0.16 Candidate Keys Identification

you-said

use this function to find candidate keys in Relation(Alice, Bruce, Charlie, Dave, Elle) ; F={{Alice} -> {Bruce, Charlie}, {Charlie, Dave} -> {Elle}, {Bruce} -> {Dave}, {Elle} -> {Alice}}. def find_candidate_key(all_attributes, relations, all_left, all_right): case_1 = [] case_2 = [] case_3 = [] case_4 = [] for attribute in all_attributes: if attribute in all_left and attribute in all_right: case_4.append(attribute) elif attribute in all_left: case_3.append(attribute) elif attribute in all_right: case_2.append(attribute) else: case_1.append(attribute) case_1_3 = sorted(case_1 + case_3) case_4 = sorted(case_4) if validate_first(case_1_3, relations, all_attributes): return [case_1_3] return sorted(additional_checks(case_1_3, case_4, relations, all_attributes)) def additional_checks(case_1_3, case_4, relations, all_attributes): cks = [] for i in range(len(case_4)): for subset in list(combinations(case_4, i + 1)): if validate_first(sorted(list(subset) + case_1_3), relations, all_attributes): cks.append(sorted(list(subset) + case_1_3)) return find_minimal(cks) # check closure def validate_first(beta, relations, all_attributes): result = set(beta) previous_result = set() while previous_result != result: previous_result = result.copy() for i in range(len(result)): for subset in list(combinations(result, i + 1)): if tuple(sorted(subset)) in relations: result.update(relations[tuple(sorted(subset))]) if set(result) == set(all_attributes): return True return False

ChatGPT said:

The candidate keys for the relation $R(Alice, Bruce, Charlie, Dave, Elle)$R(Alice,Bruce,Charlie,Dave,Elle) with the given functional dependencies are:

1. $\{Alice\}$\{Alice\}
2. $\{Bruce, Charlie\}$\{Bruce,Charlie\}
3. $\{Charlie, Dave\}$\{Charlie,Dave\}
4. $\{Elle\}$\{Elle\}

Each of these sets can determine all other attributes in the relation.

you-said-1

what will be the candidate keys for: r2(A,B ,C, D,E ,G ,H ,K) ; F = {{A, B,H}=>{ C}, {A}=>{D,E }, {B,G,H} => {K}, {K}=> {A,D,H}, {B,H}=>{G, E }.

ChatGPT said:

The candidate keys for the relation $r2(A, B, C, D, E, G, H, K)$r2(A,B,C,D,E,G,H,K) with the given functional dependencies are:

1. $\{B, H\}$\{B,H\}
2. $\{B, K\}$\{B,K\}

Both of these sets can determine all other attributes in the relation.
you-said-2

explain this briefly
ChatGPT said:
In the relation $r2(A, B, C, D, E, G, H, K)$r2(A,B,C,D,E,G,H,K), we are tasked with finding sets of attributes (candidate keys) that can uniquely identify all other attributes using the given functional dependencies (FDs):

1. $\{A, B, H\} \rightarrow \{C\}$ \{A,B,H\}→\{C\}
2. $\{A\} \rightarrow \{D, E\}$ \{A\}→\{D,E\}
3. $\{B, G, H\} \rightarrow \{K\}$ \{B,G,H\}→\{K\}
4. $\{K\} \rightarrow \{A, D, H\}$ \{K\}→\{A,D,H\}
5. $\{B, H\} \rightarrow \{G, E\}$ \{B,H\}→\{G,E\}

The candidate keys must be minimal sets of attributes from which we can derive all other attributes in the relation. After analyzing the FDs, we find that:

- $\{B, H\}$\{B,H\} can determine all attributes: $\{B, H\} \rightarrow \{G, E\}, \{B, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D, H\}, \{A, B,$
- $\{B, K\}$\{B,K\} can also determine all attributes: $\{B, K\} \rightarrow \{A, D, H\}, \{B, H\} \rightarrow \{G, E\}, \{A, B, H\} \rightarrow \{$

Thus, $\{B, H\}$\{B,H\} and $\{B, K\}$\{B,K\} are the candidate keys.
you-said-3

using this sudo code: def find_candidate_key(all_attributes, relations, all_left, all_right): case_1 = [] case_2 = [] case_3 = [] case_4 = [] for attribute in all_attributes: if attribute in all_left and attribute in all_right: case_4.append(attribute) elif attribute in all_left: case_3.append(attribute) elif attribute in all_right: case_2.append(attribute) else: case_1.append(attribute) if closure of (case_1 + case_3) is all_attributes: return [case_1 + case_3] candidate_keys = [] for all possible combinations of case_4 as x: if closure of (x+case_1+case+3) is all_attributes: candidate_keys.append(all x+case_1+case+3) Return candidate_keys find the candidate keys of Relation users (userid, name, deptid, deptname); with functional dependencies FD = {{userid} => {name}, {userid} =>{deptid}, {deptid}=>{deptname}}.
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 walk you through how to solve this manually or attempt the task again another time.

you-said-4

using this sudo code: def find_candidate_key(all_attributes, relations, all_left, all_right): case_1 = [] case_2 = [] case_3 = [] case_4 = [] for attribute in all_attributes: if attribute in all_left and attribute in all_right: case_4.append(attribute) elif attribute in all_left: case_3.append(attribute) elif attribute in all_right: case_2.append(attribute) else: case_1.append(attribute) Provide me the attributes that will be in each case for: Relation Shopping(Egg, Dress, Chocolate, Basket, Apple); Functional dependencies FD = {{Apple, Basket, Chocolate} -> {Dress}, {Apple, Basket, Chocolate} -> {Dress, Egg}, {Dress } -> { Apple , Basket}, {Egg} -> { Apple , Chocolate }}

ChatGPT said:

Let's break down the problem by applying the pseudocode to the relation and functional dependencies provided.

relation-shoppingegg-dress-chocolate-basket-apple

Relation: Shopping(Egg, Dress, Chocolate, Basket, Apple)

We need to place the attributes into four different cases:

1. **Case 1:** Attributes that do not appear on either side of any functional dependencies.
2. **Case 2:** Attributes that only appear on the right side of some functional dependencies.
3. **Case 3:** Attributes that only appear on the left side of some functional dependencies.
4. **Case 4:** Attributes that appear on both the left and right sides of functional dependencies.

functional-dependencies-fd

Functional Dependencies (FD):

1. $\{Apple, Basket, Chocolate\} \rightarrow \{Dress\}$ {Apple,Basket,Chocolate}→{Dress}
2. $\{Apple, Basket, Chocolate\} \rightarrow \{Dress, Egg\}$ {Apple,Basket,Chocolate}→{Dress,Egg}
3. $\{Dress\} \rightarrow \{Apple, Basket\}$ {Dress}→{Apple,Basket}
4. $\{Egg\} \rightarrow \{Apple, Chocolate\}$ {Egg}→{Apple,Chocolate}

step-by-step-categorization

Step-by-Step Categorization:

attributes-in-the-left-hand-side-of-fds

- $\{Apple, Basket, Chocolate\}$ {Apple,Basket,Chocolate}
- $\{Dress\}$ {Dress}
- $\{Egg\}$ {Egg}

All Left Attributes: $\{Apple, Basket, Chocolate, Dress, Egg\}$ {Apple,Basket,Chocolate,Dress,Egg}
attributes-in-the-right-hand-side-of-fds

- $\{Dress\}$ {Dress}
- $\{Dress, Egg\}$ {Dress,Egg}
- $\{Apple, Basket\}$ {Apple,Basket}
- $\{Apple, Chocolate\}$ {Apple,Chocolate}

All Right Attributes: $\{Dress, Egg, Apple, Basket, Chocolate\}$ {Dress,Egg,Apple,Basket,Chocolate}
case-assignment

Case Assignment:

1. **Case 1 (Neither Left nor Right):**
   - There are no attributes that are absent from both the left and right sides of FDs. So, **Case 1** = [].
2. **Case 2 (Only in Right):**
   - No attributes only appear on the right side of FDs without also being on the left. So, **Case 2** = [].
3. **Case 3 (Only in Left):**

  - There are no attributes that only appear on the left side without also being on the right. So, **Case 3** = [].

4. **Case 4 (Both in Left and Right):**

  - Every attribute appears on both the left and right sides of some functional dependencies.
  - **Case 4** = [**Apple, Basket, Chocolate, Dress, Egg**].

summary-of-the-cases

Summary of the Cases:

- **Case 1:** (empty)
- **Case 2:** (empty)
- **Case 3:** (empty)
- **Case 4:** $\{Apple, Basket, Chocolate, Dress, Egg\}$ {Apple,Basket,Chocolate,Dress,Egg}

you-said-5

use this algorithm on the given cases: candidate_keys = [] for all possible subsets of case_4 as x: if closure of (x) is Relation: candidate_keys.append(x) Return candidate_keys

ChatGPT said:

We will apply the algorithm step by step to find the **candidate keys** for the relation $Shopping(Egg, Dress, Chocolate, Basket, Apple)$Shopping(Egg,Dress,Chocolate,Basket,Apple) with the following functional dependencies:

functional-dependencies-fd-1

Functional Dependencies (FD):

