CS2102: Database Systems

T02: Relational Algebra & SQL

Question 1

Question 1

Preliminary

Understanding Division

R

A	B
a	1
a	2
b	1
c	1
c	2
c	3
d	2
d	3

S

B
1
2

R / S

A
a
c

Question 1

Preliminary

Understanding Division

R

A	B
a	1
a	2
b	1
c	1
c	2
c	3
d	2
d	3

S

B
1
2

R / S

A
a
c

Question 1

Preliminary

Understanding Division

R

A	B
a	1
a	2
b	1
c	1
c	2
c	3
d	2
d	3

S

B
1
2

R / S

A
a
c

Informal Description

> **R / S** captures all rows in *R* such that rows with same value of *R.A* have the set of values in *R.B* contains all the values in *S.B* *(with projection)*

R / S × S ≈ *R (i.e., almost R)*

A	B
a	1
a	2
c	1
c	2

Question 1

Preliminary

Understanding Division

R

A	B
a	1
a	2
b	1
c	1
c	2
c	3
d	2
d	3

S

B
1
2

R / S

A
a
c

Informal Description

> **R / S** captures all rows in *R* such that rows with same value of *R.A* have the set of values in *R.B* contains all the values in *S.B* *(with projection)*

In Other Words

> All *R.A* in *R* that has all the *S.B* in *B*

Question 1

Question

Q1A

Question

Find all the restaurants that sell all the pizza that Maggie likes and don't sell any pizza that Ralph likes.

Informal Description

R / S captures all rows in *R* such that rows with same value of *R.A* have the set of values in *R.B* contains all the values in *S.B* *(with projection)*

In Other Words

All *R.A* in *R* that has all the *S.B* in *B*

Question 1

Question

Q1A

> **Question**
>
> Find **all the restaurants that sell all the pizza that Maggie likes** and don't sell any pizza that Ralph likes.

- *R.A* = rname — *R* = **?**
- *S.B* = pizza *(Maggie likes)* — *S* = **?**

Informal Description

> **R / S** captures all rows in *R* such that rows with same value of *R.A* have the set of values in *R.B* contains all the values in *S.B* *(with projection)*

In Other Words

> **All *R.A* in *R* that has all the *S.B* in *B***

Question 1

Question

Q1A

> **Question**
>
> Find **all the restaurants that sell all the pizza that Maggie likes** and don't sell any pizza that Ralph likes.

- *R.A* = rname — *R* = ***Sells***
- *S.B* = pizza *(Maggie likes)* — *S* = ***Likes***

Informal Description

> **R / S** captures all rows in *R* such that rows with same value of *R.A* have the set of values in *R.B* contains all the values in *S.B* *(with projection)*

In Other Words

> **All *R.A* in *R* that has all the *S.B* in *B***

Question 1

Question

Q1A

Question

Find **all the restaurants that sell all the pizza that Maggie likes** and don't sell any pizza that Ralph likes.

- *R.A* = rname — *R* = ***Sells***
- *S.B* = pizza *(Maggie likes)* — *S* = ***Likes***

Informal Description

R / S captures all rows in *R* such that rows with same value of *R.A* have the set of values in *R.B* contains all the values in *S.B* *(with projection)*

In Other Words

All *R.A* in *R* that has all the *S.B* in *B*

Try It Out (*3 minutes*)

(just the highlighted part)

Question 1

Question

Q1A

Question

Find **all the restaurants that sell all the pizza that Maggie likes** and don't sell any pizza that Ralph likes.

- *R.A* = rname — *R* = ***Sells***
- *S.B* = pizza *(Maggie likes)* — *S* = ***Likes***

Informal Description

R / S captures all rows in *R* such that rows with same value of *R.A* have the set of values in *R.B* contains all the values in *S.B* *(with projection)*

In Other Words

All *R.A* in *R* that has all the *S.B* in *B*

Quotient

Make sure the attributes match

$Q_1 := \pi_{[rname,pizza]}(\quad) \;/\; \pi_{[pizza]}(\quad)$

Question 1

Question

Q1A

Question

Find **all the restaurants that sell all the pizza that Maggie likes** and don't sell any pizza that Ralph likes.

- *R.A* = rname — *R* = ***Sells***
- *S.B* = pizza *(Maggie likes)* — *S* = ***Likes***

Informal Description

R / S captures all rows in *R* such that rows with same value of *R.A* have the set of values in *R.B* contains all the values in *S.B* *(with projection)*

In Other Words

All *R.A* in *R* that has all the *S.B* in *B*

Relations

Make sure to use the correct relations

$Q_1 := \pi_{[rname,pizza]}(\textbf{\textit{Sells}}) \ / \ \pi_{[pizza]}(\quad)$

Question 1

Question

Q1A

Question

Find **all the restaurants that sell all the pizza that Maggie likes** and don't sell any pizza that Ralph likes.

- *R.A* = rname — *R* = ***Sells***
- *S.B* = pizza *(Maggie likes)* — *S* = ***Likes***

Informal Description

R / S captures all rows in *R* such that rows with same value of *R.A* have the set of values in *R.B* contains all the values in *S.B* *(with projection)*

In Other Words

All *R.A* in *R* that has all the *S.B* in *B*

Relations

Some conditions may be needed

$Q_1 := \pi_{[rname,pizza]}(Sells) \ / \ \pi_{[pizza]}(\sigma_{[cname='Maggie']}(Likes))$

Question 1

Question

Q1A

Question

Find all the restaurants that sell all the pizza that Maggie likes and **don't sell any pizza that Ralph likes**.

- *R.A* = rname — *R* = ***Sells***
- *S.B* = pizza *(Maggie likes)* — *S* = ***Likes***

Informal Description

R / S captures all rows in *R* such that rows with same value of *R.A* have the set of values in *R.B* contains all the values in *S.B* *(with projection)*

In Other Words

All *R.A* in *R* that has all the *S.B* in *B*

Partial Answer

What next?

$Q_1 := \pi_{[rname,pizza]}(Sells) \ / \ \pi_{[pizza]}(\sigma_{[cname='Maggie']}(Likes))$

Question 1

Question

Q1A

> **Question**
>
> Find all the restaurants that sell all the pizza that Maggie likes and **don't sell any pizza that Ralph likes**.

- *R.A* = rname — *R* = ***Sells***
- *S.B* = pizza *(Maggie likes)* — *S* = ***Likes***

Informal Description

> **R / S** captures all rows in *R* such that rows with same value of *R.A* have the set of values in *R.B* contains all the values in *S.B* *(with projection)*

In Other Words

> **All *R.A* in *R* that has all the *S.B* in *B***

Partial Answer

> How to combine?

$Q_1 := \pi_{[rname,pizza]}(Sells) \,/\, \pi_{[pizza]}(\sigma_{[cname='Maggie']}(Likes))$
$Q_2 := \pi_{[rname]}(Sells \bowtie \sigma_{[cname='Ralph']}(Likes))$

Question 1

Question

Q1A

> **Question**
>
> Find all the restaurants that sell all the pizza that Maggie likes and **don't sell any pizza that Ralph likes** .

- *R.A* = rname — *R* = ***Sells***
- *S.B* = pizza *(Maggie likes)* — *S* = ***Likes***

Informal Description

> **R / S** captures all rows in *R* such that rows with same value of *R.A* have the set of values in *R.B* contains all the values in *S.B* *(with projection)*

In Other Words

> **All *R.A* in *R* that has all the *S.B* in *B***

Answer

> Can you write in a single line?

$Q_1 := \pi_{[rname,pizza]}(Sells) \;/\; \pi_{[pizza]}(\sigma_{[cname='Maggie']}(Likes))$
$Q_2 := \pi_{[rname]}(Sells \bowtie \sigma_{[cname='Ralph']}(Likes))$

$Q_1 - Q_2$

Question 1

Question

Q1B

Question

Given *R(A,B)* and *S(B)*. write the relational algebra expression to compute the division of *R* by *S* using only the basic relational operators.

Question 1

Question

S

B
1
2

Q1B

> **Question**
>
> Given *R(A,B)* and *S(B)*. write the relational algebra expression to compute the division of *R* by *S* using only the basic relational operators.

R

A	B
a	1
a	2
b	1
c	1
c	2
c	3
d	2
d	3

R / S

A
a
c

Question 1

Question

Q1B

Question

Given *R(A,B)* and *S(B)*. write the relational algebra expression to compute the division of *R* by *S* using only the basic relational operators.

S

B
1
2

R

A	B
a	1
a	2
b	1
c	1
c	2
c	3
d	2
d	3

Q_1

A
a
c
b
d

Q_2

A
b
d

R / S

A
a
c

Question 1

Question

Q1B

Question

Given *R(A,B)* and *S(B)*. write the relational algebra expression to compute the division of *R* by *S* using only the basic relational operators.

S

B
1
2

R

A	B
a	1
a	2
b	1
c	1
c	2
c	3
d	2
d	3

Q_1

A
a
c
b
d

Q_2

A
b
d

$Q_1 - Q_2$

A
a
c

Question 1

Question

S

B
1
2

Q1B

> **Question**
>
> Given *R(A,B)* and *S(B)*. write the relational algebra expression to compute the division of *R* by *S* using only the basic relational operators.

R

A	B
a	1
a	2
b	1
c	1
c	2
c	3
d	2
d	3

$Q_1 := \pi_{[A]}(R)$

A
a
c
b
d

Q_2

A
b
d

$Q_1 - Q_2$

A
a
c

Question 1

Question

Q1B

Question

Given *R(A,B)* and *S(B)*. write the relational algebra expression to compute the division of *R* by *S* using only the basic relational operators.

S

B
1
2

R

A	B
a	1
a	2
b	1
c	1
c	2
c	3
d	2
d	3

$Q_1 := \pi_{[A]}(R)$

A
a
c
b
d

$Q_3 := Q_1 \times S$

A	B
a	1
a	2
c	1
c	2
b	1
b	2
d	1
d	2

Q_2

A
b
d

$Q_1 - Q_2$

A
a
c

Question 1

Question

S

B
1
2

Q1B

Question

Given *R(A,B)* and *S(B)*. write the relational algebra expression to compute the division of *R* by *S* using only the basic relational operators.

R

A	B
a	1
a	2
b	1
c	1
c	2
c	3
d	2
d	3

$Q_1 := \pi_{[A]}(R)$

A
a
c
b
d

$Q_3 := Q_1 \times S$

A	B
a	1
a	2
c	1
c	2
b	1
b	2
d	1
d	2

$Q_4 := Q_3 - R$

A	B
b	2
d	1

Q_2

A
b
d

$Q_1 - Q_2$

A
a
c

Question 1

Question

S

B
1
2

Q1B

Question

Given *R(A,B)* and *S(B)*. write the relational algebra expression to compute the division of *R* by *S* using only the basic relational operators.

R

A	B
a	1
a	2
b	1
c	1
c	2
c	3
d	2
d	3

$Q_1 := \pi_{[A]}(R)$

A
a
c
b
d

$Q_3 := Q_1 \times S$

A	B
a	1
a	2
c	1
c	2
b	1
b	2
d	1
d	2

$Q_4 := Q_3 - R$

A	B
b	2
d	1

$Q_2 := \pi_{[A]}(Q_4)$

A
b
d

$Q_1 - Q_2$

A
a
c

Answer

$\pi_{[A]}(R) - \pi_{[A]}((\pi_{[A]}(R) \times S) - R)$

Question 2

Question 2

Schema

Preliminary

Offices

> *Offices(office_id, building, floor, room_number, area)*

Employees

> *Employees(emp_id, name, office_id, manager_id)*

Question 2

Constraints

Primary Key

Offices

> *Offices(office_id, building, floor, room_number, area)*
>
> - Primary Key = **?**

Employees

> *Employees(emp_id, name, office_id, manager_id)*
>
> - Primary Key = **?**

Question 2

Constraints

Primary Key

Offices

> *Offices(office_id, building, floor, room_number, area)*
>
> - Primary Key = ***office_id***

Employees

> *Employees(emp_id, name, office_id, manager_id)*
>
> - Primary Key = ***emp_id***

Question 2

Constraints

Types

Offices

> *Offices(office_id, building, floor, room_number, area)*

- office_id ?
- building ?
- floor ?
- room_number ?
- area ?

Question 2

Constraints

Types

Offices

> *Offices(office_id, building, floor, room_number, area)*

- office_id `INTEGER`
- building `TEXT` *(or `VARCHAR` but cannot be `INTEGER`)*
- floor `INTEGER`
- room_number `INTEGER`
- area `INTEGER`

Question 2

Constraints

Types

Employees

> *Employees(<u>emp_id</u>, name, office_id, manager_id)*

- emp_id — ***?***
- name — ***?***
- office_id — `INTEGER` *(must be the same as Office.office_id)*
- manager_id — ***?***

Question 2

Constraints

Types

Employees

> *Employees(<u>emp_id</u>, name, office_id, manager_id)*

- emp_id — `INTEGER` *(must be the same as Employees.manager_id)*
- name — `TEXT` *(or **`VARCHAR`** but cannot be **`INTEGER`**)*
- office_id — `INTEGER` *(must be the same as Office.office_id)*
- manager_id — `INTEGER` *(must be the same as Employees.emp_id)*

Question 2

Schema
Constraints
- Primary Key
- Types
- Foreign Key
Checkpoint
Requirements

Constraints

Foreign Key

Offices

> *Offices(office_id, building, floor, room_number, area)*
>
> Foreign Key = **?**

Employees

> *Employees(emp_id, name, office_id, manager_id)*
>
> Foreign Key = **?**

Question 2

Constraints

Foreign Key

Offices

> *Offices(office_id, building, floor, room_number, area)*
>
> ~~Foreign Key = ?~~

Employees

> *Employees(emp_id, name, office_id, manager_id)*
>
> Foreign Key = (Employees.office_id ⇝ Office.office_id)

Question 2

Checkpoint

Draft #1

Offices

> *Offices(office_id, building, floor, room_number, area)*

```
CREATE TABLE Offices (
  office_id    INTEGER PRIMARY KEY,
  building     TEXT,
  floor        INTEGER,
  room_number  INTEGER,
  area         INTEGER
);
```

Question 2

Checkpoint

Draft #1

Employees

> *Employees(<u>emp_id</u>, name, office_id, manager_id)*

```
CREATE TABLE Employees (
  emp_id       INTEGER PRIMARY KEY,
  name         TEXT,
  office_id    INTEGER REFERENCES Offices,
  manager_id   INTEGER REFERENCES Employees
);
```

Question 2

Requirements

Candidate Key

> **Question**
>
> Any requirement specifies candidate key?

Requirement #2

> *{building, floor, room_number}* is a *candidate key* of *Offices*.*

*This is a *simplified* requirement. It is often not this clear cut.

Question 2

Requirements

Candidate Key

Draft #2

> *{building, floor, room_number}* is a *candidate key* of *Offices*.

```
CREATE TABLE Offices (
  office_id    INTEGER PRIMARY KEY,
  building     TEXT    NOT NULL,
  floor        INTEGER NOT NULL,
  room_number  INTEGER NOT NULL,
  area         INTEGER,
  UNIQUE (building, floor, room_number)
);
```

Question 2

Requirements

Non-Null

> **Question**
>
> Any attributes cannot be `NULL`?

Requirement #5

> The name of each employee must be a non-`NULL` value.

Requirement #6

> Each employee ***must*** be assigned to exactly one office identified by *office_id*.

Question 2

Requirements

Non-Null

Draft #3

> *{name, office_id}* are non-`NULL`.

```
CREATE TABLE Employees (
  emp_id       INTEGER PRIMARY KEY,
  name         TEXT    NOT NULL,
  office_id    INTEGER NOT NULL
    REFERENCES Offices,
  manager_id   INTEGER
    REFERENCES Employees
);
```

Question 2

Schema
Constraints
Checkpoint
Requirements
- Candidate Key
- Non-Null
- Action
- General

Requirements

Action

> **Question**
>
> Any specific action?

Requirement #9 *(similarly #10 but we omit this)*

> A record in *Offices* ***cannot be removed*** if there's some employee assigned to that office.

Requirement #11 *(similarly #12 but we omit this)*

> Any modification to *office_id* in *Offices* is propagated to other database records.

[#]Again, the actual requirements may not be this clear.

Question 2

Requirements

Action

Draft #4

> Requirement #11 and #12 needed additional keywords but not #9 and #10 *(why?)*.

```
CREATE TABLE Employees (
  emp_id       INTEGER PRIMARY KEY,
  name         TEXT    NOT NULL,
  office_id    INTEGER NOT NULL
    REFERENCES Offices ON UPDATE CASCADE,
  manager_id   INTEGER
    REFERENCES Employees ON UPDATE CASCADE
);
```

Question 2

Requirements

General

> **Question**
>
> Any general constraints?

No Other Constraints

- But what about requirement #7?

Requirement #7

> Each employee *may* be managed by at most one manager.

Managed by One Manager

- Employees.manager is not `NULL`

Not Managed by One Manager

- Employees.manager is `NULL`

Question 3

Question 3

Schema

Preliminary

- Books(isbn, title, authors, year, edition, publisher, number_pages, price)
- Customers(cust_id, name, email)
- Carts(cust_id, isbn)
- Purchase(pid, purchase_date, cust_id)
- Purchased_items(pid, isbn)

Question 3

Q3A

Books

> Books records information about the books available for sale in an online shop. Each book has a unique identifier isbn and information about its title, authors, publishers, publication year, edition, number of pages and selling price. The title and authors must have non-`NULL` values. The value of the edition must be non-`NULL` with one of the following values: paperback, hardcover or ebook. The selling price must have a positive value. If the number of pages in known, it must be a positive value.

Question 3

Q3A

Books

> Books records information about the books available for sale in an online shop. Each book has a unique identifier isbn and information about its title, authors, publishers, publication year, edition, number of pages and selling price. The title and authors must have non-`NULL` values. The value of the edition must be non-`NULL` with one of the following values: paperback, hardcover or ebook. The selling price must have a positive value. If the number of pages in known, it must be a positive value.

Keys

- What is/are the key(s)?

Question 3

Q3A

Books

> Books records information about the books available for sale in an online shop. Each book has a ***unique identifier isbn*** and information about its title, authors, publishers, publication year, edition, number of pages and selling price. The title and authors must have non-`NULL` values. The value of the edition must be non-`NULL` with one of the following values: paperback, hardcover or ebook. The selling price must have a positive value. If the number of pages in known, it must be a positive value.

Keys

- What is/are the key(s)?

Question 3

Q3A

Books

> Books records information about the books available for sale in an online shop. Each book has a unique identifier isbn and information about its title, authors, publishers, publication year, edition, number of pages and selling price. The title and authors must have non-`NULL` values. The value of the edition must be non-`NULL` with one of the following values: paperback, hardcover or ebook. The selling price must have a positive value. If the number of pages in known, it must be a positive value.

Non-Null

- What is/are the non-`NULL` attributes?

Question 3

Q3A

Books

> Books records information about the books available for sale in an online shop. Each book has a unique identifier isbn and information about its title, authors, publishers, publication year, edition, number of pages and selling price. The ***title and authors must have non-NULL values*** . The ***value of the edition must be non-NULL*** with one of the following values: paperback, hardcover or ebook. The selling price must have a positive value. If the number of pages in known, it must be a positive value.

Non-Null

- What is/are the non-`NULL` attributes?

Question 3

Q3A

Books

> Books records information about the books available for sale in an online shop. Each book has a unique identifier isbn and information about its title, authors, publishers, publication year, edition, number of pages and selling price. The title and authors must have non-`NULL` values. The value of the edition must be non-`NULL` with one of the following values: paperback, hardcover or ebook. The selling price must have a positive value. If the number of pages in known, it must be a positive value.

General

- What is/are other general constraint(s)?

Question 3

Q3A

Books

> Books records information about the books available for sale in an online shop. Each book has a unique identifier isbn and information about its title, authors, publishers, publication year, edition, number of pages and selling price. The title and authors must have non-`NULL` values. The value of the edition must be non-`NULL` with ***one of the following values: paperback, hardcover or ebook***. The selling ***price must have a positive value***. ***If the number of pages in known, it must be a positive value***.

General

- What is/are other general constraint(s)?

Question 3

Q3A

Books

```
CREATE TABLE Books (
  isbn         TEXT,
  title        TEXT NOT NULL,
  authors      TEXT NOT NULL,
  year         INTEGER,
  edition      TEXT NOT NULL
   CHECK (edition in ('hardcopy', 'paperback', 'ebook')),
  publisher    TEXT,
  number_pages INTEGER CHECK (number_pages > 0),
  price        NUMERIC NOT NULL CHECK (price > 0),
  PRIMARY KEY (isbn)
);
```

Question 3

Q3A

Customers

> Customers stores information about the shop's customers. Each customer has a unique identifier cust_id, a name and an email address. The name must have a non-NULL value.

```
CREATE TABLE Customers (
  cust_id     INTEGER,
  name        TEXT NOT NULL,
  email       TEXT NOT NULL,
  PRIMARY KEY (cust_id)
);
```

Question 3

Q3A

Carts

> Carts stores information about the books in customers' shopping carts. Each shopping cart record indicates a book that a customer is interested to purchase but hs not yet purchased.

```
CREATE TABLE Carts (
  cust_id     INTEGER,
  isbn        TEXT,
  PRIMARY KEY (cust_id, isbn),
  FOREIGN KEY (cust_id) REFERENCES Customers,
  FOREIGN KEY (isbn) REFERENCES Books
);
```

Question 3

Q3A

Purchase

> When a customer decides to purchase their selected books, a new record for this purchase is recorded in the Purchase table which has the following information: enumerate
>
> - A unique identifier pid.
> - The date of the purchase.
> - The customer identifier.

```
CREATE TABLE Purchase (
  pid            INTEGER,
  purchase_date  DATE    NOT NULL,
  cust_id        INTEGER NOT NULL,
  PRIMARY KEY (pid),
  FOREIGN KEY (cust_id) REFERENCES Customers
);
```

Question 3

Q3A

Purchase Item

> In addition, each book in the customers' shopping cart is added to the Purchased_items table and the customers' shopping cart is emptied.

```
CREATE TABLE Purchased_Items (
  pid   INTEGER,
  isbn  TEXT,
  PRIMARY KEY (pid, isbn),
  FOREIGN KEY (pid) REFERENCES Purchase,
  FOREIGN KEY (isbn) REFERENCES Books
);
```

Question 3

Q3B

Modification

> Suppose that the schema of Purchase is changed with purchase_date being replaced by purchase_timestamp, where each customer has at most one purchase at any timestamp.
> How would this change affect your answer for part (a)?

Question

- What are the changes to types?

- What are the changes to other constraints?

Question 3

Q3B

Modification

> Suppose that the schema of Purchase is changed with purchase_ ***date*** being replaced by purchase_ ***timestamp***, where each customer has at most one purchase at any timestamp.
> How would this change affect your answer for part (a)?

Question

- What are the changes to types?
  `DATE` → `TIMESTAMP`
- What are the changes to other constraints?

Question 3

Q3B

Modification

> Suppose that the schema of Purchase is changed with purchase_date being replaced by purchase_timestamp, where ***each customer has at most one purchase at any timestamp***.
> How would this change affect your answer for part (a)?

Question

- What are the changes to types?
  `DATE` → `TIMESTAMP`
- What are the changes to other constraints?
  {purchase_timestamp, cust_pid} is a **candidate key**

Question 3

Q3B

Modification

> Suppose that the schema of Purchase is changed with purchase_date being replaced by purchase_timestamp, where each customer has at most one purchase at any timestamp.
> How would this change affect your answer for part (a)?

Changes

```
CREATE TABLE Purchase (
  pid                 INTEGER,
  purchase_timestamp  TIMESTAMP NOT NULL,
  cust_id             INTEGER   NOT NULL,
  PRIMARY KEY (pid),
  FOREIGN KEY (cust_id) REFERENCES Customers,
  UNIQUE (purchase_timestamp, cust_id)
);
```

Question 3

Q3C

Recap

General Constraints

- Implemented using `CHECK`
- Can encode an *arbitrary boolean condition* that involves:
  - 1 table
  - 1 row
  - Multiple columns

- Operations*:
  - Conjunction `AND` $\wedge$
  - Disjunction `OR` $\vee$
  - Negation `NOT` $\neg$

Useful Properties *(CS1231)*

- $p \Rightarrow q \equiv \neg p \vee q$ *(by definition)*
- $\neg(p \wedge q) \equiv (\neg\, p) \vee (\neg\, q)$ *(De Morgan's Law)*
- $\neg(p \vee q) \equiv (\neg\, p) \wedge (\neg\, q)$ *(De Morgan's Law)*

*Like Python!

Question 3

Q3C

Constraint #1

> **Constraint**
>
> If a book is a hardcover edition, its selling price must be at least 30.

Question

Can it be encoded?

(*1 minute*)

Question 3

Q3C

Constraint #1

> **Constraint**
>
> If a book is a hardcover edition, its selling price must be at least 30.

Question

Can it be encoded?

YES!

(book edition is hardcover) $\Rightarrow$ (price ≥ 30)

Question 3

Q3C

Constraint #1

> **Constraint**
>
> If a book is a hardcover edition, its selling price must be at least 30.

Question

Can it be encoded?

YES!

¬(book edition is hardcover) ∨ (price ≥ 30)

Question 3

Q3C

Constraint #1

> **Constraint**
>
> If a book is a hardcover edition, its selling price must be at least 30.

Question

Can it be encoded?

YES!

(book edition is NOT hardcover) ∨ (price ≥ 30)

Question 3

Q3C

Constraint #1

> **Constraint**
>
> If a book is a hardcover edition, its selling price must be at least 30.

Question

Can it be encoded?

YES!

```
CHECK ((edition <> 'hardcover') OR (price >= 30))
```

Question 3

Q3C

Constraint #2

Constraint

If a book has both hardcover and paperback editions (*for the same book title and authors*), the selling price for the hardcover edition must be higher than the selling price for the paperback edition.

Question

Can it be encoded?

(*1 minute*)

Question 3

Q3C

Constraint #2

> **Constraint**
>
> If a book has both hardcover and paperback editions (*for the same book title and authors*), the selling price for the hardcover edition must be higher than the selling price for the paperback edition.

Question

Can it be encoded?

NO!

Requires checking multiple rows!

- (edition = 'hardcover')
- (edition = 'paperback')

So it cannot be done using only `CHECK`.

Question 3

Q3C

Constraint #3

> **Constraint**
>
> If the number of pages in a book is more than 1000, the edition of the book must be an ebook or its price must be at least 100.

Question

Can it be encoded?

(*1 minute*)

Question 3

Q3C

Constraint #3

Constraint

If the number of pages in a book is more than 1000, the edition of the book must be an ebook or its price must be at least 100.

Question

Can it be encoded?

YES!

$(\text{number\_pages} > 1000) \Rightarrow ((\text{edition} = \text{'ebook'}) \vee (\text{price} \geq 100))$

Question 3

Q3C

Constraint #3

> **Constraint**
>
> If the number of pages in a book is more than 1000, the edition of the book must be an ebook or its price must be at least 100.

Question

Can it be encoded?

YES!

$\neg(\text{number\_pages} > 1000) \vee (\text{edition} = \text{'ebook'}) \vee (\text{price} \geq 100)$

Question 3

Q3C

Constraint #3

> **Constraint**
>
> If the number of pages in a book is more than 1000, the edition of the book must be an ebook or its price must be at least 100.

Question

Can it be encoded?

YES!

$(\text{number\_pages} \leq 1000) \vee (\text{edition} = \text{'ebook'}) \vee (\text{price} \geq 100)$

Question 3

Q3C

Constraint #4

> **Constraint**
>
> All the books published by 'Acme' from 2010 onwards have only ebook edition.

Question

Can it be encoded?

(*1 minute*)

Question 3

Q3C

Constraint #4

> **Constraint**
>
> All the books published by 'Acme' from 2010 onwards have only ebook edition.

Question

Can it be encoded?

YES!

$((\text{publisher} = \text{'Acme'}) \land (\text{year} \geq 2010)) \Rightarrow (\text{edition} = \text{'ebook'})$

Question 3

Q3C

Constraint #4

> **Constraint**
>
> All the books published by 'Acme' from 2010 onwards have only ebook edition.

Question

Can it be encoded?

YES!

$\neg((\text{publisher} = \text{'Acme'}) \wedge (\text{year} \geq 2010)) \vee (\text{edition} = \text{'ebook'})$

Question 3

Q3C

Constraint #4

> **Constraint**
>
> All the books published by 'Acme' from 2010 onwards have only ebook edition.

Question

Can it be encoded?

YES!

$(\neg(\text{publisher} = \text{'Acme'}) \vee \neg(\text{year} \geq 2010)) \vee (\text{edition} = \text{'ebook'})$

Question 3

Q3C

Constraint #4

> **Constraint**
>
> All the books published by 'Acme' from 2010 onwards have only ebook edition.

Question

Can it be encoded?

YES!

(publisher ≠ 'Acme') ∨ (year < 2010) ∨ (edition = 'ebook')

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