Database Normalization Process

Rahul Champaneria and Jessica Fung

Department of Computer Science, San Jose State University

CS 157A: Introduction to Database Management Systems

Dr. Ramin Moazeni

December 4th, 2024

User Create Table

Primary Key:

User_ID

Candidate Key:

User_ID

Normalized to 1NF:

User_ID → Name, Password, Phone, Role, Created_At, Updated_At, Email
All attributes are atomic and would not contain lists, only singular values. There are also no repeating columns.

Normalized to 2NF:

User_ID → Name, Password, Phone, Role, Created_At, Updated_At, Email
All the non-primary key attributes of the User entity are fully dependent on the primary key, therefore no partial dependencies are found. .

Normalized to 3NF:

User_ID → Name, Password, Phone, Role, Created_At, Updated_At
Email is removed from the table because it causes a determination of User_ID.
After removing Email, there are no functional dependencies.
There are no transitive dependencies.

Normalized to BCNF:

User_ID → Name, Password, Phone, Role, Created_At, Updated_At
User_ID is the only candidate key (determinant) and all other attributes are dependent on User_ID.
User_ID also cannot have duplicate values.

Additional Notes:

Despite being of the ENUM type, the Role attribute does not violate BCNF because it depends on the primary key (User_ID) for functionality.

UserEmail Create Table

Primary Key:

Email

Candidate Key:

Email

Normalized to 1NF:

Email → User_ID
All attributes are atomic and would not contain lists, only singular values. There are also no repeating columns.

Normalized to 2NF:

Email → User_ID
The non-primary key attribute of the UserEmail entity is fully dependent on the primary key, therefore no partial dependencies are found. .

Normalized to 3NF:

Email → User_ID
There are no transitive dependencies.

Normalized to BCNF:

Email → User_ID
Email is the candidate key (determinant) and User_ID is dependent on Email. Email cannot have duplicate values.

Additional Notes:

This table was created because the “email” attribute was taken out of the User table. Since Email is a unique attribute, it can be used to identify a user which is a functional dependency. This table was thus formed to normalize to BCNF.

Product Create Table

Primary Key:

Product_ID

Candidate Key:

Product_ID

Normalized to 1NF:

Product_ID, Seller_ID → Title, Description, Price, Quantity, Listing_Date, Status
All attributes are atomic and would not contain lists, only singular values. There are also no repeating columns.

Normalized to 2NF:

Product_ID → Title, Description, Price, Quantity, Listing_Date, Status
Seller_ID is removed because it causes a partial dependency on the composite key of Product_ID and Seller_ID. All the non-primary key attributes of the Product entity are fully dependent on the primary key, therefore no partial dependencies are found.

Normalized to 3NF:

Product_ID → Title, Description, Price, Quantity, Listing_Date, Status
There are no transitive dependencies.

Normalized to BCNF:

Product_ID → Title, Description, Price, Quantity, Listing_Date, Status
Product_ID is the candidate key (determinant) and all other attributes are dependent on Product_ID. Product_ID cannot have duplicate values.

Additional Notes:

ProductSeller Create Table

Primary Keys:

Product_ID and User_ID

Candidate Keys:

Product_ID, Seller_ID

Normalized to 1NF:

Product_ID, Seller_ID → (Composite Key)
All attributes are atomic and would not contain lists, only singular values. There are also no repeating columns.

Normalized to 2NF:

Product_ID, Seller_ID → (Composite Key)
Since both attributes are a composite key and a primary key, there are no partial dependencies.

Normalized to 3NF:

Product_ID, Seller_ID → (Composite Key)
There are no transitive dependencies.

Normalized to BCNF:

Product_ID, Seller_ID → (Composite Key)
Product_ID and Seller_ID are also the candidate keys (determinants). Since there are no other attributes, this is in BCNF.

Additional Notes:

Although the foreign keys create connections with the Product and User tables, they don't break BCNF or add any new functional requirements.
This table was created to ensure that duplicate entries are allowed (i.e. multiple products can be associated with a single seller).

