0.11 Candidate Keys Potential Issues

you-said

Delivery (OrderID, CustomerID, ProductID, ShippingID, PaymentID, DeliveryDate) FD = { ProductID, PaymentID} -> {OrderID} {ProductID} -> {ShippingID} {OrderID} -> {CustomerID} {ShippingID} -> {CustomerID, PaymentID} {CustomerID} -> {DeliveryDate} {OrderID, ShippingID} -> {ProductID, DeliveryDate} } apply this algorithm to the above set of functional dependencies: functional_dependencies (FD) of type -> for ith functional dependency. left = Union of all in FD right = Union of all in FD Initialize case_1, case_2, case_3, case_4 as empty sets. For x in attributes: if x left & x right: case_1.add(x) if x left & x right: case_2.add(x) if x left & x right: case_3.add(x) if x left & x right: case_4.add(x) return attributes in case_1, case_2, case_3, case_4

ChatGPT said:

The attributes are classified into the following cases:

- **Case 1 (attributes not in left and not in right)**: None
- **Case 2 (attributes not in left but in right)**: {DeliveryDate}
- **Case 3 (attributes in left but not in right)**: None
- **Case 4 (attributes in both left and right)**: {CustomerID, OrderID, PaymentID, ProductID, ShippingID}

you-said-1

use this algorithm completely for closure: Closure of under F (+) result := Do result_previous = result For each → in F: If (result): result := result Union While (result_previous <> result) Return result give closure of {CustomerID}, {OrderID}, {PaymentID}, {ProductID}, {ShippingID}

2/2

ChatGPT said:

The closure for each of the given sets under the functional dependencies is as follows:

- **Closure of {CustomerID}**: {CustomerID, DeliveryDate}
- **Closure of {OrderID}**: {OrderID, CustomerID, DeliveryDate}

- **Closure of {PaymentID}**: {PaymentID}
- **Closure of {ProductID}**: {ProductID, OrderID, PaymentID, ShippingID, CustomerID, DeliveryDate}
- **Closure of {ShippingID}**: {ShippingID, CustomerID, PaymentID, DeliveryDate}

you-said-2

using same algorithm give closures of {CustomerID, OrderID}, {CustomerID, PaymentID}, {CustomerID, ShippingID}, {OrderID, PaymentID}, {OrderID, ShippingID}, {PaymentID, ShippingID}

ChatGPT said:

It seems like I can't do more advanced data analysis right now. Please try again later. If you'd like, I can explain the closure algorithm further or help with another question!

you-said-3

use this algorithm completely for closure: Closure of under F (+) result := Do result_previous = result For each → in F: If (result): result := result Union While (result_previous <> result) Return result give closures of {CustomerID, OrderID}, {CustomerID, PaymentID}, {CustomerID, ShippingID}, {OrderID, PaymentID}, {OrderID, ShippingID}, {PaymentID, ShippingID}

ChatGPT said:

Let's apply the closure algorithm to each of the specified sets:

functional-dependencies-fd

Functional Dependencies (FD)

$$FD = \{\{\text{ProductID}, \text{PaymentID}\} \rightarrow \{\text{OrderID}\}, \{\text{ProductID}\} \rightarrow \{\text{ShippingID}\}, \{\text{OrderID}\} \rightarrow \{\text{CustomerID}\},$$

FD={{ProductID,PaymentID}→{OrderID},{ProductID}→{ShippingID},{OrderID}→{CustomerID},{Shipp

closure-of-customerid-orderid

Closure of {CustomerID, OrderID}

1. Start with $\{\text{CustomerID}, \text{OrderID}\}${CustomerID,OrderID}.
2. $\{\text{OrderID}\} \rightarrow \{\text{CustomerID}\}${OrderID}→{CustomerID} but already in the set.
3. $\{\text{CustomerID}\} \rightarrow \{\text{DeliveryDate}\}${CustomerID}→{DeliveryDate}, so add DeliveryDateDeliveryDate.

