

1. Select the make_name, vehicle_id, and model_name of all vehicles, regardless of whether they are on the lot or not, which have a first production year of 1987.

π Make.make_name, Vehicle.vehicle_id, Model.model_name (σ Model.first_production_year = 1987 ((Vehicle \bowtie Vehicle.fk_make_id = Make.make_id (Make)) \bowtie Vehicle.fk_model_id = Model.model_id (Model)))

The screenshot shows a web-based relational algebra calculator. On the left, there is a schema definition for tables: Make (make_id, make_name, country), Color (color_id, name, code), Vehicle (vehicle_id, fk_make_id, fk_model_id, year), Inventory (inventory_id, fk_vehicle_id, fk_color_id, price), Incentive (incentive_id, type, amount, conditions), and Vehicle_Incentive (fk_vehicle_id, fk_incentive_id, valid_till). The main area contains a query editor with two lines of SQL:
 1. π Make.make_name, Vehicle.vehicle_id, Model.model_name (
 2. σ Model.first_production_year = '1987' ((Vehicle \bowtie Vehicle.fk_make_id = Make.make_id (Make)) \bowtie Vehicle.fk_model_id = Model.model_id (Model)))
 Below the editor is an 'execute query' button. To the right of the editor is a 'download' button and a 'history' dropdown. Below these is a visual execution plan diagram. The diagram starts with a projection operator (π) on top of a selection operator (σ). The selection operator is connected to a join operator (\bowtie). This join operator has two inputs: a join operator (\bowtie) and a table node 'Model'. The join operator (\bowtie) has two inputs: a table node 'Vehicle' and a table node 'Make'. Below the diagram, the query is repeated in a text box, followed by a table with three columns: Make.make_name, Vehicle.vehicle_id, and Model.model_name. The table contains two rows: (Make2, 3, Model2). The bottom of the screen shows a Windows taskbar with the date 11/12/2019 and time 3:07 PM.

2. Select the make_name, vehicle_id, and model_name of all vehicles with the color name “Sky Blue” and which are on the lot.

π Make.make_name, Vehicle.vehicle_id, Model.model_name (σ Color.name="Sky Blue" (((((Vehicle \bowtie Vehicle.fk_make_id = Make.make_id(Make)) \bowtie Vehicle.fk_model_id = Model.model_id(Model)) \bowtie Inventory.fk_vehicle_id = Vehicle.vehicle_id (Inventory)) \bowtie Inventory.fk_color_id = Color.color_id(Color)))

relational algebra calculator

https://dbis-uibk.github.io/relax/calc.htm

color_id number
name string
code string

Vehicle
vehicle_id number
fk_make_id number
fk_model_id number
year string

Inventory
inventory_id number
fk_vehicle_id number
fk_color_id number
price number

Incentive
incentive_id number
type string
amount number
conditions string

Vehicle_Incentive
fk_vehicle_id number
fk_incentive_id number
valid_till string

execute query

download history

π Make.make_name, Vehicle.vehicle_id, Model.model_name

σ Color.name = 'Sky Blue'

\bowtie Inventory.fk_color_id = Color.color_id

\bowtie Inventory.fk_vehicle_id = Vehicle.vehicle_id

Color

\bowtie Vehicle.fk_make_id = Model.model_id

Inventory

\bowtie Vehicle.fk_model_id = Model.model_id

Vehicle

Make

π Make.make_name, Vehicle.vehicle_id, Model.model_name (σ Color.name = 'Sky Blue' (\bowtie Vehicle.fk_make_id = Make.make_id (Make) \bowtie Vehicle.fk_model_id = Model.model_id (Model) \bowtie Inventory.fk_vehicle_id = Vehicle.vehicle_id (Inventory) \bowtie Inventory.fk_color_id = Color.color_id (Color)))

Make.make_name	Vehicle.vehicle_id	Model.model_name
BMW	2	Focus

Type here to search

3:12 PM 11/12/2019

3. Select the make_name, model_name and incentive amount for all vehicles on the lot with an incentive type “dealer”.

π Make.make_name, Model.model_name, Incentive.amount (σ Incentive.type = 'dealer' (\bowtie Vehicle.fk_make_id = Make.make_id (Make) \bowtie Vehicle.fk_model_id = Model.model_id (Model) \bowtie Vehicle_Incentive.fk_vehicle_id = Vehicle.vehicle_id

$(\text{Vehicle_Incentive})) \bowtie \text{Incentive.incentive_id} = \text{Vehicle_Incentive.fk_incentive_id} (\text{Incentive}))$

relational algebra calculator

execute query

download history

The interface shows a query tree for the following query:

```

π Make.make_name, Model.model_name, Incentive.amount
(
  σ Incentive.type = 'dealer'
  (
    ⋈ Incentive.incentive_id = Vehicle_Incentive.fk_incentive_id
    (
      ⋈ Vehicle_Incentive.fk_vehicle_id = Vehicle.vehicle_id
      (
        ⋈ Vehicle.fk_model_id = Model.model_id
        (
          ⋈ Vehicle.fk_make_id = Make.make_id
          (
            Vehicle
            ⋈
            Make
          )
        )
      )
    )
  )
  ⋈
  Incentive
)

```

The SQL translation is:

```

π Make.make_name, Model.model_name, Incentive.amount (
  σ Incentive.type = 'dealer' (
    ((
      (Vehicle ⋈ Vehicle.fk_make_id = Make.make_id (Make))
      ⋈ Vehicle.fk_model_id = Model.model_id (Model))
      ⋈ Vehicle_Incentive.fk_vehicle_id = Vehicle.vehicle_id
      (Vehicle_Incentive))
      ⋈ Incentive.incentive_id = Vehicle_Incentive.fk_incentive_id (Incentive))
  )
)

```

Make.make_name	Model.model_name	Incentive.amount
Make1	Model1	400
BMW	Focus	400

4. Convert the following query to relational algebra

```

SELECT Player.id, Team.name, City.name FROM Player
INNER JOIN Team ON Player.team_id = Team.id
INNER JOIN City ON Team.city_id = City.id
WHERE Player.score = 100;

```

$\pi \text{ Player.id, Team.name, City.name } ($

$\sigma \text{ Player.score} = '100' ((\text{Team} \bowtie \text{Team.id} = \text{Player.team_id} (\text{Player}))$

$\bowtie \text{Team.city_id} = \text{City.id} (\text{City}))$

5. For problem 3 above, convert your relational algebra query into a SQL query

relational algebra calculator

https://dbis-uibk.github.io/relax/calc.htm

```

1 SELECT Make.make_name, Model.model_name, Incentive.amount FROM Make
2 INNER JOIN Vehicle ON Vehicle.fk_make_id = Make.make_id
3 INNER JOIN Model ON Model.model_id = Vehicle.fk_model_id
4 INNER JOIN Vehicle_Incentive ON Vehicle_Incentive.fk_vehicle_id = Vehicle.vehicle_id
5 INNER JOIN Incentive ON Incentive.incentive_id = Vehicle_Incentive.fk_incentive_id
6 WHERE Incentive.type = 'dealer'
7
8

```

Warning: DISTINCT is missing; relational algebra uses implicit duplicate elimination

execute query

download history

Make

- make_id number
- make_name string
- country string

Color

- color_id number
- name string
- code string

Vehicle

- vehicle_id number
- fk_make_id number
- fk_model_id number
- year string

Inventory

- inventory_id number
- fk_vehicle_id number
- fk_color_id number
- price number

Incentive

- incentive_id number
- type string
- amount number
- conditions string

Vehicle_Incentive

- fk_vehicle_id number
- fk_incentive_id number
- valid_till string

Type here to search

4:17 PM 11/12/2019

second page

relational algebra calculator

https://dbis-uibk.github.io/relax/calc.htm

Inventory

- inventory_id number
- fk_vehicle_id number
- fk_color_id number
- price number

Incentive

- incentive_id number
- type string
- amount number
- conditions string

Vehicle_Incentive

- fk_vehicle_id number
- fk_incentive_id number
- valid_till string

Π Make.make_name, Model.model_name, Incentive.amount

σ Incentive.type = 'dealer'

⋈ Incentive.incentive_id = Vehicle_Incentive.fk_incentive_id

⋈ Vehicle_Incentive.fk_vehicle_id = Vehicle.vehicle_id

Incentive

⋈ Model.model_id = Vehicle.fk_model_id

Vehicle_Incentive

⋈ Vehicle.fk_make_id = Make.make_id

Model

Make

Vehicle

Π Make.make_name, Model.model_name, Incentive.amount σ Incentive.type = 'dealer' Make ⋈ Vehicle.fk_make_id = Make.make_id Vehicle ⋈ Model.model_id = Vehicle.fk_model_id Model ⋈ Vehicle_Incentive.fk_vehicle_id = Vehicle.vehicle_id Vehicle_Incentive ⋈ Incentive.incentive_id = Vehicle_Incentive.fk_incentive_id Incentive

Make.make_name	Model.model_name	Incentive.amount
Make1	Model1	400
BMW	Focus	400

Type here to search

4:19 PM 11/12/2019