CS2102: Database Systems

T01: Relational Model & Algebra

Question 1

Question 1

Recap

Superkey

Definition

A **superkey** is a *subset* of attributes that *uniquely identifies* a tuple in a relation.

Terminology

- ***Relation*** A table *(as a set, the order of rows doesn't matter)*
- ***Attribute*** A column of the table
- ***Tuple*** A row on the table *(as a tuple, the order of values or columns matter)*

Question 1

Recap
- Superkey
- Key
Relation
Question

Recap

Superkey

Definition

A **superkey** is a *subset* of attributes that *uniquely identifies* a tuple in a relation.

Violations

Given a relation R, a set of attributes A is not a superkey if there are two tuples t_1 and t_2 such that

- $\forall\ a \in A : t_1.a = t_2.a$ *(i.e., all values in attributes in A are equal)*
- $\exists\ b \in Attr(R) - A : t_1.b \neq t_2.b$ *(i.e., at least one value in other attributes that is different)*

In this case, we have a *counter-example* where there is a value $t_1.b$ and $t_2.b$ that cannot be *uniquely identified*.

Question 1

Recap
- Superkey
- Key
Relation
Question

Recap

Key

Definition

A **key** is a ***superkey*** that is also <u>*minimal*</u> *(i.e., no **proper subset** of the key is a superkey)*.

Important

> Minimal $\neq$ Minimum

Terminology

- ***Possible*** There is no violation, at least in the current instance
- ***Definite*** No possible *(valid)* instance will violate this

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Q1A

Question

Assuming that the given relation instance is *valid*, write down all the *possible* superkeys of *R*.

How?

Any idea?

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Q1A

Question

Assuming that the given relation instance is *valid*, write down all the *possible* superkeys of *R*.

How?

Steps

1. Pick a set of attributes
2. Try to find **violations!**
    - If none found, then it is *possible* that the set is a superkey
    - Otherwise, the set is *definitely* not a superkey

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Example

- Consider *{B, C}*

Q1A

> **Question**
>
> Assuming that the given relation instance is *valid*, write down all the *possible* superkeys of *R*.

How?

Steps

1. Pick a set of attributes
2. Try to find **violations!**
   - If none found, then it is *possible* that the set is a superkey
   - Otherwise, the set is *definitely* not a superkey

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	**1**	**2**	0
1	**1**	**2**	0
0	0	1	2

Example

- Consider *{B, C}*
- Look at row 2 & 3

Q1A

Question

Assuming that the given relation instance is *valid*, write down all the *possible* superkeys of *R*.

How?

Steps

1. Pick a set of attributes
2. Try to find **violations!**
   - If none found, then it is *possible* that the set is a superkey
   - Otherwise, the set is *definitely* not a superkey

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	**1**	**2**	0
1	**1**	**2**	0
0	0	1	2

Example

- Consider *{B, C}*
- Look at row 2 & 3
- Different value for *A*!

Q1A

Question

Assuming that the given relation instance is *valid*, write down all the *possible* superkeys of *R*.

How?

Steps

1. Pick a set of attributes
2. Try to find **violations!**
   - If none found, then it is *possible* that the set is a superkey
   - Otherwise, the set is *definitely* not a superkey

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Example

- Consider *{A, B, C}*

Q1A

Question

Assuming that the given relation instance is *valid*, write down all the *possible* superkeys of *R*.

How?

Steps

1. Pick a set of attributes
2. Try to find **violations!**
   - If none found, then it is *possible* that the set is a superkey
   - Otherwise, the set is *definitely* not a superkey

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Example

- Consider *{A, B, C}*
- All different values
  - No violation

Q1A

> **Question**
>
> Assuming that the given relation instance is *valid*, write down all the *possible* superkeys of *R*.

How?

Steps

1. Pick a set of attributes
2. Try to find **violations!**
   - If none found, then it is *possible* that the set is a superkey
   - Otherwise, the set is *definitely* not a superkey

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Q1A

Question

Assuming that the given relation instance is *valid*, write down all the *possible* superkeys of *R*.

How?

Steps

1. Pick a set of attributes
2. Try to find **violations!**
    - If none found, then it is *possible* that the set is a superkey
    - Otherwise, the set is *definitely* not a superkey

Try it out (*3 minutes*)

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Q1A

Question

Assuming that the given relation instance is *valid*, write down all the *possible* superkeys of *R*.

Answer

- *{A,C}*
- *{A,D}*
- *{A,B,C,D}*
- *{A,B,C}*
- *{A,B,D}*
- *{A,C,D}*

Question 1

Relation

$R(A,B,C,D)$

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Q1A

Question

Assuming that the given relation instance is *valid*, write down all the *possible* superkeys of R.

Answer

- $\{A,C\}$
- $\{A,D\}$
- $\{A,B,C,D\}$
- $\{A,B,C\}$
- $\{A,B,D\}$
- $\{A,C,D\}$

Question

Do you need check **ALL** possible subset of attributes? *(how many are there?!)*

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Q1A

Question

Assuming that the given relation instance is *valid*, write down all the *possible* superkeys of *R*.

Answer

- *{A,C}*
- *{A,D}*
- *{A,B,C,D}*
- *{A,B,C}*
- *{A,B,D}*
- *{A,C,D}*

Improvement

- Consider *{A,C}* *(it is a superkey)*

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Q1A

Question

Assuming that the given relation instance is *valid*, write down all the *possible* superkeys of *R*.

Answer

- *{A,C}*
- *{A,D}*
- *{**A**,B,**C**,D}*
- *{**A**,B,**C**}*
- *{A,B,D}*
- *{**A**,**C**,D}*

Improvement

- Consider *{A,C}* *(it is a superkey)*
- All superset of *{A,C}* are also superkey! *(why?!)*

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Q1B

Question

Suppose that *{A,C}* is *definitely* a superkey of *R*. Write down all the *possible* candidate keys of *R*.

How?

Do we need to recompute *everything*?

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Q1B

Question

Suppose that *{A,C}* is *definitely* a superkey of *R*. Write down all the *possible* candidate keys of *R*.

Superkey

- ***{A,C}***
- *{A,D}*
- *{A,B,C,D}*
- *{A,B,C}*
- *{A,B,D}*
- *{A,C,D}*

Steps

1. Highlight the sets that are *definitely* superkeys

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Q1B

Question

Suppose that *{A,C}* is *definitely* a superkey of *R*. Write down all the *possible* candidate keys of *R*.

Superkey

- ***{A,C}***
- *{A,D}*
- *{A,B,C,D}*
- *{A,B,C}*
- *{A,B,D}*
- *{A,C,D}*

Steps

1. Highlight the sets that are *definitely* superkeys
2. Remove all supersets of these sets as they are *definitely* **NOT** a key

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Q1B

> **Question**
>
> Suppose that *{A,C}* is ***definitely*** a superkey of *R*. Write down all the *possible* candidate keys of *R*.

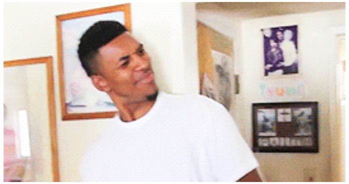
Superkey

- ***{A,C}***
- ***{A,D}***
- *{A,B,C,D}*
- *{A,B,C}*
- ***{A,B,D}***
- *{A,C,D}*

Steps

1. Highlight the sets that are *definitely* superkeys
2. Remove all supersets of these sets as they are *definitely* **NOT** a key
3. The rest are the *possible* keys

Question 1



Recap
Relation
Question
- *Q1A*
- ***Q1B***
- *Q1C*

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Q1B

Question

Suppose that *{A,C}* is *definitely* a superkey of *R*. Write down all the *possible* candidate keys of *R*.

Superkey

- *{A,C}*
- ***{A,D}***
- *{A,B,C,D}*
- *{A,B,C}*
- ***{A,B,D}***
- *{A,C,D}*

Huh?

Why is *{A,B,D}* still a *possible* key when we have *{A,D}*?

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	**0**	**1**
2	1	2	0
1	1	2	0
0	0	1	2
0	0	**1**	**1**

Q1B

Question

Suppose that *{A,C}* is *definitely* a superkey of *R*. Write down all the *possible* candidate keys of *R*.

Superkey

- *{A,C}*
- ***{A,D}***
- *{A,B,C,D}*
- *{A,B,C}*
- ***{A,B,D}***
- *{A,C,D}*

Ans

The table can still be extended to make *{A,D}* not a superkey.

Question 1

Relation

R(A,B,C,D)

R

A	B	C	D
0	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

Q1C

Question

Which of these *(if any)* is likely to be the candidate key of *R*?

Discuss (*3 minutes*)

Question 2

Question 2

Recap
- Foreign Key
Relation
Question

Recap

Superkey

Definition

A **foreign key** is a *subset* of attributes of relation R_1 that *refers to the* ***primary key*** of relation R_2.

Requirement: Each foreign key in R_1 must satisfy one of the following:

- Appear as ***primary key*** in R_2
- Be a NULL value *(or a tuple containing at least one* ***NULL*** *value)*

Terminology

- ***Referencing*** R_1 *(check the value in this table)*
- ***Referenced*** R_2 *(that the value appears in this table)*

Question 2

Recap
Relation
Question

Relation

R(<u>A</u>,B)   *S(<u>W</u>,X,Y,Z)*

R

<u>A</u>	B
3	0
2	1
1	1
0	0

S

<u>W</u>	X	Y	Z
0	4	0	NULL
1	NULL	2	NULL
2	1	2	NULL
3	0	1	NULL

Question 2

Recap
Relation
Question
- Q2

Relation

R(<u>A</u>,B) *S(<u>W</u>,X,Y,Z)*

R

<u>A</u>	B
3	0
2	1
1	1
0	0

S

<u>W</u>	X	Y	Z
0	4	0	NULL
1	NULL	2	NULL
2	1	2	NULL
3	0	1	NULL

Q2

Question

Write down all the *possible* foreign key in *S* that refer to attribute *A* in *R* *(i.e., R.A)*.

How?

Any idea?

Question 2

Recap
Relation
Question
- Q2

Relation

R(<u>A</u>,B) *S(<u>W</u>,X,Y,Z)*

R

<u>A</u>	B
3	0
2	1
1	1
0	0

S

<u>W</u>	X	Y	Z
0	4	0	NULL
1	NULL	2	NULL
2	1	2	NULL
3	0	1	NULL

Q2

Question

Write down all the *possible* foreign key in *S* that refer to attribute *A* in *R* *(i.e., R.A)*.

Worst-Case

For each column, check **row-by-row** *(maybe skip any **NULL** value)*

Question 2

Relation

R(<u>A</u>,B) *S(<u>W</u>,X,Y,Z)*

R

<u>A</u>	B
3	0
2	1
1	1
0	0

S

<u>W</u>	X	Y	Z
0	4	0	NULL
1	NULL	2	NULL
2	1	2	NULL
3	0	1	NULL

Example

- Consider *S.W*

Q2

Question

Write down all the *possible* foreign key in *S* that refer to attribute *A* in *R* *(i.e., R.A)*.

Worst-Case

For each column, check **row-by-row** *(maybe skip any* ***NULL*** *value)*

Question 2

Relation

R(<u>A</u>,B) *S(<u>W</u>,X,Y,Z)*

R

<u>A</u>	B
3	0
2	1
1	1
0	0

S

<u>W</u>	X	Y	Z
0	4	0	NULL
1	NULL	2	NULL
2	1	2	NULL
3	0	1	NULL

Example

- Consider *S.W*
- Check ***row 1***

Q2

Question

Write down all the *possible* foreign key in *S* that refer to attribute *A* in *R* *(i.e., R.A)*.

Worst-Case

For each column, check **row-by-row** *(maybe skip any* ***NULL*** *value)*

Question 2

Recap
Relation
Question
- Q2

Relation

R(<u>A</u>,B) *S(<u>W</u>,X,Y,Z)*

R

<u>A</u>	B
3	0
2	1
1	1
0	0

S

<u>W</u>	X	Y	Z
0	4	0	NULL
1	NULL	2	NULL
2	1	2	NULL
3	0	1	NULL

Example

- Consider *S.W*
- Check ***row 1*** ; ***row 2***

Q2

Question

Write down all the *possible* foreign key in *S* that refer to attribute *A* in *R* *(i.e., R.A)*.

Worst-Case

For each column, check **row-by-row** *(maybe skip any **NULL** value)*

Question 2

Relation

R(<u>*A*</u>,*B*) *S*(<u>*W*</u>,*X*,*Y*,*Z*)

R

<u>A</u>	B
3	0
2	1
1	1
0	0

S

<u>W</u>	X	Y	Z
0	4	0	NULL
1	NULL	2	NULL
2	1	2	NULL
3	0	1	NULL

Example

- Consider *S.W*
- Check ***row 1*** ; ***row 2*** ; ***row 3***

Q2

Question

Write down all the *possible* foreign key in *S* that refer to attribute *A* in *R* *(i.e., R.A)*.

Worst-Case

For each column, check **row-by-row** *(maybe skip any **NULL** value)*

Question 2

Relation

R(A,B) *S(W,X,Y,Z)*

R

A	B
3	0
2	1
1	1
0	0

S

W	X	Y	Z
0	4	0	NULL
1	NULL	2	NULL
2	1	2	NULL
3	0	1	NULL

Example

- Consider *S.W*
- Check ***row 1*** ; ***row 2*** ; ***row 3*** ; ***row 4***

Q2

Question

Write down all the *possible* foreign key in *S* that refer to attribute *A* in *R* *(i.e., R.A)*.

Worst-Case

For each column, check **row-by-row** *(maybe skip any **NULL** value)*

Question 2

Recap
Relation
Question
- Q2

Relation

R(<u>A</u>,B) *S(<u>W</u>,X,Y,Z)*

R

<u>A</u>	B
3	0
2	1
1	1
0	0

S

<u>W</u>	X	Y	Z
0	4	0	NULL
1	NULL	2	NULL
2	1	2	NULL
3	0	1	NULL

Example

- Consider *S.W*
- Check ***row 1*** ; ***row 2*** ; ***row 3*** ; ***row 4***
- **Done** *(no violation, so it's possible)*

Q2

Question

Write down all the *possible* foreign key in *S* that refer to attribute *A* in *R* *(i.e., R.A)*.

Worst-Case

For each column, check **row-by-row** *(maybe skip any **NULL** value)*

Question 2

Recap
Relation
Question
- Q2

Relation

R(<u>A</u>,B) *S(<u>W</u>,X,Y,Z)*

R

<u>A</u>	B
3	0
2	1
1	1
0	0

S

<u>W</u>	X	Y	Z
0	4	0	NULL
1	NULL	2	NULL
2	1	2	NULL
3	0	1	NULL

Example

- Consider *S.X*

Q2

Question

Write down all the *possible* foreign key in *S* that refer to attribute *A* in *R* *(i.e., R.A)*.

Worst-Case

For each column, check **row-by-row** *(maybe skip any **NULL** value)*

Question 2

Relation

R(<u>A</u>,B) *S(<u>W</u>,X,Y,Z)*

R

<u>A</u>	B
3	0
2	1
1	1
0	0

S

<u>W</u>	X	Y	Z
0	**4**	0	NULL
1	NULL	2	NULL
2	1	2	NULL
3	0	1	NULL

Example

- Consider *S.X*
- Check ***row 1***

Q2

Question

Write down all the *possible* foreign key in *S* that refer to attribute *A* in *R* *(i.e., R.A)*.

Worst-Case

For each column, check **row-by-row** *(maybe skip any **NULL** value)*

Question 2

Relation

R(<u>A</u>,B) *S(<u>W</u>,X,Y,Z)*

R

<u>A</u>	B
3	0
2	1
1	1
0	0

S

<u>W</u>	X	Y	Z
0	4	0	NULL
1	NULL	2	NULL
2	1	2	NULL
3	0	1	NULL

Example

- Consider *S.X*
- Check ***row 1***
- **Done** *(failed, so it's impossible)*

Q2

Question

Write down all the *possible* foreign key in *S* that refer to attribute *A* in *R* *(i.e., R.A)*.

Worst-Case

For each column, check **row-by-row** *(maybe skip any **NULL** value)*

Question 2

Recap
Relation
Question
- Q2

Relation

R(<u>A</u>,B) *S(<u>W</u>,X,Y,Z)*

R

<u>A</u>	B
3	0
2	1
1	1
0	0

S

<u>W</u>	X	Y	Z
0	4	0	NULL
1	NULL	2	NULL
2	1	2	NULL
3	0	1	NULL

Q2

Question

Write down all the *possible* foreign key in *S* that refer to attribute *A* in *R* *(i.e., R.A)*.

Worst-Case

For each column, check **row-by-row** *(maybe skip any **NULL** value)*

Can you try the rest (*3 minutes*)?

Question 2

Relation

R(<u>*A*</u>,*B*) *S*(<u>*W*</u>,*X*,*Y*,*Z*)

R

<u>A</u>	B
3	0
2	1
1	1
0	0

S

<u>W</u>	X	Y	Z
0	4	0	NULL
1	NULL	2	NULL
2	1	2	NULL
3	0	1	NULL

Q2

Question

Write down all the *possible* foreign key in *S* that refer to attribute *A* in *R* *(i.e., R.A)*.

Answer

- *(S.W)* ⇝ *(R.A)*
- *(S.Y)* ⇝ *(R.A)*
- *(S.Z)* ⇝ *(R.A)* *(since all* ***NULL****)*

Question 3

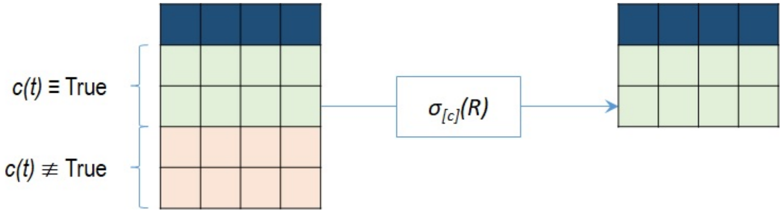
Question 3

Recap
- Unary
Schema
Question

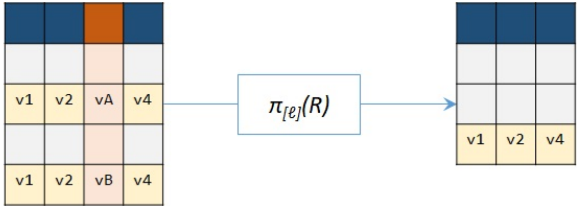
Recap

Unary

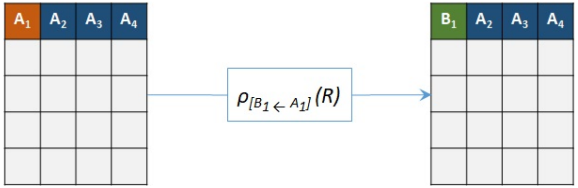
Selection



Projection



Renaming



Question 3

Recap
Schema
Question

Schema

R(<u>A</u>,C)

R

<u>A</u>	C

S(<u>A</u>,D)

S

<u>A</u>	D

T(<u>X</u>,Y)

T

<u>X</u>	Y

Question 3

Question

Q3(i)

> **Question**
>
> Is $\pi_{[A]}(\sigma_{[A < 10]}(R)) \equiv \sigma_{[A < 10]}(\pi_{[A]}(R))$?

Can we use the algebraic law?

Question 3

Question

Q3(i)

Question

Is $\pi_{[A]}(\sigma_{[A < 10]}(R)) \equiv \sigma_{[A < 10]}(\pi_{[A]}(R))$?

Can we use the algebraic law?

Reference

$\pi_{[\ell]}(\sigma_{[\theta]}(R)) \not\equiv \sigma_{[\theta]}(\pi_{[\ell]}(R))$ *(unless θ uses only attributes in ℓ)*

Question 3

Question

Q3(i)

Question

Is $\pi_{[A]}(\sigma_{[A\,<\,10]}(R)) \equiv \sigma_{[A\,<\,10]}(\pi_{[A]}(R))$?

Visual Proof

A	C
σ[A < 10]	σ[A < 10]
σ[A ≥ 10]	σ[A ≥ 10]

Question 3

Question

Q3(i)

> **Question**
>
> Is $\pi_{[A]}(\sigma_{[A\,<\,10]}(R)) \equiv \sigma_{[A\,<\,10]}(\pi_{[A]}(R))$?

Visual Proof

<u>**A**</u>	**C**
σ[A < 10]	σ[A < 10]
σ[A ≥ 10]	σ[A ≥ 10]

<u>**A**</u>	**C**
σ[A < 10]	σ[A < 10]

<u>**A**</u>
σ[A < 10]
σ[A ≥ 10]

Question 3

Question

Q3(i)

> **Question**
>
> Is $\pi_{[A]}(\sigma_{[A < 10]}(R)) \equiv \sigma_{[A < 10]}(\pi_{[A]}(R))$?

Visual Proof

A	C
σ[A < 10]	σ[A < 10]
σ[A ≥ 10]	σ[A ≥ 10]

A	C
σ[A < 10]	σ[A < 10]

A
σ[A < 10]
σ[A ≥ 10]

A
σ[A < 10]

A
σ[A < 10]

Question

Q3(ii)

> **Question**
>
> Is $\pi_{[A]}(\sigma_{[C < 10]}(R)) \equiv \sigma_{[C < 10]}(\pi_{[A]}(R))$?

Can we use the algebraic law?

Question 3

Question

Q3(ii)

> **Question**
>
> Is $\pi_{[A]}(\sigma_{[C < 10]}(R)) \equiv \sigma_{[C < 10]}(\pi_{[A]}(R))$?

Can we use the algebraic law?

Reference

$\pi_{[\ell]}(\sigma_{[\theta]}(R)) \not\equiv \sigma_{[\theta]}(\pi_{[\ell]}(R))$ *(unless θ uses only attributes in ℓ)*

Question 3

Question

Q3(ii)

> **Question**
>
> Is $\pi_{[A]}(\sigma_{[C < 10]}(R)) \equiv \sigma_{[\boldsymbol{C} < 10]}(\pi_{[\boldsymbol{A}]}(R))$?

Can we use the algebraic law?

Reference

$\pi_{[\ell]}(\sigma_{[\theta]}(R)) \not\equiv \sigma_{[\theta]}(\pi_{[\ell]}(R))$ *(unless θ uses only attributes in ℓ)*

Question

Q3(iii)

> **Question**
>
> Is $\pi_{[D,Y]}(S \times T) \equiv \pi_{[D]}(S) \times \pi_{[Y]}(T)$?

Visual Proof

<u>A</u>	D

<u>X</u>	Y

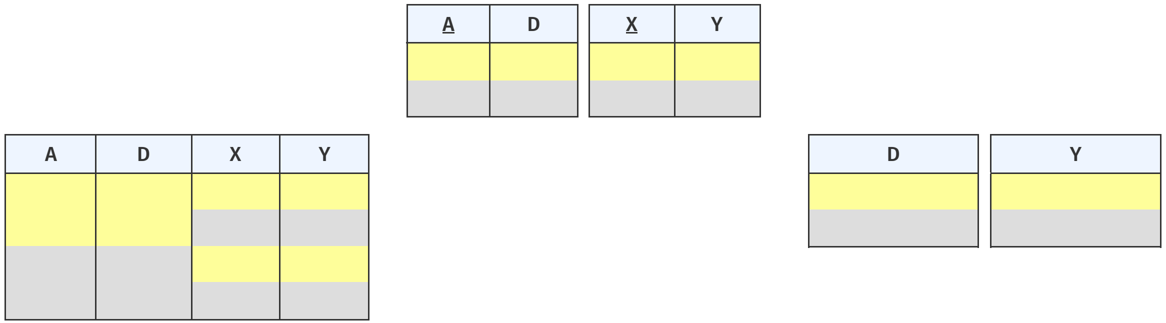
Question 3

Question

Q3(iii)

> **Question**
>
> Is $\pi_{[D,Y]}(S \times T) \equiv \pi_{[D]}(S) \times \pi_{[Y]}(T)$?

Visual Proof



Question 3

Recap
Schema
Question
- Q3(i)
- Q3(ii)
- **Q3(iii)**
- Q3(iv)

Question

Q3(iii)

Question

Is $\pi_{[D,Y]}(S \times T) \equiv \pi_{[D]}(S) \times \pi_{[Y]}(T)$?

Visual Proof

A	D

X	Y

D	Y

D	Y

Question 3

Question

Q3(iv)

> **Question**
>
> Is $\pi_{[D,Y]}(S \times T) \equiv \pi_{[D,Y]}(T \times S)$?

Try it out (3 minutes)

Question 4

Question 4

Question

Q4(i)

> **Question**
>
> Find all pizzas that Moe likes but is not liked by Lisa.

Question 4

Question

Q4(i)

> **Question**
>
> Find all pizzas that Moe likes but is not liked by Lisa.

Ask Yourself

- What are the required attributes?
- Which relation(s) hold the required attributes?
- What operation(s) do you need?
- What are the final attribute(s) needed?

Question 4

Question

Q4(i)

> **Question**
>
> Find all pizzas that Moe likes but is not liked by Lisa.

Ask Yourself

- What are the required attributes?
  cname and *pizza*
- Which relation(s) hold the required attributes?
  Likes
- What operation(s) do you need?
  Set difference
- What are the final attribute(s) needed?
  pizza

Question 4

Question

Q4(i)

> **Question**
>
> Find all pizzas that Moe likes but is not liked by Lisa.

Ask Yourself

- What are the required attributes?
  cname and *pizza*
- Which relation(s) hold the required attributes?
  Likes
- What operation(s) do you need?
  Set difference
- What are the final attribute(s) needed?
  pizza

Possible Answer

$$\pi_{[pizza]}(\sigma_{[cname='Moe']}(Likes)) - \pi_{[pizza]}(\sigma_{[cname='Lisa']}(Likes))$$

Question 4

Question
- Q4(i)
- Q4(ii)
- Q4(iii)

Question

Q4(i)

Question

Find all pizzas that Moe likes but is not liked by Lisa.

Alternative Step

Decomposition!

- Can you find all the pizzas that Moe likes?
- Can you find all the pizzas the Lisa likes?
- Can you subtract one from another?

Possible Answer

$$\pi_{[pizza]}(\sigma_{[cname='Moe']}(Likes)) - \pi_{[pizza]}(\sigma_{[cname='Lisa']}(Likes))$$

Question 4

Question

Q4(ii)

> **Question**
>
> Find all customer-restaurant pairs *(C,R)* where *C* and *R* both located in the same area and *C* likes some pizza that is sold by *R*.

Ask Yourself

- What are the required attributes?
- Which relation(s) hold the required attributes?
- What operation(s) do you need?
- What are the final attribute(s) needed?

Question 4

Question

Q4(ii)

> **Question**
>
> Find all customer-restaurant pairs *(C,R)* where *C* and *R* both located in the same area and *C* likes some pizza that is sold by *R*.

Ask Yourself

- What are the required attributes?
  cname, *rname*, *area* and *pizza*
- Which relation(s) hold the required attributes?
  Customers, *Restaurants*, *Sells* *(is it enough?)*
- What operation(s) do you need?
  at least equi-join *(can we do better?)*
- What are the final attribute(s) needed?
  cname and *rname*

Question 4

Question

Q4(ii)

> **Question**
>
> Find all customer-restaurant pairs *(C,R)* where *C* and *R* both located in the same area and *C* likes some pizza that is sold by *R*.

Ask Yourself

- What are the required attributes?
  cname, *rname*, *area* and *pizza*
- Which relation(s) hold the required attributes?
  Customers, *Restaurants*, *Sells*, and *Likes*
- What operation(s) do you need?
  natural join!
- What are the final attribute(s) needed?
  cname and *rname*

Question 4

Question

Q4(ii)

Question

Find all customer-restaurant pairs *(C,R)* where *C* and *R* both located in the same area and *C* likes some pizza that is sold by *R*.

Ask Yourself

- What are the required attributes?
  cname, *rname*, *area* and *pizza*
- Which relation(s) hold the required attributes?
  Customers, *Restaurants*, *Sells*, and *Likes*
- What operation(s) do you need?
  natural join!
- What are the final attribute(s) needed?
  cname and *rname*

Possible Answer

$\pi_{[cname,rname]}($
Customers $\bowtie$
Restaurants $\bowtie$
Sells $\bowtie$
Likes
$)$

Can you find other answers?

Question 4

Question

Q4(iii)

> **Question**
>
> Suppose the relation *Likes* contains all information about all customers. In other words, if the pair *(cname, pizza)* is not in the relation *Likes*, it means that the customer *cname* <u>*dislikes*</u> the pizza *pizza*. Write a relational algebra expression to find for all customers, the pizza that they dislike. The result should be of the form *(cname, pizza)*.

Decomposition

> Let's apply decomposition. What simpler operations can you do?

Question 4

Question

Q4(iii)

> **Question**
>
> Suppose the relation *Likes* contains all information about all customers. In other words, if the pair *(cname, pizza)* is not in the relation *Likes*, it means that the customer *cname* *dislikes* the pizza *pizza*. Write a relational algebra expression to find for all customers, the pizza that they dislike. The result should be of the form *(cname, pizza)*.

Decomposition

> Let's apply decomposition. What simpler operations can you do?
>
> - Can you find **ALL** possible *cname* and *pizza* pair?

Question 4

Question

Q4(iii)

> **Question**
>
> Suppose the relation *Likes* contains all information about all customers. In other words, if the pair *(cname, pizza)* is not in the relation *Likes*, it means that the customer *cname* *dislikes* the pizza *pizza*. Write a relational algebra expression to find for all customers, the pizza that they dislike. The result should be of the form *(cname, pizza)*.

Decomposition

> Let's apply decomposition. What simpler operations can you do?
>
> - Can you find **ALL** possible *cname* and *pizza* pair?
> - Then we simply subtract the pizza that the customer likes.

Question 4

Question
- *Q4(i)*
- *Q4(ii)*
- ***Q4(iii)***

Question

Q4(iii)

Question

Suppose the relation *Likes* contains all information about all customers. In other words, if the pair *(cname, pizza)* is not in the relation *Likes*, it means that the customer *cname* *dislikes* the pizza *pizza*. Write a relational algebra expression to find for all customers, the pizza that they dislike. The result should be of the form *(cname, pizza)*.

Decomposition

Let's apply decomposition. What simpler operations can you do?

- Can you find **ALL** possible *cname* and *pizza* pair?
- Then we simply subtract the pizza that the customer likes.

Can you write this solution down?

```
postgres=# exit
Press any key to continue . . .
```