Orders Create Table

Primary Key:

Order_ID

Candidate Key:

Order_ID

Un-Normalized:

Order_ID → User_ID, Total_Amount, Payment_Status, Shipping_Status, Order_Date, Product_ID, Quantity, Shipping_Address, Tracking_Number, Shipping_Method, Shipping_Date, Delivery_Date

Normalized to 1NF:

Order_ID → User_ID, Total_Amount, Payment_Status, Shipping_Status, Order_Date
Product_ID, Quantity, Shipping_Address, Tracking_Number, Shipping_Method, Shipping_Date, and Delivery_Date are removed from the table due to partial dependencies on the primary key and dependencies being made on non-primary keys.
All attributes are atomic and would not contain lists, only singular values. There are also no repeating columns.

Normalized to 2NF:

Order_ID → User_ID, Total_Amount, Payment_Status, Shipping_Status, Order_Date
All the non-primary key attributes of the Orders entity are fully dependent on the primary key, therefore no partial dependencies are found.

Normalized to 3NF:

Order_ID → Total_Amount, Payment_Status, Shipping_Status, Order_Date, User_ID
There are no transitive dependencies.

Normalized to BCNF:

Order_ID → Total_Amount, Payment_Status, Shipping_Status, Order_Date, User_ID

Order_ID is the candidate key (determinant) and all other attributes are dependent on Order_ID. Order_ID also cannot have duplicate values.

Additional Notes:

OrderItems Create Table

Primary Key:

Order_ID and Product_ID

Candidate Key:

Order_ID

Normalized to 1NF:

Order_ID, Product_ID → Quantity
All attributes are atomic and would not contain lists, only singular values. There are also no repeating columns.

Normalized to 2NF:

Order_ID, Product_ID → Quantity
All the non-primary key attributes of the User entity are fully dependent on the primary key.

Normalized to 3NF:

Order_ID, Product_ID → Quantity

Normalized to BCNF:

Order_ID, Product_ID → Quantity

Additional Notes:

This is a valid BCNF because there are no partial dependencies. Since Quantity and all other non-key properties rely solely on the composite primary key (Order_ID, Product_ID), there are no partial dependencies. The integrity of the primary key's dependency is preserved by the foreign key constraints connecting to the Orders and Product tables, which do not add any new dependencies. Additionally, non-key attributes do not rely on other non-key characteristics because there are no transitive dependencies.

ShoppingCart Create Table

Primary Key:

User_ID, Product_ID

Candidate Key:

User_ID, Product_ID

Normalized to 1NF:

User_ID, Product_ID → Quantity
All attributes are atomic and would not contain lists, only singular values. There are also no repeating columns.

Normalized to 2NF:

User_ID, Product_ID → Quantity
User_ID and Product_ID create a composite key and are both primary keys. Quantity (a non-primary key attribute) is dependent on both keys. There are no partial dependencies.

Normalized to 3NF:

User_ID, Product_ID → Quantity
There are no transitive dependencies.

Normalized to BCNF:

User_ID, Product_ID → Quantity
User_ID and Product_ID are the candidate keys (determinants) and quantity is dependent on both keys. They do not allow duplicate values.

Additional Notes:

Auction Create Table

Primary Key:

Auction_ID

Candidate Key:

Auction_ID

Normalized to 1NF:

Auction_ID → Starting_Price, End_Date, Product_ID, Highest_Bid
All attributes are atomic and would not contain lists, only singular values. There are also no repeating columns.

Normalized to 2NF:

Auction_ID → Starting_Price, End_Date, Product_ID, Highest_Bid
All the non-primary key attributes of the Auction entity are fully dependent on the primary key, therefore there are no partial dependencies.

Normalized to 3NF:

Auction_ID → Starting_Price, End_Date, Product_ID, Highest_Bid
There are no transitive dependencies.

Normalized to BCNF:

Auction_ID → Starting_Price, End_Date, Product_ID, Highest_Bid
Auction_ID is the candidate key (determinant) and all non-primary key attributes are dependent on Auction_ID. Auction_ID also does not allow duplicate values.

Additional Notes:

Bid Create Table

Primary Key:

Bid_ID

Candidate Key:

Bid_ID

Normalized to 1NF:

Bid_ID → Auction_ID, User_ID, Bid_Amount, Bid_Time, Current_HighestBid
All attributes are atomic and would not contain lists, only singular values. There are also no repeating columns.

