

## 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 4<sup>th</sup>, 2024

<b>User Create Table.....</b>	<b>5</b>
Primary Key:.....	5
Candidate Key:.....	5
Normalized to 1NF:.....	5
Normalized to 2NF:.....	5
Normalized to 3NF:.....	5
Normalized to BCNF:.....	5
Additional Notes:.....	5
<b>UserEmail Create Table.....</b>	<b>6</b>
Primary Key:.....	6
Candidate Key:.....	6
Normalized to 1NF:.....	6
Normalized to 2NF:.....	6
Normalized to 3NF:.....	6
Normalized to BCNF:.....	6
Additional Notes:.....	6
<b>Product Create Table.....</b>	<b>7</b>
Primary Key:.....	7
Candidate Key:.....	7
Normalized to 1NF:.....	7
Normalized to 2NF:.....	7
Normalized to 3NF:.....	7
Normalized to BCNF:.....	7
Additional Notes:.....	7
<b>ProductSeller Create Table.....</b>	<b>8</b>
Primary Keys:.....	8
Candidate Keys:.....	8
Normalized to 1NF:.....	8
Normalized to 2NF:.....	8
Normalized to 3NF:.....	8
Normalized to BCNF:.....	8
Additional Notes:.....	8
<b>Orders Create Table.....</b>	<b>9</b>
Primary Key:.....	9
Candidate Key:.....	9
Un-Normalized:.....	9
Normalized to 1NF:.....	9
Normalized to 2NF:.....	9

Normalized to 3NF:.....	9
Normalized to BCNF:.....	9
Additional Notes:.....	9
<b>OrderItems Create Table.....</b>	<b>10</b>
Primary Key:.....	10
Candidate Key:.....	10
Normalized to 1NF:.....	10
Normalized to 2NF:.....	10
Normalized to 3NF:.....	10
Normalized to BCNF:.....	10
Additional Notes:.....	10
<b>ShoppingCart Create Table.....</b>	<b>11</b>
Primary Key:.....	11
Candidate Key:.....	11
Normalized to 1NF:.....	11
Normalized to 2NF:.....	11
Normalized to 3NF:.....	11
Normalized to BCNF:.....	11
Additional Notes:.....	11
<b>Auction Create Table.....</b>	<b>12</b>
Primary Key:.....	12
Candidate Key:.....	12
Normalized to 1NF:.....	12
Normalized to 2NF:.....	12
Normalized to 3NF:.....	12
Normalized to BCNF:.....	12
Additional Notes:.....	12
<b>Bid Create Table.....</b>	<b>13</b>
Primary Key:.....	13
Candidate Key:.....	13
Normalized to 1NF:.....	13
Normalized to 2NF:.....	13
Normalized to 3NF:.....	13
Normalized to BCNF:.....	13
Additional Notes:.....	13
<b>Notification Create Table.....</b>	<b>14</b>
Primary Key:.....	14
Candidate Key:.....	14
Normalized to 1NF:.....	14
Normalized to 2NF:.....	14
Normalized to 3NF:.....	14

Normalized to BCNF:.....	14
Additional Notes:.....	14
<b>Payment Create Table.....</b>	<b>15</b>
Primary Key:.....	15
Candidate Key:.....	15
Normalized to 1NF:.....	15
Normalized to 2NF:.....	15
Normalized to 3NF:.....	15
Normalized to BCNF:.....	15
Additional Notes:.....	15
<b>Review Create Table.....</b>	<b>16</b>
Primary Key:.....	16
Candidate Key:.....	16
Normalized to 1NF:.....	16
Normalized to 2NF:.....	16
Normalized to 3NF:.....	16
Normalized to BCNF:.....	16
Additional Notes:.....	16
<b>OrderShipping Create Table.....</b>	<b>17</b>
Primary Key:.....	17
Candidate Key:.....	17
Normalized to 1NF:.....	17
Normalized to 2NF:.....	17
Normalized to 3NF:.....	17
Normalized to BCNF:.....	17
Additional Notes:.....	17

## 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  $\rightarrow$  User\_ID**

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

Normalized to 2NF:

**Email  $\rightarrow$  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  $\rightarrow$  User\_ID**

There are no transitive dependencies.

Normalized to BCNF:

**Email  $\rightarrow$  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  $\rightarrow$  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  $\rightarrow$  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  $\rightarrow$  Quantity**

There are no transitive dependencies.

Normalized to BCNF:

**User\_ID, Product\_ID  $\rightarrow$  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: