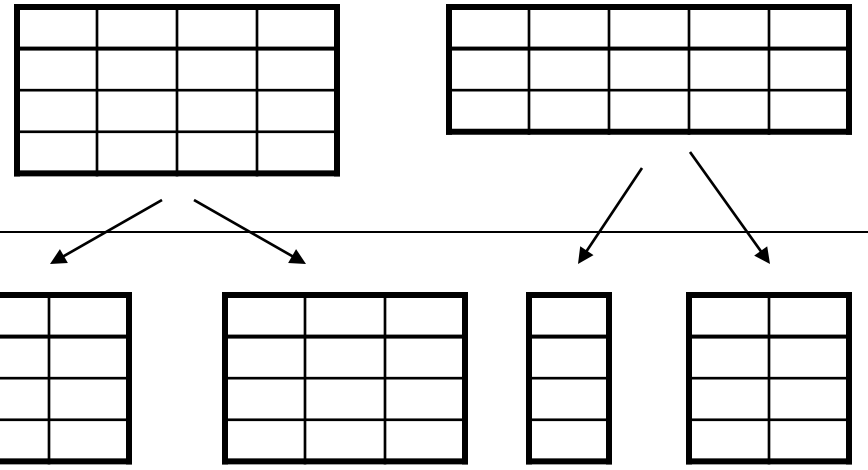
Conceptual Model:



Relational Model:

plus FD's

FD: Functional dependencies



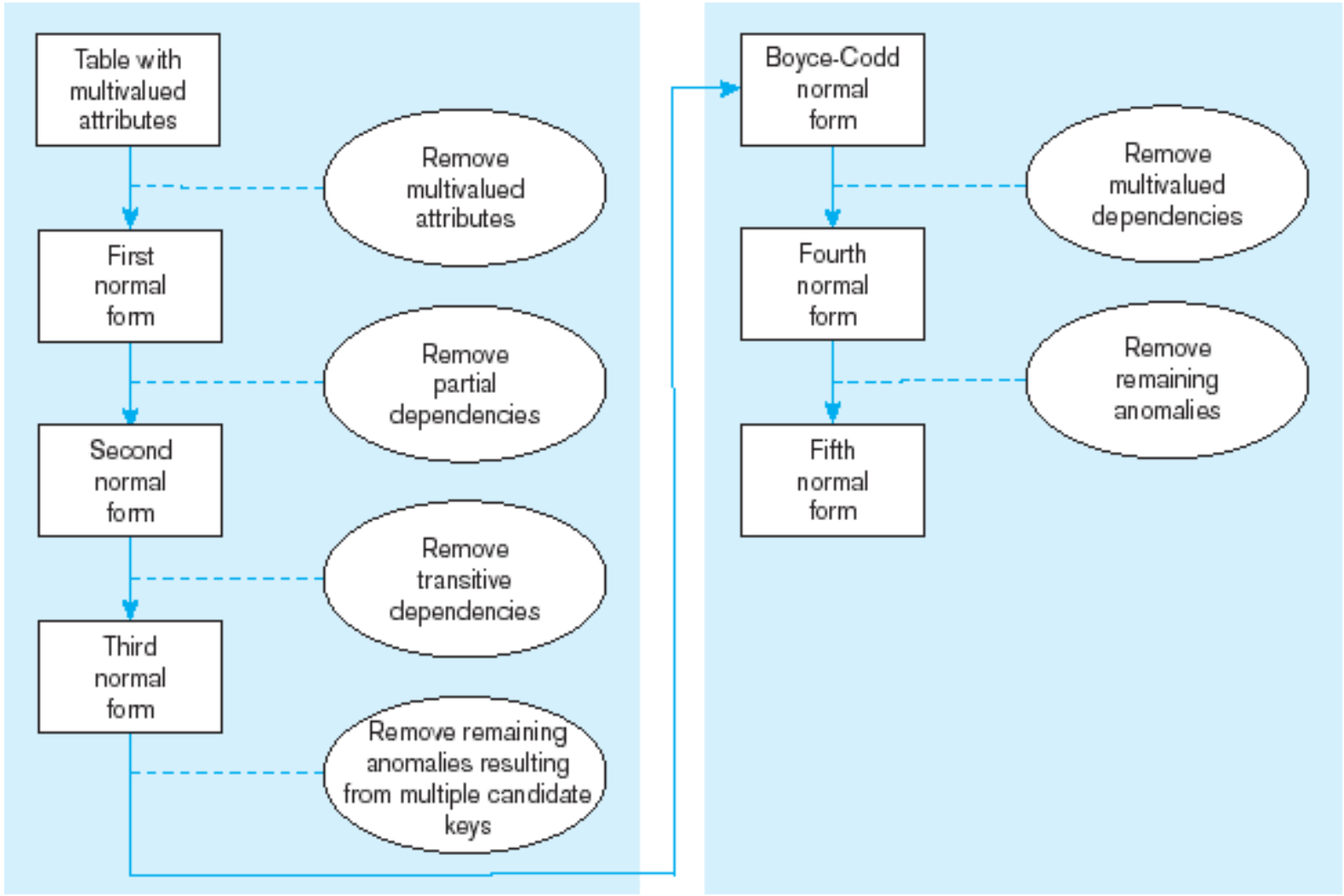
Normalization:

Eliminates ***anomalies***

Anomaly

- **Insertion Anomaly**–adding new rows forces user to create duplicate data
- **Deletion Anomaly**–deleting rows may cause a loss of data that would be needed for other future rows
- **Modification Anomaly**–changing data in a row forces changes to other rows because of duplication

Steps in normalization



Functional dependency

- a relationship between two attributes, typically between the PK and other non-key attributes within a table.

- **X ———–> Y**

- The left side of the above FD diagram is called the *determinant*, and the right side is the *dependent*. Here are a few examples.

Functional dependency (example)

SIN ———> Name, Address, Birthdate

SIN, Course ———> DateCompleted

ISBN ———> Title

First Normal Form

- No multivalued attributes
- Every attribute value is atomic
- ***All relations*** **are in 1st Normal Form**

Table with multivalued attributes, not in 1st normal form

Figure 5-25
INVOICE data (Pine Valley Furniture Company)

Order_ID	Order_ Date	Customer_ ID	Customer_ Name	Customer_ Address	Product_ID	Product_ Description	Product_ Finish	Unit_ Price	Ordered_ Quantity
1006	10/24/2006	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
					5	Writer's Desk	Cherry	325.00	2
					4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2006	6	Furniture Gallery	Boulder, CO	11	4-Dr Dresser	Oak	500.00	4
					4	Entertainment Center	Natural Maple	650.00	3

Note: this is NOT a relation(Not in first normal form)

Table with no multivalued attributes and unique rows, in 1st normal form

Order_ID	Order_ Date	Customer_ ID	Customer_ Name	Customer_ Address	Product_ID	Product_ Description	Product_ Finish	Unit_ Price	Ordered_ Quantity
1006	10/24/2006	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
1006	10/24/2006	2	Value Furniture	Plano, TX	5	Writer's Desk	Cherry	325.00	2
1006	10/24/2006	2	Value Furniture	Plano, TX	4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2006	6	Furniture Gallery	Boulder, CO	11	4-Dr Dresser	Oak	500.00	4
1007	10/25/2006	6	Furniture Gallery	Boulder, CO	4	Entertainment Center	Natural Maple	650.00	3

Figure 5-26
INVOICE relation (1NF) (Pine Valley Furniture Company)

Product_ID → Product_Description, Product_Finish, Unit_Price
Order_ID, Product_ID → Ordered_Quantity

Note: this is relation, but not a well-structured one (in 1st normal form)

Anomalies in this Table

- **Insertion**–if new product is ordered for order 1007 of existing customer, customer data must be re-entered, causing duplication
- **Deletion**–if we delete the Dining Table from Order 1006, we lose information concerning this item's finish and price
- **Update**–changing the price of product ID 4 requires update in several records

Why do these anomalies exist?

Because there are multiple themes (entity types) in one relation. This results in duplication and an unnecessary dependency between the entities

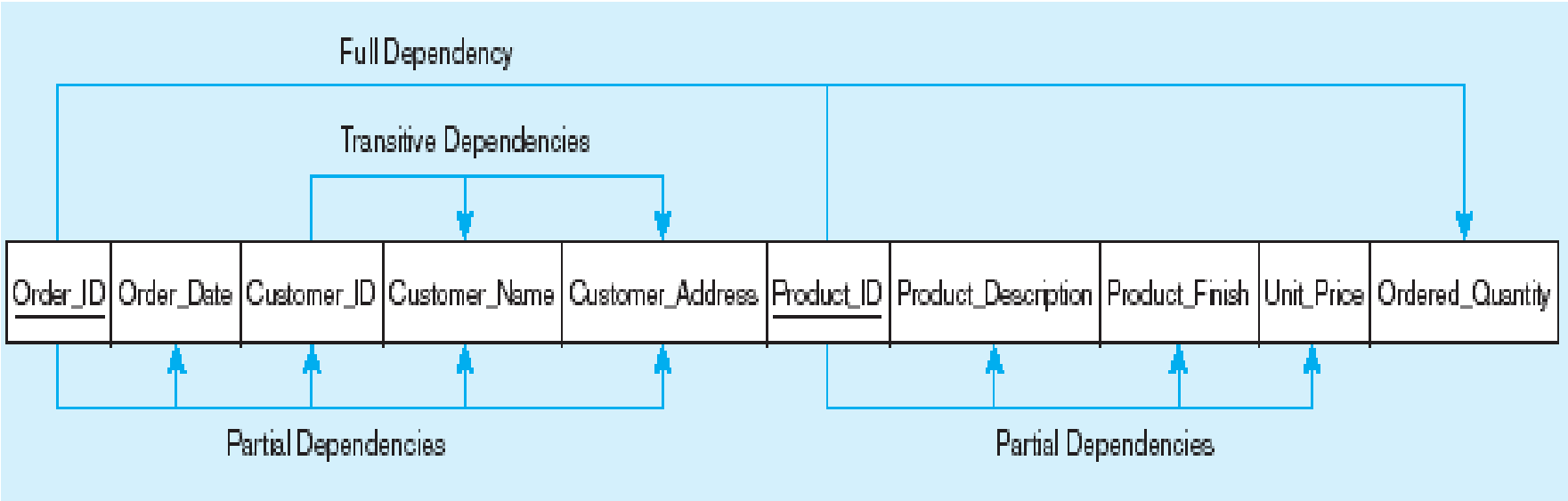
Second Normal Form

- 1NF PLUS ***every non-key attribute is fully functionally dependent on the ENTIRE primary key***
  - Every non-key attribute must be defined by the entire key, not by only part of the key
  - No partial functional dependencies

Example

- Seller(Id, Product, Price)
- **Candidate Key :** Id, Product
  Non prime attribute : Price
- **Price attribute only depends on only Product attribute** which is a subset of candidate key, **Not the whole candidate key(Id, Product) key** . It is called partial dependency.
- So we can say that **Product->Price** is partial dependency.

Functional dependency diagram for INVOICE



Order_ID ➔ Order_Date, Customer_ID, Customer_Name, Customer_Address

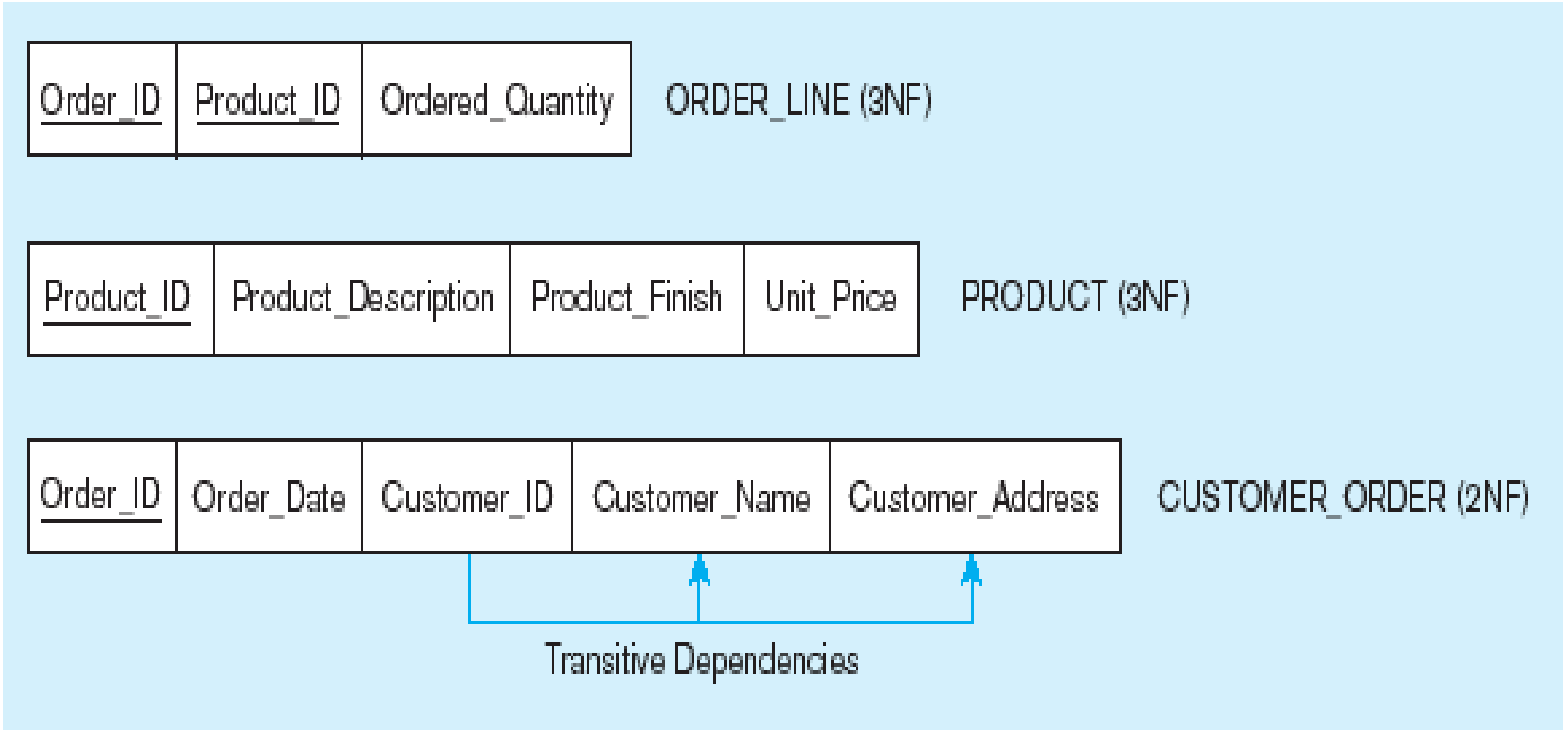
Customer_ID ➔ Customer_Name, Customer_Address

