In this assignment you are to write relational algebra queries over a small database, executed using our RA Workbench. Behind the scenes, the RA workbench translates relational algebra expressions into SQL queries over the database stored in SQLite. Since relational algebra symbols aren't readily available on most keyboards, RA uses a special syntax described in our RA Relational Algebra Syntax guide.

We've created a small sample database to use for this assignment. It contains four relations:

```
Person(name, age, gender)       // name is a key
Frequents(name, pizzeria)       // [name,pizzeria] is a key
Eats(name, pizza)               // [name,pizza] is a key
Serves(pizzeria, pizza, price)  // [pizzeria,pizza] is a key
```

View the database. (You can also download the schema and data.)

Instructions: You are to write relational algebra expressions over the pizza database. We strongly suggest that you work the queries out on paper first, using conventional relational algebra symbols. When you click "Check Answer" our back-end runs your query against the sample database. It displays the result and compares your answer against the correct one. When you're satisfied with your solution for a given problem, click the "Submit" button to check your answer.

Please Note: *You are to translate the English into an expression that computes the desired result over all possible databases.* All we actually check is that your query gets the right answer on the small sample database. Thus, even if your solution is marked as correct, it is possible that your query does not correctly reflect the problem at hand. (For example, if we ask for a complex condition that requires accessing all of the tables, but over our small data set in the end the condition is satisfied only by Amy, then the query "\project_{name} (\select_{name='Amy'} Person)" will be marked correct even though it doesn't reflect the actual question.) Circumventing the system in this fashion will get you a high score on the exercises, but it won't help you learn relational algebra. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

You may perform these exercises as many times as you like, so we strongly encourage you to keep working with them until you complete the exercises with full credit.

Q1 (1/1 point)

Find all pizzas eaten by at least one female over the age of 20.

- View the RA Relational Algebra Syntax guide
- If you generate an error, you will see the message from the underlying SQLite system -- apologies for the lack of better error messages

Help

age?

Problem on F

```
\project_{pizza}
(
(\select_{gender='female' and age > 20} Person) \join Eats
)
```

Correct

Correct

Your Query Result:

cheese
mushroom
supreme

Expected Query Result:

cheese
mushroom
supreme

Submit Reset

Q2 (1/1 point)

Find the names of all females who eat at least one pizza served by Straw Hat. (Note: The pizza need not be eaten at Straw Hat.)

- View the RA Relational Algebra Syntax guide
- If you generate an error, you will see the message from the underlying SQLite system -- apologies for the lack of better error messages

```
\project_{name}
(
(\select_{gender = 'female' and pizzeria = 'Straw Hat'} (Person \join Eats \join Serves))
)
```

Correct

Correct

Your Query Result:

Amy
Hil

Expected Query Result:

Amy
Hil

Submit Reset

Q3 (1/1 point)

Find all pizzerias that serve at least one pizza for less than $10 that either Amy or Fay (or both) eat.

- View the RA Relational Algebra Syntax guide
- If you generate an error, you will see the message from the underlying SQLite system -- apologies for the lack of better error messages

```
\project_{pizzeria}
(
(\select_{price<'10'} Serves)
\join (\select_{name = 'Amy' or name = 'Fay'} Eats)
)
```

Correct

Correct

Your Query Result:

Little Caesars
New York Pizza
Straw Hat

Expected Query Result:

Little Caesars
New York Pizza
Straw Hat

Submit Reset

Q4 (1/1 point)

Find all pizzerias that serve at least one pizza for less than $10 that both Amy and Fay eat.

- View the RA Relational Algebra Syntax guide
- If you generate an error, you will see the message from the underlying SQLite system -- apologies for the lack of better error messages

```
\project_{pizzeria}
(
\project_{pizza}(\select_{name='Amy'} Eats)
\intersect
\project_{pizza}(\select_{name='Fay'} Eats)
\join
(\select_{price<'10'} Serves)
)
```

Correct

Correct

Your Query Result:

Little Caesars

Expected Query Result:

Little Caesars

Submit Reset

Q5 (1/1 point)

Find the names of all people who eat at least one pizza served by Dominos but who do not frequent Dominos.

- View the RA Relational Algebra Syntax guide

- If you generate an error, you will see the message from the underlying SQLite system -- apologies for the lack of better error messages

```
\project_{name}
(
(\select_{pizzeria = 'Dominos'} Serves)
\join Eats
)
\diff
\project_{name}
(
(\select_{pizzeria = 'Dominos'} Frequents)
)
```

Correct

Correct

Your Query Result:

Amy
Ben
Dan
Eli
Gus

Expected Query Result:

Amy
Ben
Dan
Eli
Gus

Submit **Reset**

Q6 (1/1 point)

Find all pizzas that are eaten only by people younger than 24, or that cost less than $10 everywhere they're served.

- View the RA Relational Algebra Syntax guide
- If you generate an error, you will see the message from the underlying SQLite system -- apologies for the lack of better error messages

```
((\project_{pizza}Eats)
\diff
(\project_{pizza}((\select_{age >= '24'} Person) \join Eats)))
\union
((\project_{pizza} Serves)
\diff
(\project_{pizza}(\select_{price >= '10'} Serves)))
```

Correct

Correct

Your Query Result:

cheese
pepperoni
sausage

Expected Query Result:

cheese
pepperoni
sausage

Submit Reset

Q7 (1/1 point)

Find the age of the oldest person (or people) who eat mushroom pizza.
(This query is quite challenging; congratulations if you get it right.)

- View the RA Relational Algebra Syntax guide
- If you generate an error, you will see the message from the underlying SQLite system -- apologies for the lack of better error messages

```
\project_{age}(Person \join (\select_{pizza='mushroom'} Eats))
\diff
\project_{age1}
(
\rename_{age1}(\project_{age} (Person \join (\select_{pizza='mushroom'} Eats)))
```

Correct

Correct

Your Query Result:

24

Expected Query Result:

24

Submit Reset

Q8 (1/1 point)

Find all pizzerias that serve only pizzas eaten by people over 30.
(This query is quite challenging; congratulations if you get it right.)

- View the RA Relational Algebra Syntax guide
- If you generate an error, you will see the message from the underlying SQLite system -- apologies for the lack of better error messages

```
\project_{pizzeria}(Serves)
\diff
\project_{pizzeria}(
Serves
\join
(
(\project_{pizza} Serves)
\diff
(\project_{pizza} ((\select_{age > 30} Person) \join Eats))
)
)

```

Correct

Correct

Your Query Result:

Chicago Pizza

Expected Query Result:

Chicago Pizza

Submit Reset

Q9 (1/1 point)

Find all pizzerias that serve every pizza eaten by people over 30.
(This query is very challenging; extra congratulations if you get it right.)

- View the RA Relational Algebra Syntax guide
- If you generate an error, you will see the message from the underlying SQLite system -- apologies for the lack of better error messages

```
(\project_{pizzeria}Serves)
\diff
    (\project_{pizzeria}((\project_{pizzeria}Serves)
        \cross
        (\project_{pizza}(\select_{age>'30'}Person \join Eats))
    \diff
    (\project_{pizzeria,pizza}
    ((\select_{age>'30'}Person \join Eats) \join Serves))))
```

Correct

Correct

Your Query Result:

Chicago Pizza
New York Pizza
Pizza Hut

Expected Query Result:

Chicago Pizza
New York Pizza
Pizza Hut

Submit Reset

Previous Next