1. $\{Apple, Basket, Chocolate\} \rightarrow \{Dress\}$ {Apple,Basket,Chocolate}→{Dress}
2. $\{Apple, Basket, Chocolate\} \rightarrow \{Dress, Egg\}$ {Apple,Basket,Chocolate}→{Dress,Egg}
3. $\{Dress\} \rightarrow \{Apple, Basket\}$ {Dress}→{Apple,Basket}
4. $\{Egg\} \rightarrow \{Apple, Chocolate\}$ {Egg}→{Apple,Chocolate}

step-1-attributes

Step 1: Attributes

The relation has the following attributes: $\{Egg, Dress, Chocolate, Basket, Apple\}${Egg,Dress,Chocolate,Ba
From the previous analysis:

- **Case 4 (attributes that are both on the left and right sides of FDs):** $\{Apple, Basket, Chocolate, Dress, Egg\}${Apple,Basket,Chocolate,Dress,Egg}.

step-2-algorithm-application

Step 2: Algorithm Application

We'll check all possible subsets of **Case 4** (which includes all the attributes), and for each subset, we will calculate its closure. If the closure of a subset contains all the attributes in the relation, that subset is a candidate key.

subsets-of-case-4

We need to check the closure of the following subsets:

- $\{Apple\}${Apple}
- $\{Basket\}${Basket}
- $\{Chocolate\}${Chocolate}
- $\{Dress\}${Dress}
- $\{Egg\}${Egg}
- $\{Apple, Basket\}${Apple,Basket}
- $\{Apple, Chocolate\}${Apple,Chocolate}
- $\{Apple, Dress\}${Apple,Dress}
- $\{Apple, Egg\}${Apple,Egg}
- $\{Basket, Chocolate\}${Basket,Chocolate}
- $\{Basket, Dress\}${Basket,Dress}
- $\{Basket, Egg\}${Basket,Egg}
- $\{Chocolate, Dress\}${Chocolate,Dress}
- $\{Chocolate, Egg\}${Chocolate,Egg}
- $\ldots\ldots$ (and so on, for all subsets)

step-3-calculating-closures

Step 3: Calculating Closures

Now, we will calculate the closure of each subset to check if it can determine all the attributes of the relation (i.e., $\{Egg, Dress, Chocolate, Basket, Apple\}${Egg,Dress,Chocolate,Basket,Apple}).

1. **Closure of $\{Apple, Basket, Chocolate\}${Apple,Basket,Chocolate}:**

    - From FD 1 and FD 2, $\{Apple, Basket, Chocolate\} \rightarrow \{Dress, Egg\}${Apple,Basket,Chocolate}→{D
    - Now, we have $\{Apple, Basket, Chocolate, Dress, Egg\}${Apple,Basket,Chocolate,Dress,Egg}, which is all the attributes.
    - **So, $\{Apple, Basket, Chocolate\}${Apple,Basket,Chocolate} is a candidate key.**

2. **Closure of $\{Egg\}${Egg}:**

    - From FD 4, $\{Egg\} \rightarrow \{Apple, Chocolate\}${Egg}→{Apple,Chocolate}.
    - From FD 1 and FD 2, $\{Apple, Basket, Chocolate\} \rightarrow \{Dress, Egg\}${Apple,Basket,Chocolate}→{D
    - So, $\{Egg\} \rightarrow \{Apple, Basket, Chocolate, Dress, Egg\}${Egg}→{Apple,Basket,Chocolate,Dress,Egg which is all the attributes.
    - **So, $\{Egg\}${Egg} is a candidate key.**

step-4-conclusion

Step 4: Conclusion

The candidate keys for the relation are:

- $\{Apple, Basket, Chocolate\}${Apple,Basket,Chocolate}
- $\{Egg\}${Egg}

Both of these sets can uniquely determine all other attributes in the relation.
you-said-6

for FDs: {Apple,Basket,Chocolate}→{Dress} {Apple,Basket,Chocolate}→{Dress,Egg} {Dress}→{Apple,Basket} {Egg}→{Apple,Chocolate} Give me Closure of {Egg}
ChatGPT said:
To calculate the closure of $\{Egg\}${Egg}, we use the given functional dependencies and check which attributes can be determined starting from $\{Egg\}${Egg}.
functional-dependencies

Functional Dependencies:

1. $\{Apple, Basket, Chocolate\} \to \{Dress\}$ {Apple,Basket,Chocolate}→{Dress}
2. $\{Apple, Basket, Chocolate\} \to \{Dress, Egg\}$ {Apple,Basket,Chocolate}→{Dress,Egg}
3. $\{Dress\} \to \{Apple, Basket\}$ {Dress}→{Apple,Basket}
4. $\{Egg\} \to \{Apple, Chocolate\}$ {Egg}→{Apple,Chocolate}

step-by-step-closure-calculation-for-eggegg

Step-by-Step Closure Calculation for $\{Egg\}$**{Egg}:Step-by-Step Closure Calculation for \{ Egg\}{Egg}:**

1. **Start with** $\{Egg\}$**{Egg}:**
    - Initial closure: $\{Egg\}$ {Egg}
2. **Apply FD 4:**
    - $\{Egg\} \to \{Apple, Chocolate\}$ {Egg}→{Apple,Chocolate}
    - Closure now: $\{Egg, Apple, Chocolate\}$ {Egg,Apple,Chocolate}
3. **Check FD 1 and FD 2:**
    - With $\{Apple, Chocolate\}$ {Apple,Chocolate} in the closure, FD 1 or FD 2 applies if we also have $\{Basket\}$ {Basket}.
    - However, at this stage, $\{Basket\}$ {Basket} is not in the closure, so FD 1 and FD 2 cannot be applied yet.
4. **No further FDs can be applied at this point.**

final-closure-of-eggegg

Final Closure of $\{Egg\}$**{Egg}**:Final Closure of \\{ Egg\\}{Egg}:

$$\{Egg, Apple, Chocolate\}$$

{Egg,Apple,Chocolate}

The closure of $\{Egg\}${Egg} includes $\{Egg, Apple, Chocolate\}${Egg,Apple,Chocolate}.

you-said-7

Give me Closure of {Apple}

ChatGPT said:

Let's calculate the **closure of** $\{Apple\}$**{Apple}** using the given functional dependencies.

functional-dependencies-1

Functional Dependencies:

1. $\{Apple, Basket, Chocolate\} \to \{Dress\}${Apple,Basket,Chocolate}→{Dress}
2. $\{Apple, Basket, Chocolate\} \to \{Dress, Egg\}${Apple,Basket,Chocolate}→{Dress,Egg}
3. $\{Dress\} \to \{Apple, Basket\}${Dress}→{Apple,Basket}
4. $\{Egg\} \to \{Apple, Chocolate\}${Egg}→{Apple,Chocolate}

step-by-step-closure-calculation-for-appleapple

Step-by-Step Closure Calculation for $\{Apple\}$**{Apple}**:Step-by-Step Closure Calculation for \\{ Apple\\}{Apple}:

1. **Start with** $\{Apple\}$**{Apple}:**
   - Initial closure: $\{Apple\}${Apple}
2. **Check each FD:**
   - None of the FDs can be directly applied with just $\{Apple\}${Apple}, because each FD requires more attributes on the left-hand side (like Basket, Chocolate, etc.).
3. **No further FDs apply at this point.**

final-closure-of-appleapple

Final Closure of $\{Apple\}$**{Apple}**:Final Closure of \\{ Apple\\}{Apple}:

$$\{Apple\}$$

{Apple}

Since $\{Apple\}$\{Apple\} alone doesn't determine any other attributes based on the given FDs, the closure of $\{Apple\}$\{Apple\} is simply $\{Apple\}$\{Apple\}.

you-said-8

Give me Closure of {Basket}

ChatGPT said:

Let's calculate the **closure of** $\{Basket\}$**{Basket}** using the given functional dependencies.

functional-dependencies-2

Functional Dependencies:

1. $\{Apple, Basket, Chocolate\} \rightarrow \{Dress\}$ {Apple,Basket,Chocolate}→{Dress}
2. $\{Apple, Basket, Chocolate\} \rightarrow \{Dress, Egg\}$ {Apple,Basket,Chocolate}→{Dress,Egg}
3. $\{Dress\} \rightarrow \{Apple, Basket\}$ {Dress}→{Apple,Basket}
4. $\{Egg\} \rightarrow \{Apple, Chocolate\}$ {Egg}→{Apple,Chocolate}

step-by-step-closure-calculation-for-basketbasket

Step-by-Step Closure Calculation for $\{Basket\}$**{Basket}**:Step-by-Step Closure Calculation for \\{ Basket\\}{Basket}:

1. **Start with $\{Basket\}${Basket}:**
   - Initial closure: $\{Basket\}$\{Basket\}
2. **Check each FD:**
   - None of the FDs can be directly applied with just $\{Basket\}$\{Basket\}, because each FD requires more attributes on the left-hand side (like Apple, Chocolate, etc.).
3. **No further FDs apply at this point.**

final-closure-of-basketbasket

Final Closure of $\{Basket\}$**{Basket}**:Final Closure of \{ Basket\}{Basket}:

$$\{Basket\}$$

{Basket}

Since $\{Basket\}${Basket} alone doesn't determine any other attributes based on the given FDs, the closure of $\{Basket\}${Basket} is simply $\{Basket\}${Basket}.