Product_ID ➔ Product_Description, Product_Finish, Unit_Price

Order_ID, Product_ID ➔ Order_Quantity

Therefore, NOT in 2nd Normal Form

Removing partial dependencies



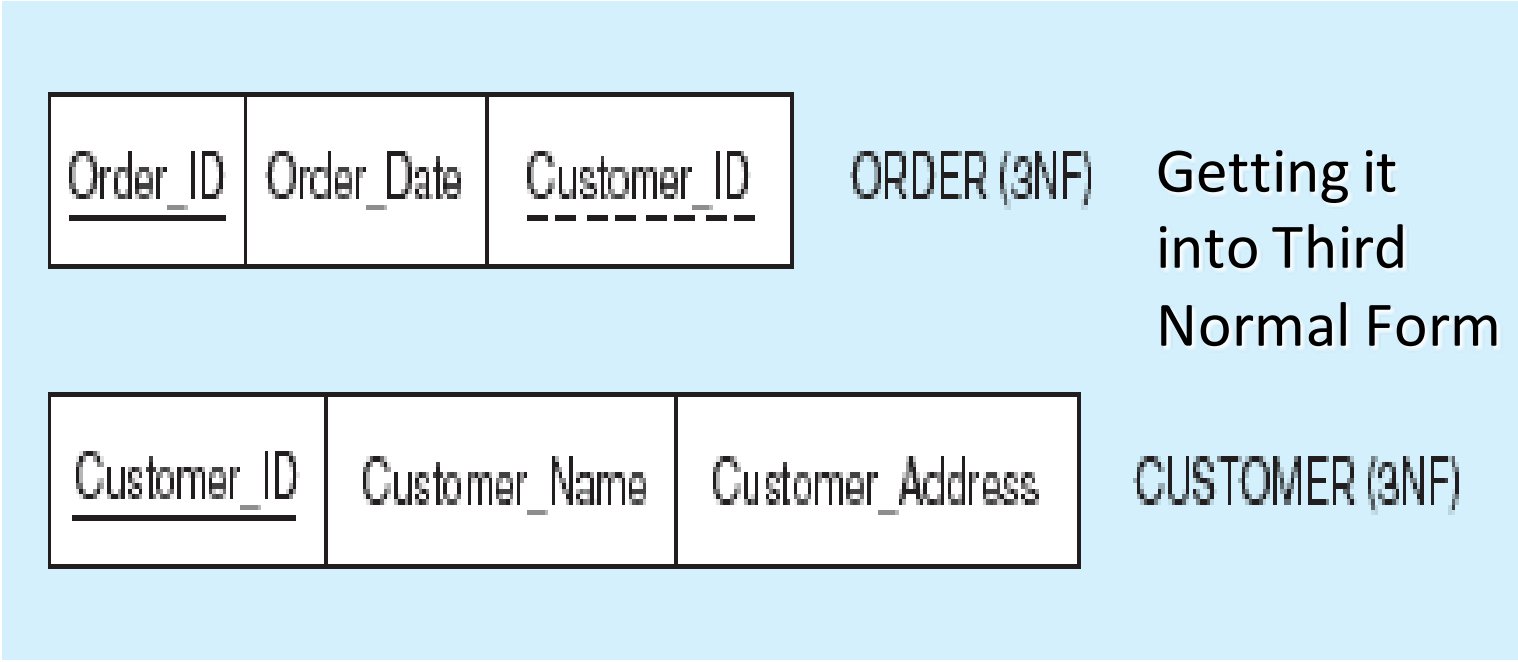
Getting it into Second Normal Form

Partial dependencies are removed, but there are still transitive dependencies

Third Normal Form

- 2NF PLUS ***no transitive dependencies*** (functional dependencies on non-primary-key attributes)
- Note: This is called transitive, because the primary key is a determinant for another attribute, which in turn is a determinant for a third
- Solution: Non-key determinant with transitive dependencies go into a new table; non-key determinant becomes primary key in the new table and stays as foreign key in the old table

Figure 5-28 Removing partial dependencies

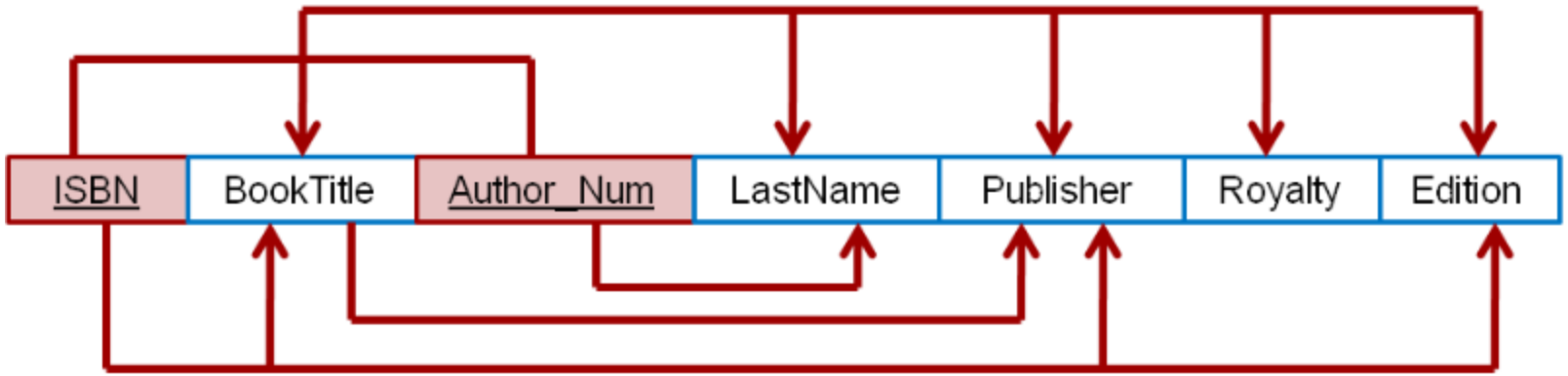


Transitive dependencies are removed

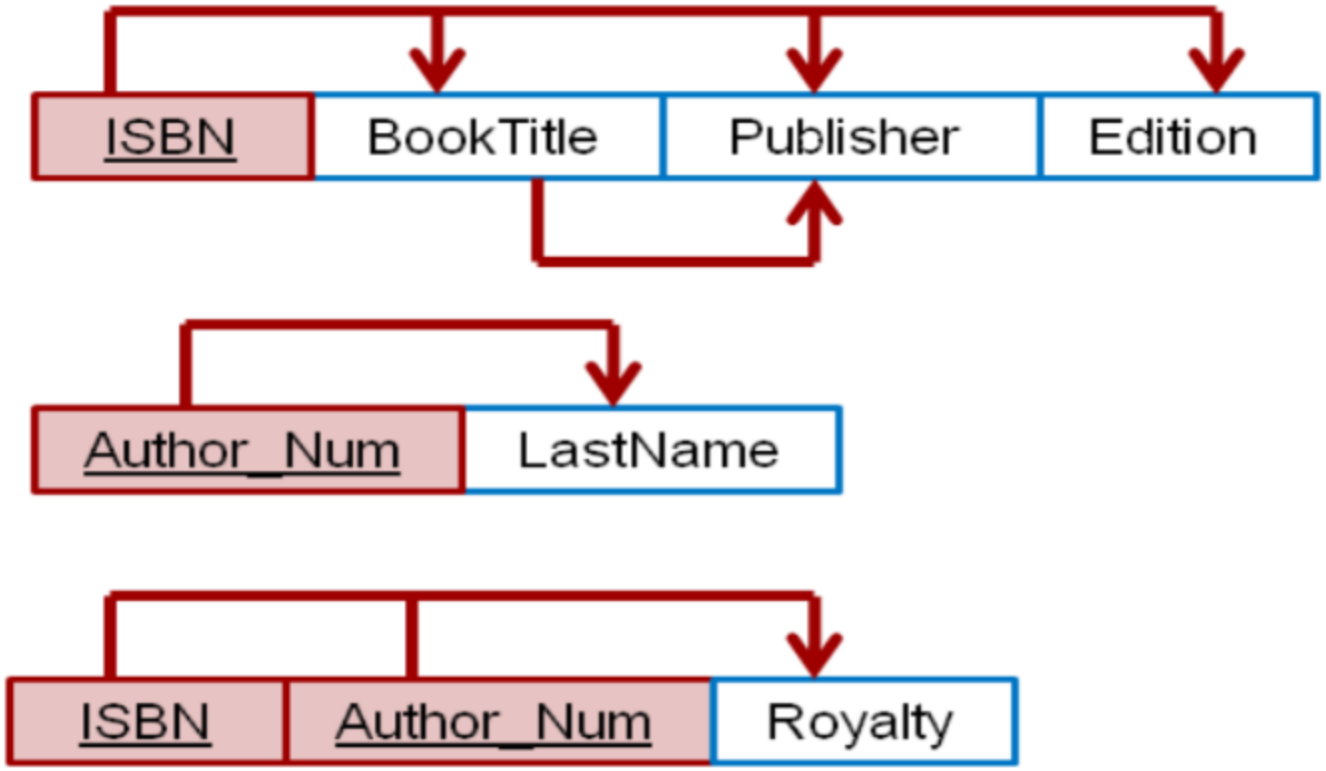
Example

The dependency diagram in Figure Q6.7 indicates that authors are paid royalties for each book that they write for a publisher. The amount of the royalty can vary by author, by book, and by edition of the book.

Figure Q6.7 Book royalty dependency diagram



Example


ISBN
BookTitle
Publisher
Edition
Author_Num
LastName
ISBN
Author_Num
Royalty