4. Final closure: {CustomerID, OrderID, DeliveryDate}{CustomerID,OrderID,DeliveryDate}.

closure-of-customerid-paymentid

Closure of {CustomerID, PaymentID}

1. Start with {CustomerID, PaymentID}{CustomerID,PaymentID}.
2. {CustomerID} → {DeliveryDate}{CustomerID}→{DeliveryDate}, so add DeliveryDateDeliveryDate.
3. No other functional dependencies apply.
4. Final closure: {CustomerID, PaymentID, DeliveryDate}{CustomerID,PaymentID,DeliveryDate}.

closure-of-customerid-shippingid

Closure of {CustomerID, ShippingID}

1. Start with {CustomerID, ShippingID}{CustomerID,ShippingID}.
2. {ShippingID} → {CustomerID, PaymentID}{ShippingID}→{CustomerID,PaymentID}, so add PaymentIDPaymentID.
3. {CustomerID} → {DeliveryDate}{CustomerID}→{DeliveryDate}, so add DeliveryDateDeliveryDate.
4. Final closure: {CustomerID, ShippingID, PaymentID, DeliveryDate}{CustomerID,ShippingID,PaymentI

closure-of-orderid-paymentid

Closure of {OrderID, PaymentID}

1. Start with {OrderID, PaymentID}{OrderID,PaymentID}.
2. {OrderID} → {CustomerID}{OrderID}→{CustomerID}, so add CustomerIDCustomerID.
3. {CustomerID} → {DeliveryDate}{CustomerID}→{DeliveryDate}, so add DeliveryDateDeliveryDate.
4. Final closure: {OrderID, PaymentID, CustomerID, DeliveryDate}{OrderID,PaymentID,CustomerID,Deli

closure-of-orderid-shippingid

cl

Closure of {OrderID, ShippingID}

1. Start with {OrderID, ShippingID}{OrderID,ShippingID}.
2. {OrderID, ShippingID} → {ProductID, DeliveryDate} {OrderID,ShippingID}→{ProductID,DeliveryDat
   so add ProductID, DeliveryDateProductID,DeliveryDate.
3. {OrderID} → {CustomerID} {OrderID}→{CustomerID}, so add CustomerIDCustomerID.
4. {ProductID} → {ShippingID} {ProductID}→{ShippingID} but already in the set.
5. Final closure: {OrderID, ShippingID, ProductID, CustomerID, DeliveryDate}{OrderID,ShippingID,Prod

closure-of-paymentid-shippingid

Closure of {PaymentID, ShippingID}

1. Start with {PaymentID, ShippingID}{PaymentID,ShippingID}.
2. {ShippingID} → {CustomerID, PaymentID} {ShippingID}→{CustomerID,PaymentID},
   so add CustomerIDCustomerID.
3. {CustomerID} → {DeliveryDate} {CustomerID}→{DeliveryDate}, so add DeliveryDateDeliveryDate.
4. Final closure: {PaymentID, ShippingID, CustomerID, DeliveryDate}{PaymentID,ShippingID,CustomerI

final-results

Final Results:

- Closure of {CustomerID, OrderID}: {CustomerID, OrderID, DeliveryDate}{CustomerID,OrderID,Delive
- Closure of {CustomerID, PaymentID}: {CustomerID, PaymentID, DeliveryDate}{CustomerID,PaymentI
- Closure of {CustomerID, ShippingID}: {CustomerID, ShippingID, PaymentID, DeliveryDate}{Customer
- Closure of {OrderID, PaymentID}: {OrderID, PaymentID, CustomerID, DeliveryDate}{OrderID,Paymen
- Closure of {OrderID, ShippingID}: {OrderID, ShippingID, ProductID, CustomerID, DeliveryDate}{Orde
- Closure of {PaymentID, ShippingID}: {PaymentID, ShippingID, CustomerID, DeliveryDate}{PaymentID

you-said-4

using algorithm explain Closure of {OrderID, ShippingID} step by step

ChatGPT said:

Let's go through the **closure of {OrderID, ShippingID}** step by step using the closure algorithm.

initial-setup

Initial Setup:

We start with:

$$\text{result} := \{\text{OrderID}, \text{ShippingID}\}$$

result:={OrderID,ShippingID}

This is the initial result set.

step-1-apply-the-functional-dependencies

Step 1: Apply the Functional Dependencies

We check each functional dependency (FD) and see if its left-hand side (γ) is a subset of the current result set. If it is, we add the right-hand side (δ) of that FD to the result.

1. **FD:** $\{\textbf{ProductID}, \textbf{PaymentID}\} \rightarrow \{\textbf{OrderID}\}$ **{ProductID,PaymentID}→{OrderID}**
    - Left side: $\{\text{ProductID}, \text{PaymentID}\}$ {ProductID,PaymentID}
    - $\{\text{ProductID}, \text{PaymentID}\}$ {ProductID,PaymentID} is not a subset of $\{\text{OrderID}, \text{ShippingID}\}$ {OrderI so nothing is added.
2. **FD:** $\{\textbf{ProductID}\} \rightarrow \{\textbf{ShippingID}\}$ **{ProductID}→{ShippingID}**
    - Left side: $\{\text{ProductID}\}$ {ProductID}
    - $\{\text{ProductID}\}$ {ProductID} is not a subset of $\{\text{OrderID}, \text{ShippingID}\}$ {OrderID,ShippingID}, so nothing is added.
3. **FD:** $\{\textbf{OrderID}\} \rightarrow \{\textbf{CustomerID}\}$ **{OrderID}→{CustomerID}**
    - Left side: $\{\text{OrderID}\}$ {OrderID}
    - $\{\text{OrderID}\}$ {OrderID} is a subset of $\{\text{OrderID}, \text{ShippingID}\}$ {OrderID,ShippingID}, so we add $\{\text{CustomerID}\}$ {CustomerID} to the result.
    - New result: $\{\text{OrderID}, \text{ShippingID}, \text{CustomerID}\}$ {OrderID,ShippingID,CustomerID}

4. **FD:** $\{\textbf{ShippingID}\} \rightarrow \{\textbf{CustomerID}, \textbf{PaymentID}\}$ **{ShippingID}→{CustomerID,PaymentID**
   - Left side: $\{\text{ShippingID}\}$\{ShippingID\}
   - $\{\text{ShippingID}\}$\{ShippingID\} is a subset of the current result, so we add $\{\text{CustomerID}, \text{PaymentID}\}$\{C to the result.
   - $\{\text{CustomerID}\}$\{CustomerID\} is already in the result, but $\{\text{PaymentID}\}$\{PaymentID\} is new.
   - New result: $\{\text{OrderID}, \text{ShippingID}, \text{CustomerID}, \text{PaymentID}\}$\{OrderID,ShippingID,CustomerID,Pa
5. **FD:** $\{\textbf{CustomerID}\} \rightarrow \{\textbf{DeliveryDate}\}$ **{CustomerID}→{DeliveryDate}**
   - Left side: $\{\text{CustomerID}\}$\{CustomerID\}
   - $\{\text{CustomerID}\}$\{CustomerID\} is a subset of the current result, so we add $\{\text{DeliveryDate}\}$\{DeliveryDa to the result.
   - New result: $\{\text{OrderID}, \text{ShippingID}, \text{CustomerID}, \text{PaymentID}, \text{DeliveryDate}\}$\{OrderID,ShippingID,C
6. **FD:** $\{\textbf{OrderID}, \textbf{ShippingID}\} \rightarrow \{\textbf{ProductID}, \textbf{DeliveryDate}\}$ **{OrderID,ShippingID}→{Produ**
   - Left side: $\{\text{OrderID}, \text{ShippingID}\}$\{OrderID,ShippingID\}
   - $\{\text{OrderID}, \text{ShippingID}\}$\{OrderID,ShippingID\} is a subset of the current result, so we add $\{\text{ProductID}, \text{DeliveryDate}\}$\{ProductID,DeliveryDate\}.
   - $\{\text{DeliveryDate}\}$\{DeliveryDate\} is already in the result, but $\{\text{ProductID}\}$\{ProductID\} is new.
   - New result: $\{\text{OrderID}, \text{ShippingID}, \text{CustomerID}, \text{PaymentID}, \text{DeliveryDate}, \text{ProductID}\}$\{OrderID,Sl

step-2-check-for-changes

Step 2: Check for Changes

At this point, the result has stabilized, and no new attributes can be added.
final-closure

Final Closure:

The closure of $\{\text{OrderID}, \text{ShippingID}\}$\{OrderID,ShippingID\} is:

$$\{\text{OrderID}, \text{ShippingID}, \text{CustomerID}, \text{PaymentID}, \text{DeliveryDate}, \text{ProductID}\}$$

{OrderID,ShippingID,CustomerID,PaymentID,DeliveryDate,ProductID}

you-said-5

use this algorithm completely for closure: Closure of under F (+) result := Do result_previous = result For each → in F: If (result): result := result Union While (result_previous <> result) Return result give closure of {CustomerID, OrderID, PaymentID}

ChatGPT said:

Let's walk through the closure of $\{\text{CustomerID}, \text{OrderID}, \text{PaymentID}\}${CustomerID,OrderID,PaymentID} using the given algorithm step by step.

functional-dependencies-fd-1

Functional Dependencies (FD)

$$\text{FD} = \{\{\text{ProductID}, \text{PaymentID}\} \rightarrow \{\text{OrderID}\}, \{\text{ProductID}\} \rightarrow \{\text{ShippingID}\}, \{\text{OrderID}\} \rightarrow \{\text{CustomerID}\},$$

FD={{ProductID,PaymentID}→{OrderID},{ProductID}→{ShippingID},{OrderID}→{CustomerID},{Shipp

initial-setup-1

Initial Setup:

We begin with:

$$\text{result} := \{\text{CustomerID}, \text{OrderID}, \text{PaymentID}\}$$

result:={CustomerID,OrderID,PaymentID}

step-1-apply-the-functional-dependencies-1

Step 1: Apply the Functional Dependencies

Now, we apply each functional dependency (FD) and check whether the left-hand side (γ) is a subset of the current result set. If it is, we add the right-hand side (δ) to the result.

1. **FD: $\{\textbf{ProductID}, \textbf{PaymentID}\} \rightarrow \{\textbf{OrderID}\}$ {ProductID,PaymentID}→{OrderID}**
    - Left side: $\{\text{ProductID}, \text{PaymentID}\}${ProductID,PaymentID}
    - $\{\text{ProductID}, \text{PaymentID}\}${ProductID,PaymentID} is not a subset of $\{\text{CustomerID}, \text{OrderID}, \text{Payme}$ so nothing is added.
2. **FD: $\{\textbf{ProductID}\} \rightarrow \{\textbf{ShippingID}\}$ {ProductID}→{ShippingID}**
    - Left side: $\{\text{ProductID}\}${ProductID}

- {ProductID}{ProductID} is not a subset of {CustomerID, OrderID, PaymentID}{CustomerID,Orde
  so nothing is added.

3. **FD: {OrderID} → {CustomerID}{OrderID}→{CustomerID}**

   - Left side: {OrderID}{OrderID}
   - {OrderID}{OrderID} is a subset of {CustomerID, OrderID, PaymentID}{CustomerID,OrderID,Pay
     but {CustomerID}{CustomerID} is already in the result.
   - No change in the result set.

4. **FD: {ShippingID} → {CustomerID, PaymentID}{ShippingID}→{CustomerID,PaymentID**

   - Left side: {ShippingID}{ShippingID}
   - {ShippingID}{ShippingID} is not a subset of {CustomerID, OrderID, PaymentID}{CustomerID,Ord
     so nothing is added.

5. **FD: {CustomerID} → {DeliveryDate}{CustomerID}→{DeliveryDate}**

   - Left side: {CustomerID}{CustomerID}
   - {CustomerID}{CustomerID} is a subset of {CustomerID, OrderID, PaymentID}{CustomerID,Orde
     so we add {DeliveryDate}{DeliveryDate} to the result.
   - New result: {CustomerID, OrderID, PaymentID, DeliveryDate}{CustomerID,OrderID,PaymentID,I

6. **FD: {OrderID, ShippingID} → {ProductID, DeliveryDate}{OrderID,ShippingID}→{Produ**

   - Left side: {OrderID, ShippingID}{OrderID,ShippingID}
   - {OrderID, ShippingID}{OrderID,ShippingID} is not a subset of the current result, so nothing is added.

step-2-check-for-changes-1

Step 2: Check for Changes

At this point, no more changes can be made to the result set, as no other functional dependency applies.

final-closure-1

Final Closure:

The closure of $\{\text{CustomerID}, \text{OrderID}, \text{PaymentID}\}${CustomerID,OrderID,PaymentID} is:

$$\{\text{CustomerID}, \text{OrderID}, \text{PaymentID}, \text{DeliveryDate}\}$$

{CustomerID,OrderID,PaymentID,DeliveryDate}

You've hit the Free plan limit for GPT-4o.

Responses will use another model until your limit resets after 11:08 PM.

Get Plus

prompt-textarea

composer-background