Normalized to 2NF:

Bid_ID → Auction_ID, User_ID, Bid_Amount, Bid_Time, Current_HighestBid

All the non-primary key attributes of the Bid entity are fully dependent on the primary key, therefore there are no partial dependencies.

Normalized to 3NF:

Bid Table: Bid_ID → Auction_ID, User_ID, Bid_Amount, Bid_Time
Current_HighestBid was removed from the table because it created a transitive dependency through Auction_ID.
After the removal of Current_HighestBid, there are no transitive dependencies.

Normalized to BCNF:

Bid_ID → Auction_ID, User_ID, Bid_Amount, Bid_Time
Bid_ID is the candidate key (determinant) and all other attributes are dependent on Bid_ID. Bid_ID also does not allow duplicate values.

Additional Notes:

Notification Create Table

Primary Key:

Notification_ID

Candidate Key:

Notification_ID

Normalized to 1NF:

Notification_ID → User_ID, Message, Notification_Date, Status
All attributes are atomic and would not contain lists, only singular values. There are also no repeating columns.

Normalized to 2NF:

Notification_ID → User_ID, Message, Notification_Date, Status
All the non-primary key attributes of the Notification entity are fully dependent on the primary key, therefore there are no partial dependencies.

Normalized to 3NF:

Notification_ID → User_ID, Message, Notification_Date, Status
There are no transitive dependencies.

Normalized to BCNF:

Notification_ID → User_ID, Message, Notification_Date, Status

Notification_ID is the candidate key (determinant) and all other attributes are dependent on Notification_ID. Notification_ID also does not allow duplicate values.

Additional Notes:

Payment Create Table

Primary Key:

Payment_Order_ID

Candidate Key:

Payment_Order_ID

Normalized to 1NF:

Payment_Order_ID → Payment_Amount, Payment_Method, Payment_Date
All attributes are atomic and would not contain lists, only singular values. There are also no repeating columns.

Normalized to 2NF:

Payment_Order_ID → Payment_Amount, Payment_Method, Payment_Date
All the non-primary key attributes of the Payment entity are fully dependent on the primary key, therefore there are no partial dependencies.

Normalized to 3NF:

Payment_Order_ID → Payment_Amount, Payment_Method, Payment_Date
There are no transitive dependencies.

Normalized to BCNF:

Payment_Order_ID → Payment_Amount, Payment_Method, Payment_Date
Payment_Order_ID is the candidate key (determinant) and all other attributes are dependent on the candidate key.

Additional Notes:

Review Create Table

Primary Key:

Review_ID

Candidate Key:

Review_ID

Normalized to 1NF:

Review_ID → Order_ID, Review_Text, Review_Date, Rating
All attributes are atomic and would not contain lists, only singular values. There are also no repeating columns.

Normalized to 2NF:

Review_ID → Order_ID, Review_Text, Review_Date, Rating
All the non-primary key attributes of the Review entity are fully dependent on the primary key, therefore there are no partial dependencies.

Normalized to 3NF:

Review_ID → Order_ID, Review_Text, Review_Date, Rating
There are no transitive dependencies.

Normalized to BCNF:

Review_ID → Order_ID, Review_Text, Review_Date, Rating
Review_ID is the candidate key (determinant) and all other attributes are dependent on Review_ID. Review_ID also does not allow duplicate values.

Additional Notes:

OrderShipping Create Table

Primary Key:

Order_ID

Candidate Key:

Order_ID

Normalized to 1NF:

Order_ID → User_ID, Shipping_Address, Tracking_Number, Shipping_Method, Shipping_Date, Delivery_Date
All attributes are atomic and would not contain lists, only singular values. There are also no repeating columns.

Normalized to 2NF:

Order_ID → User_ID, Shipping_Address, Tracking_Number, Shipping_Method, Shipping_Date, Delivery_Date
All the non-primary key attributes of the OrderShipping entity are fully dependent on the primary key.

Normalized to 3NF:

Order_ID → Shipping_Address, Tracking_Number, Shipping_Method, Shipping_Date, Delivery_Date
There are no transitive dependencies.

Normalized to BCNF:

Order_ID → Shipping_Address, Tracking_Number, Shipping_Method, Shipping_Date, Delivery_Date
Order_ID is the candidate key (determinant) and all other attributes are dependent on Order_ID.

Additional Notes: