

NAME- PRABAL GHOSH

SUBJECT- SEMANTIC WEB (WEB OF DATA) PROJECT

Universite Cote d'Azur

Modeling of an RDF database for the information system of a recipe box company. You can find the description of example such companies on their Web sites (e.g. Hello Fresh) or in the [BBC Good Food magazine review](#).

You should consider the management of the recipes and menus proposed, the ingredients, the customers orders and reviews, and the delivery people. The description of the recipes and ingredients should integrate various aspects: gastronomy, health, budget, etc.

The knowledge graph should be stored in a file rbox.ttl (Turtle format)

You should define your own data model in a file rbox_ontology.ttl containing the ontology (OWL). Optionally, you may also reuse existing ontologies. Optionally you may also define a validating schema in a file rbox_shapes.ttl (SHACL).

Write 10 interesting SPARQL queries implementing competency questions dealing with your knowledge graph. Store them in a single file rbox_query.txt where, for each one, you should indicate in natural language the competency question it implements.

Write a short report (rbox_report.pdf) presenting in natural language your modeling choices.

SPARQL queries:

1. Retrieve the names of all customers who have placed an order containing a recipe with a high calorie count (more than 250):

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX : <http://prabal.org/foodOntology#>

SELECT DISTINCT ?customerName

WHERE {

 ?order rdf:type :Order ;
 :orderContains ?recipe .

Optional{ ?recipe
 rdf:type :Recipe ;
 :name ?recipeName ;
 :calories ?calories .

}

Optional{ ?recipe
 rdf:type :Menu ;
 :includesRecipe ?Menutype .

?Menutype a :Recipe;
 :name ?recipeName ;

:calories ?calories .

```
}
?order :orderedByCustomer ?customer .
?customer rdf:type :Customer ;
        :name ?customerName .
FILTER(?calories > 250)
}
```

	Graph	XML/RDF	Table	Validate
1	<http://prabal.org/foodOntology#order1>	<http://prabal.org/foodOntology#...	Spaghetti Carbo...	500
2	<http://prabal.org/foodOntology#order2>	<http://prabal.org/foodOntology#...	Tomato Basil Pa...	350
3	<http://prabal.org/foodOntology#order3>	<http://prabal.org/foodOntology#...	Spaghetti Carbo...	500
4	<http://prabal.org/foodOntology#order3>	<http://prabal.org/foodOntology#...	Tomato Basil Pa...	350
5	<http://prabal.org/foodOntology#order4>	<http://prabal.org/foodOntology#...	Spaghetti Carbo...	500
6	<http://prabal.org/foodOntology#order4>	<http://prabal.org/foodOntology#...	Tomato Basil Pa...	350

Corese 4.5.0 - Inria UCA I3S - 2023-10-14

File Edit Engine Debug Query User Query Template Display Shacl Shex Event Explain ?

Query4 x Query5 x Query7 x Query8 x Query9 x Query10 x Query11 x Query12 x Query13 x Query14 x Query15 x Query16 x Query17 x Query18 x +

System Shacl editor Turtle editor Query1 x Query2 x Query3 x

Query Modifier Shacl Shex Header Log Push Copy Browse Compare Stop Validate to SPIN to SPARQL Search Refresh stylesheet Default stylesheet

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://prabal.org/foodOntology#>
3 SELECT DISTINCT ?customerName
4 WHERE {
5   ?order rdf:type :Order ;
6   :orderContains ?recipe .
7 Optional{ ?recipe
8   rdf:type :Recipe ;
9   :name ?recipeName ;
10  :calories ?calories .
11 }
12 Optional{ ?recipe
13   rdf:type :Menu ;
14   :includesRecipe ?Menutype .
15 ?Menutype a :Recipe;
16 :name ?recipeName ;
17 :calories ?calories .
18 }
19 ?order :orderedByCustomer ?customer .
20 ?customer rdf:type :Customer ;
21 :name ?customerName .
22 FILTER(?calories > 250)
23 }
```

Graph	XML/RDF	Table	Validate
num		?customerName	
1		Prabal Ghosh	
2		Rohit PS	
3		KL Rahul	
4		Pavel Ghosh	

Explanation :-

This SPARQL query retrieves the names of all customers who have placed an order containing a recipe with a high calorie count (more than 250). Here's a breakdown in natural language:

Pattern 1: Matches triples where an ?order is of type :Order and contains a ?recipe.

Pattern 2 (Optional): Retrieves information about the ?recipe, including its :name and :calories. This pattern is optional because not all orders may have recipes associated with them.

Pattern 3 (Optional): If the previous pattern does not match, this one tries to retrieve information about the :Menu type associated with the ?recipe, and then fetches the recipes included in that menu (?Menutype). Then, it

retrieves the :name and :calories of those recipes. This pattern is also optional as not all orders may contain menus with recipes.

Pattern 4: Fetches the :orderedByCustomer relationship between the ?order and the ?customer.

Pattern 5: Retrieves the name of the ?customer.

FILTER clause: Ensures that only results where the calorie count of the recipe (?calories) is greater than 250 are included in the final result set.

In summary, the query retrieves the names of customers who have placed orders containing recipes or menu items with a calorie count exceeding 250, considering both recipes and menu items if present.

2. Find the names of all menus along with the recipes they include:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://prabal.org/foodOntology#>

```
SELECT ?menu ?menuName ?recipeName
WHERE {
  ?menu rdf:type :Menu ;
        :name ?menuName ;
        :includesRecipe ?recipe .
  ?recipe :name ?recipeName .
}
```

Corese 4.5.0 - Inria UCA I3S - 2023-10-14

File Edit Engine Debug Query User Query Template Display Shacl Shex Event Explain ?

Query5 * Query7 * Query8 * Query9 * Query10 * Query11 * Query12 * Query13 * Query14 * Query15 * Query16 * Query17 * Query18 * Query19 * +

System Shacl editor Turtle editor Query1 * Query2 * Query3 * Query4 *

Query Modifier Shacl Shex Header Log Push Copy Browse Compare Stop Validate to SPIN to SPARQL Search Refresh stylesheet Default stylesheet

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://prabal.org/foodOntology#>
3
4 SELECT ?menu ?menuName ?recipeName
5 WHERE {
6   ?menu rdf:type :Menu ;
7         :name ?menuName ;
8         :includesRecipe ?recipe .
9   ?recipe :name ?recipeName .
10 }
11
```

Graph XML/RDF Table Validate

num	?menu	?menuName	?recipeName
1	<http://prabal.org/foodOntology#menu1>	Spaghetti Carbonara + Grilled Chicken Salad	Spaghetti Carbonara
2	<http://prabal.org/foodOntology#menu1>	Spaghetti Carbonara + Grilled Chicken Salad	Grilled Chicken Salad
3	<http://prabal.org/foodOntology#menu2>	Grilled Chicken Salad + Tomato Basil Pasta	Grilled Chicken Salad
4	<http://prabal.org/foodOntology#menu2>	Grilled Chicken Salad + Tomato Basil Pasta	Tomato Basil Pasta
5	<http://prabal.org/foodOntology#menu3>	Spaghetti Carbonara + Tomato Basil Pasta	Spaghetti Carbonara
6	<http://prabal.org/foodOntology#menu3>	Spaghetti Carbonara + Tomato Basil Pasta	Tomato Basil Pasta
7	<http://prabal.org/foodOntology#menu4>	All Recipes together	Spaghetti Carbonara
8	<http://prabal.org/foodOntology#menu4>	All Recipes together	Grilled Chicken Salad
9	<http://prabal.org/foodOntology#menu4>	All Recipes together	Tomato Basil Pasta

17°C Ciel couvert 14:41 04-04-2024

This SPARQL query retrieves the names of all menus along with the recipes they include.

In natural language, the query can be explained as follows:

Pattern 1: matches any resource that is of type `:Menu` and binds it to the variable `?menu`. It also retrieves the name of the menu and binds it to the variable `?menuName`.

Pattern 2: matches any resource that is included in the menu (`:includesRecipe` relation) and binds it to the variable `?recipe`.

Pattern 3: matches the name of each recipe (`:name` property) and binds it to the variable `?recipeName`.

3. Get the names of all customers who have a high subscription:

PREFIX `rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>`

PREFIX `:` `<http://prabal.org/foodOntology#>`

SELECT `?customerName`

WHERE {

`?customer rdf:type :Customer ;`

`:name ?customerName ;`

`:hasSubscription :HighSubscription .`

}

The screenshot shows the Corese 4.5.0 SPARQL query editor interface. The query is entered in the main text area and is as follows:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://prabal.org/foodOntology#>
3
4 SELECT ?customerName
5 WHERE {
6   ?customer rdf:type :Customer ;
7   :name ?customerName ;
8   :hasSubscription :HighSubscription .
9 }
10
```

Below the query editor, the results are displayed in a table view. The table has two columns: `num` and `?customerName`. The first row shows the result:

num	?customerName
1	Prabal Ghosh

The interface includes various toolbars for editing, running, and visualizing the query. The Windows taskbar at the bottom shows the system time as 19:21 on 04-04-2024.

This SPARQL query retrieves the names of customers who have a high subscription level from the provided RDF data.

In natural language, the query can be explained as follows:

The triple pattern matches any resource that is of type :Customer and binds it to the variable ?customer.

It also retrieves the name of each customer and binds it to the variable ?customerName.

Additionally, it specifies that the customer must have a subscription of type :HighSubscription.

4. Retrieve all ingredients with their names and categories:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX : <http://prabal.org/foodOntology#>

```
SELECT ?ingredient ?name ?category
```

```
WHERE {
```

```
  ?ingredient rdf:type :Ingredient ;
```

```
    :name ?name ;
```

```
    :hasCategory ?category .
```

```
}
```

The screenshot shows the Corese 4.5.0 web interface. The top menu bar includes File, Edit, Engine, Debug, Query, User Query, Template, Display, Shacl, Shex, Event, and Explain. Below the menu is a toolbar with buttons for Query, Modifier, Shacl, Shex, Header, Log, Push, Copy, Browse, Compare, Stop, Validate, to SPIN, to SPARQL, Search, Refresh stylesheet, and Default stylesheet. The main area displays the following SPARQL query:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://prabal.org/foodOntology#>
3
4 SELECT ?ingredient ?name ?category
5 WHERE {
6   ?ingredient rdf:type :Ingredient ;
7     :name ?name ;
8     :hasCategory ?category .
9 }
10
```

Below the query editor, the results are displayed in a table with the following columns: num, ?ingredient, ?name, and ?category. The table contains four rows of data:

num	?ingredient	?name	?category
1	<http://prabal.org/foodOntology#ingredient1>	Spaghetti	Pasta
2	<http://prabal.org/foodOntology#ingredient4>	Chicken Breast	Meat
3	<http://prabal.org/foodOntology#ingredient3>	Tomato	Vegetable
4	<http://prabal.org/foodOntology#ingredient2>	Eggs	Protein

The bottom of the screenshot shows a Windows taskbar with the date 04-04-2024 and time 19:33.

This SPARQL query retrieves information about ingredients, including their names and categories, from the RDF data.

5. List all delivery personnel with their names and contact information:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX : <http://prabal.org/foodOntology#>

```
SELECT ?deliveryPerson ?name ?contactInfo
WHERE {
  ?deliveryPerson rdf:type :DeliveryPersonnel ;
    :name ?name ;
    :hasContactInfo ?contactInfo .
}
```

The screenshot shows the Corese 4.5.0 web interface. The query editor contains the following SPARQL query:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://prabal.org/foodOntology#>
3
4 SELECT ?deliveryPerson ?name ?contactInfo
5 WHERE {
6   ?deliveryPerson rdf:type :DeliveryPersonnel ;
7     :name ?name ;
8     :hasContactInfo ?contactInfo .
9 }
10
```

The results are displayed in a table with the following data:

num	?deliveryPerson	?name	?contactInfo
1	<http://prabal.org/foodOntology#deliveryPerson2>	Aswini	+91 9732620117
2	<http://prabal.org/foodOntology#deliveryPerson1>	John Doe	+91 9732620116
3	<http://prabal.org/foodOntology#deliveryPerson3>	Rahul	+91 9732620118

In summary, this query facilitates the extraction of details about delivery personnel from the ontology, enabling access to their names and contact information.

6. Find all reviews along with the customer names and the rated recipes with their ratings and review:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX : <http://prabal.org/foodOntology#>

```
SELECT ?review ?customerName ?recipe ?customerReview ?rating
WHERE {
  ?review rdf:type :Review ;
    :reviewedBy ?customer ;
    :ratedRecipe ?recipe ;
    :customerReview ?customerReview ;
    :customerRating ?rating .
  ?customer rdf:type :Customer ;
```

```

    :name ?customerName .
}

```

The screenshot shows the Corese 4.5.0 SPARQL query editor. The query is as follows:

```

3 SELECT ?review ?customerName ?recipe ?customerReview ?rating
4 WHERE {
5   ?review rdf:type :Review ;
6   :reviewedBy ?customer ;
7   :ratedRecipe ?recipe ;
8   :customerReview ?customerReview ;
9   :customerRating ?rating .
10  ?customer rdf:type :Customer ;
11  :name ?customerName .
12 }
13
14

```

The results are displayed in a table with the following data:

n.	?review	?customerName	?recipe	?customerReview	?rating
1	<http://prabal.org/foodOntology#review1>	KL Rahul	<http://prabal.org/foodOntology#recipe1>	Tasty Spaghetti Carbonara!	5
2	<http://prabal.org/foodOntology#review2>	Rohit PS	<http://prabal.org/foodOntology#recipe2>	Great salad, loved the grilled chicken!	5
3	<http://prabal.org/foodOntology#review3>	Prabal Ghosh	<http://prabal.org/foodOntology#recipe3>	Delicious pasta, loved the fresh basil!	5

In summary, this query enables the extraction of comprehensive details about reviews, including customer information, rated recipes, and associated ratings and reviews, facilitating analysis or presentation of review data.

7. Find the average rating for each recipe along with its name:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://prabal.org/foodOntology#>

```

```

SELECT ?recipeName (AVG(?rating) AS ?averageRating)
WHERE {
  ?review rdf:type :Review ;
    :ratedRecipe ?recipe ;
    :customerRating ?rating .
  ?recipe rdf:type :Recipe ;
    :name ?recipeName .
}
GROUP BY ?recipeName

```

Corese 4.5.0 - Inria UCA I3S - 2023-10-14

File Edit Engine Debug Query User Query Template Display Shacl Shex Event Explain ?

Query11 x Query12 x Query13 x Query14 x Query15 x Query16 x Query17 x Query18 x Query19 x Query20 x Query21 x Query22 x Query23 x Query24 x Query25 x +

System Shacl editor Turtle editor Query1 x Query2 x Query3 x Query4 x Query5 x Query7 x Query8 x Query9 x Query10 x

Query Modifier Shacl Shex Header Log Push Copy Browse Compare Stop Validate to SPIN to SPARQL Search Refresh stylesheet Default stylesheet

```
3
4 SELECT ?recipeName (AVG(?rating) AS ?averageRating)
5 WHERE {
6   ?review rdf:type :Review ;
7   :ratedRecipe ?recipe ;
8   :customerRating ?rating .
9   ?recipe rdf:type :Recipe ;
10  :name ?recipeName .
11 }
12 GROUP BY ?recipeName
13
14
```

Graph XML/RDF Table Validate

	num	?recipeName	?averageRating
1		Grilled Chicken Salad	5.00000
2		Spaghetti Carbonara	5.00000
3		Tomato Basil Pasta	5.00000

19°C Eclaircies 19:46 04-04-2024

In summary, this query facilitates the extraction of information regarding the average rating for each recipe, providing valuable insights into the overall reception of various recipes among customers.

8. Get the names of customers who have placed orders containing a specific ingredient (e.g., "Chicken Breast"):

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://prabal.org/foodOntology#>

```
SELECT ?order ?customerName
WHERE {
  {
    ?order rdf:type :Order ;
    :orderContains ?ingredient .
    ?ingredient rdf:type :Ingredient ;
    :name "Chicken Breast" .
  }
  UNION
  {
    ?order rdf:type :Order ;
    :orderContains ?ingredient .
    ?ingredient rdf:type :Recipe ;
    :hasIngredient ?recipeIngredient .
    ?recipeIngredient rdf:type :Ingredient ;
    :name "Chicken Breast" .
  }
  UNION
  {
```



```

?order rdf:type :Order ;
      :orderContains ?ingredient .
?ingredient rdf:type :Menu ;
      :includesRecipe ?recipe .
?recipe rdf:type :Recipe ;
      :hasIngredient ?recipeIngredient .
?recipeIngredient rdf:type :Ingredient ;
      :name "Chicken Breast" .
}

?order :orderedByCustomer ?customer .
?customer rdf:type :Customer ;
      :name ?customerName .
}

```

The screenshot shows the Corese 4.5.0 web interface. The top menu bar includes File, Edit, Engine, Debug, Query, User Query, Template, Display, Shacl, Shex, Event, and Explain. Below the menu is a toolbar with buttons for Query, Modifier, Shacl, Shex, Header, Log, Push, Copy, Browse, Compare, Stop, Validate, to SPIN, to SPARQL, and Search. The main area displays a SPARQL query with line numbers 1 through 20. The query is a SELECT statement with a WHERE clause containing two patterns joined by a UNION. The first pattern matches orders containing 'Chicken Breast' as an ingredient. The second pattern matches orders containing 'Chicken Breast' as a recipe ingredient. The results are displayed in a table with three columns: num, ?order, and ?customerName.

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://prabal.org/foodOntology#>
3
4 SELECT ?order ?customerName
5 WHERE {
6   {
7     ?order rdf:type :Order ;
8       :orderContains ?ingredient .
9     ?ingredient rdf:type :Ingredient ;
10       :name "Chicken Breast" .
11   }
12   UNION
13   {
14     ?order rdf:type :Order ;
15       :orderContains ?ingredient .
16     ?ingredient rdf:type :Recipe ;
17       :hasIngredient ?recipeIngredient .
18     ?recipeIngredient rdf:type :Ingredient ;
19       :name "Chicken Breast" .
20   }

```

num	?order	?customerName
1	<http://prabal.org/foodOntology#order5>	Hardik Pandya
2	<http://prabal.org/foodOntology#order5>	Hardik Pandya
3	<http://prabal.org/foodOntology#order6>	Sachin Tendulkar
4	<http://prabal.org/foodOntology#order7>	Prabal Ghosh
5	<http://prabal.org/foodOntology#order1>	Prabal Ghosh
6	<http://prabal.org/foodOntology#order2>	Rohit PS
7	<http://prabal.org/foodOntology#order4>	Pavel Ghosh

This SPARQL query retrieves the names of customers who have placed orders containing a specific ingredient, such as "Chicken Breast", from the RDF data:

Query Intent:

The objective of this query is to find customers who have ordered items containing the specified ingredient, "Chicken Breast".

Pattern Matching:

The WHERE clause defines multiple patterns to match.

It begins by searching for triples where the subject is an order (?order) of type :Order.

It then retrieves the ingredient associated with each order.

The first pattern captures cases where the ingredient directly matches the specified ingredient, "Chicken Breast".

The second pattern handles cases where the ingredient is part of a recipe. It retrieves the recipe ingredients and checks if any of them match "Chicken Breast".

The third pattern addresses cases where the ingredient is part of a menu. It retrieves the recipes included in the menu, then checks if any of their ingredients match "Chicken Breast".

Upon execution, the query retrieves orders containing "Chicken Breast" as an ingredient and then finds the corresponding customers who placed those orders. It returns the names of these customers.

9. Write a SPARQL query that retrieves information about customers, their orders, the recipes they ordered, and the delivery personnel associated with those orders:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX : <http://prabal.org/foodOntology#>

```
SELECT ?customerName ?orderDate ?recipeName ?deliveryPersonName ?deliveryDay
WHERE {
  # Subquery to fetch customers and their orders
  {
    SELECT ?customer ?customerName ?order ?orderDate
    WHERE {
      ?customer rdf:type :Customer ;
        :name ?customerName .
      ?order rdf:type :Order ;
        :orderedByCustomer ?customer ;
        :hasDeliveryDate ?orderDate .
    }
  }

  # Subquery to fetch recipes in orders and their associated delivery personnel
  {
    SELECT ?order ?recipe ?recipeName ?deliveryPerson ?deliveryPersonName ?deliveryDay
    WHERE {
      ?order :orderContains ?recipe .
      ?recipe :name ?recipeName .
      ?order :deliveryPersonDelivers ?deliveryPerson .
      ?deliveryPerson :name ?deliveryPersonName .
      ?order :deliveryDay ?deliveryDay .
    }
  }
}
```

Corese 4.5.0 - Inria UCA I3S - 2023-10-14

File Edit Engine Debug Query User Query Template Display Shacl Shex Event Explain ?

Query14 x Query15 x Query16 x Query17 x Query18 x Query19 x Query20 x Query21 x Query22 x Query23 x Query24 x Query25 x Query26 x Query27 x Query28 x +

System Shacl editor Turtle editor Query1 x Query2 x Query3 x Query4 x Query5 x Query7 x Query8 x Query9 x Query10 x Query11 x Query12 x Query13 x

Query Modifier Shacl Shex Header Log Push Copy Browse Compare Stop Validate to SPIN to SPARQL Search Refresh stylesheet Default stylesheet

```
4 SELECT ?customerName ?orderDate ?recipeName ?deliveryPersonName ?deliveryDay
5 WHERE {
6   # Subquery to fetch customers and their orders
7   {
8     SELECT ?customer ?customerName ?order ?orderDate
9     WHERE {
10      ?customer rdf:type :Customer ;
11      :name ?customerName .
12      ?order rdf:type :Order ;
13      :orderedByCustomer ?customer ;
14      :hasDeliveryDate ?orderDate .
15    }
16  }
17
18  # Subquery to fetch recipes in orders and their associated delivery personnel
19  {
20    SELECT ?order ?recipe ?recipeName ?deliveryPerson ?deliveryPersonName ?deliveryDay
```

Graph XML/RDF Table Validate

num	?customerName	?orderDate	?recipeName	?deliveryPersonName	?deliveryDay
1	KL Rahul	"2022-04-02"^^xsd:date	Spaghetti Carbonara + Tomato Basil Pasta	Aswini	Sunday
2	Rohit PS	"2023-04-02"^^xsd:date	Grilled Chicken Salad + Tomato Basil Pasta	Aswini	Friday
3	Prabal Ghosh	"2024-04-02"^^xsd:date	Spaghetti Carbonara + Grilled Chicken Salad	Aswini	Monday
4	Prabal Ghosh	"2024-04-02"^^xsd:date	Grilled Chicken Salad	John Doe	Saturday
5	Pavel Ghosh	"2024-04-03"^^xsd:date	All Recipes together	John Doe	Monday
6	Hardik Pandya	"2024-04-02"^^xsd:date	Grilled Chicken Salad	John Doe	Friday
7	Hardik Pandya	"2024-04-02"^^xsd:date	Chicken Breast	John Doe	Friday
8	Sachin Tendulkar	"2024-04-02"^^xsd:date	Spaghetti	Rahul	Saturday
9	Sachin Tendulkar	"2024-04-02"^^xsd:date	Grilled Chicken Salad	Rahul	Saturday

This query selects the names of customers, the dates of their orders, the names of recipes in those orders, the names of delivery personnel assigned to those orders, and the delivery days. It uses nested subqueries to first fetch customers and their orders, and then within another subquery, it fetches recipes within those orders along with delivery personnel and delivery days.

10. This provides us with a list of customers who have a mid subscription and have ordered menus containing "Tomato", along with the names of those menus.

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://prabal.org/foodOntology#>

```
SELECT ?customerName ?menuName
WHERE {
  {
    SELECT ?customer ?menu
    WHERE {
      ?order rdf:type :Order ;
      :orderContains ?menu .
      ?menu rdf:type :Menu .
    }
    SELECT ?customer ?menu
```

```

WHERE {
  ?customer rdf:type :Customer ;
    :hasSubscription :MidSubscription .
  ?menu rdf:type :Menu ;
    :includesRecipe ?recipe .
  ?recipe rdf:type :Recipe ;
    :hasIngredient ?ingredient .
  ?ingredient rdf:type :Ingredient ;
    :name "Tomato" .
}
}
}
}
?customer :name ?customerName .
?menu :name ?menuName .
}

```

The screenshot shows the Corese 4.5.0 web interface. The query editor contains the following SPARQL query:

```

7 SELECT ?customer ?menu
8 WHERE {
9   ?order rdf:type :Order ;
10    :orderContains ?menu .
11   ?menu rdf:type :Menu .
12   {
13     SELECT ?customer ?menu
14     WHERE {
15       ?customer rdf:type :Customer ;
16         :hasSubscription :MidSubscription .
17       ?menu rdf:type :Menu ;
18         :includesRecipe ?recipe .
19       ?recipe rdf:type :Recipe ;
20         :hasIngredient ?ingredient .
21       ?ingredient rdf:type :Ingredient ;
22         :name "Tomato" .
23     }
24   }
25 }
26 }
27 ?customer :name ?customerName .
28 ?menu :name ?menuName .

```

The results are displayed in a table with the following columns: num, ?customerName, and ?menuName.

num	?customerName	?menuName
1	KL Rahul	Grilled Chicken Salad + Tomato Basil Pasta
2	Rohit PS	Grilled Chicken Salad + Tomato Basil Pasta
3	KL Rahul	Spaghetti Carbonara + Tomato Basil Pasta
4	Rohit PS	Spaghetti Carbonara + Tomato Basil Pasta
5	KL Rahul	All Recipes together
6	Rohit PS	All Recipes together

This query finds customers with mid subscriptions who have ordered menus containing "Tomato". It retrieves the names of these customers along with the names of the menus they ordered. It uses nested subqueries to filter customers and menus based on subscription type and menu contents. Finally, it extracts the names of the customers and menus from the filtered results.