you-said-9

Give me Closure of {Chocolate}

ChatGPT said:

Let's calculate the **closure of** $\{Chocolate\}$**{Chocolate}** using the given functional dependencies.

functional-dependencies-3

Functional Dependencies:

1. $\{Apple, Basket, Chocolate\} \to \{Dress\}$ {Apple,Basket,Chocolate}→{Dress}
2. $\{Apple, Basket, Chocolate\} \to \{Dress, Egg\}$ {Apple,Basket,Chocolate}→{Dress,Egg}
3. $\{Dress\} \to \{Apple, Basket\}$ {Dress}→{Apple,Basket}
4. $\{Egg\} \to \{Apple, Chocolate\}$ {Egg}→{Apple,Chocolate}

step-by-step-closure-calculation-for-chocolatechocolate

Step-by-Step Closure Calculation for $\{Chocolate\}$**{Chocolate}**:Step-by-Step Closure Calculation for \{ Chocolate\}{Chocolate}:

1. **Start with** $\{Chocolate\}$**{Chocolate}:**
    - Initial closure: $\{Chocolate\}${Chocolate}
2. **Check each FD:**
    - None of the FDs can be directly applied with just $\{Chocolate\}${Chocolate}, because they require additional attributes (like Apple, Basket, etc.).
3. **No further FDs apply at this point.**

final-closure-of-chocolatechocolate

Final Closure of $\{Chocolate\}$**{Chocolate}:Final Closure of \\{ Chocolate\\}{Chocolate}:**

$$\{Chocolate\}$$

{Chocolate}

Since $\{Chocolate\}${Chocolate} alone doesn't determine any other attributes based on the given FDs, the closure of $\{Chocolate\}${Chocolate} is simply $\{Chocolate\}${Chocolate}.

you-said-10

Give me Closure of {Dress}

ChatGPT said:

Let's calculate the **closure of** $\{Dress\}$**{Dress}** using the given functional dependencies.

functional-dependencies-4

Functional Dependencies:

1. $\{Apple, Basket, Chocolate\} \rightarrow \{Dress\}$ {Apple,Basket,Chocolate}→{Dress}
2. $\{Apple, Basket, Chocolate\} \rightarrow \{Dress, Egg\}$ {Apple,Basket,Chocolate}→{Dress,Egg}
3. $\{Dress\} \rightarrow \{Apple, Basket\}$ {Dress}→{Apple,Basket}
4. $\{Egg\} \rightarrow \{Apple, Chocolate\}$ {Egg}→{Apple,Chocolate}

step-by-step-closure-calculation-for-dressdress

Step-by-Step Closure Calculation for $\{Dress\}$**{Dress}:Step-by-Step Closure Calculation for \\{ Dress\\}{Dress}:**

1. **Start with** $\{Dress\}$**{Dress}:**
    - Initial closure: $\{Dress\}${Dress}
2. **Apply FD 3:**
    - $\{Dress\} \rightarrow \{Apple, Basket\}$ {Dress}→{Apple,Basket}
    - Closure now: $\{Dress, Apple, Basket\}${Dress,Apple,Basket}
3. **Check if any other FDs can be applied:**

- With $\{Apple, Basket, Dress\}${Apple,Basket,Dress} in the closure, we can apply FD 1 or FD 2.
- From FD 1: $\{Apple, Basket, Chocolate\} \rightarrow \{Dress\}${Apple,Basket,Chocolate}→{Dress}. However, we don't yet have $\{Chocolate\}${Chocolate} in the closure, so FD 1 and FD 2 do not apply at this stage.

4. **No further FDs can be applied.**

final-closure-of-dressdress

Final Closure of $\{Dress\}${Dress}:Final Closure of \{ Dress\}{Dress}:

$$\{Dress, Apple, Basket\}$$

{Dress,Apple,Basket}

Thus, the closure of $\{Dress\}${Dress} includes $\{Dress, Apple, Basket\}${Dress,Apple,Basket}.

you-said-11

Give me Closure of {Apple, Basket}

ChatGPT said:

Let's calculate the **closure of $\{Apple, Basket\}$ {Apple,Basket}** using the given functional dependencies.

functional-dependencies-5

Functional Dependencies:

1. $\{Apple, Basket, Chocolate\} \rightarrow \{Dress\}${Apple,Basket,Chocolate}→{Dress}
2. $\{Apple, Basket, Chocolate\} \rightarrow \{Dress, Egg\}${Apple,Basket,Chocolate}→{Dress,Egg}
3. $\{Dress\} \rightarrow \{Apple, Basket\}${Dress}→{Apple,Basket}
4. $\{Egg\} \rightarrow \{Apple, Chocolate\}${Egg}→{Apple,Chocolate}

step-by-step-closure-calculation-for-applebasketapplebasket

Step-by-Step Closure Calculation for $\{Apple, Basket\}${Apple,Basket}:Step-by-Step Closure Calculation for \{ Apple,Basket\}{Apple,Basket}:

1. **Start with $\{Apple, Basket\}${Apple,Basket}:**

- Initial closure: $\{Apple, Basket\}${Apple,Basket}

2. **Check FDs:**

- FD 1 and FD 2 both require $\{Chocolate\}${Chocolate} to apply. Since $\{Chocolate\}${Chocolate} is not in the closure, we can't apply them yet.

3. **No other FDs can be applied with just $\{Apple, Basket\}${Apple,Basket} at this stage.**

final-closure-of-applebasketapplebasket

Final Closure of $\{Apple, Basket\}${Apple,Basket}:Final Closure of \{ Apple,Basket\}{Apple,Baske

$$\{Apple, Basket\}$$

{Apple,Basket}

Since $\{Apple, Basket\}${Apple,Basket} alone doesn't allow any further FDs to apply, the closure remains $\{Apple, Basket\}${Apple,Basket}.

you-said-12

Give me closure of all these [{Apple, Chocolate}, {Apple, Dress}, {Apple, Egg}, {Basket, Chocolate}, {Basket, Dress}, {Basket, Egg}, {Chocolate, Dress}, {Chocolate, Egg}, {Dress, Egg}]

ChatGPT said:

Let's calculate the closures for each set of attributes one by one, based on the given functional dependencies.

functional-dependencies-6

Functional Dependencies:

1. $\{Apple, Basket, Chocolate\} \rightarrow \{Dress\}${Apple,Basket,Chocolate}→{Dress}
2. $\{Apple, Basket, Chocolate\} \rightarrow \{Dress, Egg\}${Apple,Basket,Chocolate}→{Dress,Egg}
3. $\{Dress\} \rightarrow \{Apple, Basket\}${Dress}→{Apple,Basket}
4. $\{Egg\} \rightarrow \{Apple, Chocolate\}${Egg}→{Apple,Chocolate}

closure-calculations

Closure Calculations:

---

closure-of-applechocolateapplechocolate

- Start with $\{Apple, Chocolate\}$\{Apple,Chocolate\}
- No functional dependencies can be applied directly since $\{Basket\}$\{Basket\} is missing.
- **Closure:** $\{Apple, Chocolate\}$**\{Apple,Chocolate\}**

---

closure-of-appledressappledress

- Start with $\{Apple, Dress\}$\{Apple,Dress\}
- Apply FD 3: $\{Dress\} \rightarrow \{Apple, Basket\}$\{Dress\}→\{Apple,Basket\}, so $\{Apple, Dress\} \rightarrow \{Apple, Bask$
- No further FDs can be applied.
- **Closure:** $\{Apple, Basket, Dress\}$**\{Apple,Basket,Dress\}**

---

closure-of-appleeggappleegg

- Start with $\{Apple, Egg\}$\{Apple,Egg\}
- Apply FD 4: $\{Egg\} \rightarrow \{Apple, Chocolate\}$\{Egg\}→\{Apple,Chocolate\}, so $\{Apple, Egg\} \rightarrow \{Apple, Cho$
- No further FDs can be applied.
- **Closure:** $\{Apple, Chocolate, Egg\}$**\{Apple,Chocolate,Egg\}**

---

closure-of-basketchocolatebasketchocolate

- Start with $\{Basket, Chocolate\}$\{Basket,Chocolate\}
- No FDs can be applied directly because $\{Apple\}$\{Apple\} is missing.
- **Closure:** $\{Basket, Chocolate\}$**\{Basket,Chocolate\}**

---

closure-of-basketdressbasketdress

- Start with $\{Basket, Dress\}$\{Basket,Dress\}
- Apply FD 3: $\{Dress\} \rightarrow \{Apple, Basket\}$\{Dress\}→\{Apple,Basket\}, so $\{Basket, Dress\} \rightarrow \{Apple, Ba$
- No further FDs can be applied.
- **Closure:** $\{Apple, Basket, Dress\}$**\{Apple,Basket,Dress\}**

---

closure-of-basketeggbasketegg

- Start with $\{Basket, Egg\}$\{Basket,Egg\}
- Apply FD 4: $\{Egg\} \rightarrow \{Apple, Chocolate\}$\{Egg\}→\{Apple,Chocolate\}, so $\{Basket, Egg\} \rightarrow \{Basket, A$
- Apply FD 2: $\{Apple, Basket, Chocolate\} \rightarrow \{Dress, Egg\}$\{Apple,Basket,Chocolate\}→\{Dress,Egg\}, so $\{Basket, Egg\} \rightarrow \{Basket, Apple, Chocolate, Dress, Egg\}$\{Basket,Egg\}→\{Basket,Apple,Chocolate,
- **Closure:** $\{Basket, Apple, Chocolate, Dress, Egg\}$**\{Basket,Apple,Chocolate,Dress,Egg\}**

---

closure-of-chocolatedresschocolatedress

- Start with $\{Chocolate, Dress\}$ {Chocolate,Dress}
- Apply FD 3: $\{Dress\} \rightarrow \{Apple, Basket\}$ {Dress}→{Apple,Basket}, so $\{Chocolate, Dress\} \rightarrow \{Apple,$
- Apply FD 2: $\{Apple, Basket, Chocolate\} \rightarrow \{Dress, Egg\}$ {Apple,Basket,Chocolate}→{Dress,Egg}, so $\{Chocolate, Dress\} \rightarrow \{Apple, Basket, Chocolate, Dress, Egg\}$ {Chocolate,Dress}→{Apple,Basket,C
- **Closure:** $\{Apple, Basket, Chocolate, Dress, Egg\}$ **{Apple,Basket,Chocolate,Dress,Egg}**

---

closure-of-chocolateeggchocolateegg

- Start with $\{Chocolate, Egg\}$ {Chocolate,Egg}
- Apply FD 4: $\{Egg\} \rightarrow \{Apple, Chocolate\}$ {Egg}→{Apple,Chocolate}, so $\{Chocolate, Egg\} \rightarrow \{Apple,$
- No further FDs can be applied.
- **Closure:** $\{Apple, Chocolate, Egg\}$ **{Apple,Chocolate,Egg}**

---

closure-of-dresseggdressegg

- Start with $\{Dress, Egg\}$ {Dress,Egg}
- Apply FD 3: $\{Dress\} \rightarrow \{Apple, Basket\}$ {Dress}→{Apple,Basket}, so $\{Dress, Egg\} \rightarrow \{Apple, Baske$
- Apply FD 4: $\{Egg\} \rightarrow \{Apple, Chocolate\}$ {Egg}→{Apple,Chocolate}, so $\{Dress, Egg\} \rightarrow \{Apple, Bas$
- **Closure:** $\{Apple, Basket, Dress, Egg, Chocolate\}$ **{Apple,Basket,Dress,Egg,Chocolate}**

---

final-closures-summary

Final Closures Summary:

- **Closure of** $\{Apple, Chocolate\}$**{Apple,Chocolate}**: $\{Apple, Chocolate\}${Apple,Chocolate}
- **Closure of** $\{Apple, Dress\}$**{Apple,Dress}**: $\{Apple, Basket, Dress\}${Apple,Basket,Dress}
- **Closure of** $\{Apple, Egg\}$**{Apple,Egg}**: $\{Apple, Chocolate, Egg\}${Apple,Chocolate,Egg}
- **Closure of** $\{Basket, Chocolate\}$**{Basket,Chocolate}**: $\{Basket, Chocolate\}${Basket,Chocolate}
- **Closure of** $\{Basket, Dress\}$**{Basket,Dress}**: $\{Apple, Basket, Dress\}${Apple,Basket,Dress}
- **Closure of** $\{Basket, Egg\}$**{Basket,Egg}**: $\{Basket, Apple, Chocolate, Dress, Egg\}${Basket,Apple,Cho
- **Closure of** $\{Chocolate, Dress\}$**{Chocolate,Dress}**: $\{Apple, Basket, Chocolate, Dress, Egg\}${Apple,B
- **Closure of** $\{Chocolate, Egg\}$**{Chocolate,Egg}**: $\{Apple, Chocolate, Egg\}${Apple,Chocolate,Egg}
- **Closure of** $\{Dress, Egg\}$**{Dress,Egg}**: $\{Apple, Basket, Dress, Egg, Chocolate\}${Apple,Basket,Dress,

you-said-13

Give me Closure of all [{Apple, Basket, Chocolate}, {Apple, Basket, Dress}, {Apple, Chocolate, Egg}]

ChatGPT said:

Let's calculate the closures for each of the given sets of attributes based on the functional dependencies.

functional-dependencies-7

Functional Dependencies:

1. $\{Apple, Basket, Chocolate\} \rightarrow \{Dress\}${Apple,Basket,Chocolate}→{Dress}
2. $\{Apple, Basket, Chocolate\} \rightarrow \{Dress, Egg\}${Apple,Basket,Chocolate}→{Dress,Egg}
3. $\{Dress\} \rightarrow \{Apple, Basket\}${Dress}→{Apple,Basket}
4. $\{Egg\} \rightarrow \{Apple, Chocolate\}${Egg}→{Apple,Chocolate}

closure-calculations-1

Closure Calculations:

---

closure-of-applebasketchocolateapplebasketchocolate

- Start with $\{Apple, Basket, Chocolate\}${Apple,Basket,Chocolate}
- Apply FD 1: $\{Apple, Basket, Chocolate\} \rightarrow \{Dress\}${Apple,Basket,Chocolate}→{Dress}, so $\{Apple, Basket, Chocolate\} \rightarrow \{Apple, Basket, Chocolate, Dress\}${Apple,Basket,Chocolate}→{Appl
- Apply FD 2: $\{Apple, Basket, Chocolate\} \rightarrow \{Dress, Egg\}${Apple,Basket,Chocolate}→{Dress,Egg}, so $\{Apple, Basket, Chocolate, Dress\} \rightarrow \{Apple, Basket, Chocolate, Dress, Egg\}${Apple,Basket,Chocol
- **Closure:** $\{Apple, Basket, Chocolate, Dress, Egg\}$**{Apple,Basket,Chocolate,Dress,Egg}**

---

closure-of-applebasketdressapplebasketdress

- Start with $\{Apple, Basket, Dress\}${Apple,Basket,Dress}
- Apply FD 1 or FD 2: These FDs require $\{Chocolate\}${Chocolate} to be present, but $\{Chocolate\}${Chocolate} is not in the current set.
- No further FDs can be applied.
- **Closure:** $\{Apple, Basket, Dress\}$**{Apple,Basket,Dress}**

---

closure-of-applechocolateeggapplechocolateegg

- Start with $\{Apple, Chocolate, Egg\}${Apple,Chocolate,Egg}
- Apply FD 4: $\{Egg\} \rightarrow \{Apple, Chocolate\}${Egg}→{Apple,Chocolate}, so $\{Apple, Chocolate, Egg\} \rightarrow \{$ (no new attributes gained from this FD).
- No further FDs apply since $\{Basket\}${Basket} is missing for FDs 1 and 2, and $\{Dress\}${Dress} is not involved in any FDs relevant to the current set.
- **Closure:** $\{Apple, Chocolate, Egg\}$**{Apple,Chocolate,Egg}**

---

final-closures-summary-1

Final Closures Summary:

- **Closure of** $\{Apple, Basket, Chocolate\}$**{Apple,Basket,Chocolate}**: $\{Apple, Basket, Chocolate, Dres$
- **Closure of** $\{Apple, Basket, Dress\}$**{Apple,Basket,Dress}**: $\{Apple, Basket, Dress\}$\{Apple,Basket,Dr
- **Closure of** $\{Apple, Chocolate, Egg\}$**{Apple,Chocolate,Egg}**: $\{Apple, Chocolate, Egg\}$\{Apple,Choco

you-said-14

Based on above Closures which sets closure contains all {Apple,Basket,Chocolate,Dress,Egg}
ChatGPT said:
Based on the calculated closures, the set that contains all attributes $\{Apple, Basket, Chocolate, Dress, Egg$
is:

- **Closure of** $\{Apple, Basket, Chocolate\}$**{Apple,Basket,Chocolate}**: $\{Apple, Basket, Chocolate, Dres$

This closure includes all the attributes $\{Apple, Basket, Chocolate, Dress, Egg\}$\{Apple,Basket,Chocolate,D
The other sets' closures do not contain all five attributes:

- **Closure of** $\{Apple, Basket, Dress\}$**{Apple,Basket,Dress}**: $\{Apple, Basket, Dress\}$\{Apple,Basket,Dr
- **Closure of** $\{Apple, Chocolate, Egg\}$**{Apple,Chocolate,Egg}**: $\{Apple, Chocolate, Egg\}$\{Apple,Choco

Thus, the only closure that contains all of $\{Apple, Basket, Chocolate, Dress, Egg\}$\{Apple,Basket,Chocolate
is the closure of $\{Apple, Basket, Chocolate\}$\{Apple,Basket,Chocolate\}.
4o mini
you-said-15

The candiate keys are {Apple,Basket,Chocolate}, {Basket,Egg}, {Chocolate,Dress}, {Dress,Egg} for Relation Shopping(Egg, Dress, Chocolate, Basket, Apple); Functional dependencies FD = {{Apple, Basket, Chocolate} -> {Dress}, {Apple, Basket, Chocolate} -> {Dress, Egg}, {Dress } -> { Apple , Basket}, {Egg} -> { Apple , Chocolate }} No answer needed.
ChatGPT said:
Memory updated
Got it! If you have any further questions or need assistance with anything else, feel free to ask!
4o mini
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