**SPARQL queries**

**Prefixes**

PREFIX owl: <http://www.w3.org/2002/07/owl#>

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

PREFIX rdfs: [http://www.w3.org/2000/01/rdf-schema#](http://www.w3.org/2000/01/rdf-schema)

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX : <http://www.semanticweb.org/name/ontologies/2019/9/untitled-ontology-9#>

**1. What wines pair with chicken and fish, and has a maximum price of 500kr?**

SELECT **?wine** **?price**

WHERE

{

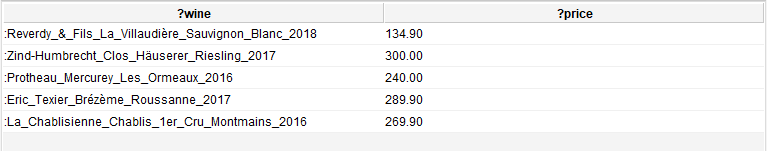
**?wine** a :Wine;

:pairsWithDish :Poultry, :Fish;

:hasPrice **?price**.

FILTER(**?price** <= 500.00).

}



**2. I forgot the full name of a Portuguese red wine I once tried. I remember it paired well with beef and was called something with “dog”. What could it have been?**

SELECT **?wine**

WHERE

{

**?wine** a :Red\_Wine;

:hasName ?wineName;

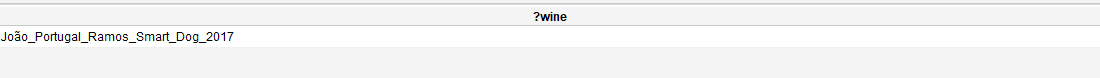
:pairsWithDish :Beef;

:isWineFromRegion ?region.

?region a :Portugese\_Region.

FILTER(CONTAINS(LCASE(?wineName), "dog"))

}



**3. I want an organic red wine from France that has a vintage of 5 years or older.**

SELECT **?wine ?vintageYear ?yearsOld**

WHERE

{

**?wine** a :Red\_Wine;

:hasVintageYear **?vintageYear**;

:isOrganic true;

:isWineFromRegion ?region.

?region a :French\_Region.

BIND(substr(str(now()), 0, 4) AS ?currentYear)

BIND((xsd:integer(?currentYear) - **?vintageYear**) AS **?yearsOld**)

FILTER(**?yearsOld** >= 5)

}



**4. I would like an overview of wine countries, their regions and the grapes they use.**

SELECT DISTINCT **?country** **?region ?grape**

WHERE

{

**?country** a:Country;

:hasRegion **?region**.

**?region** a :Region;

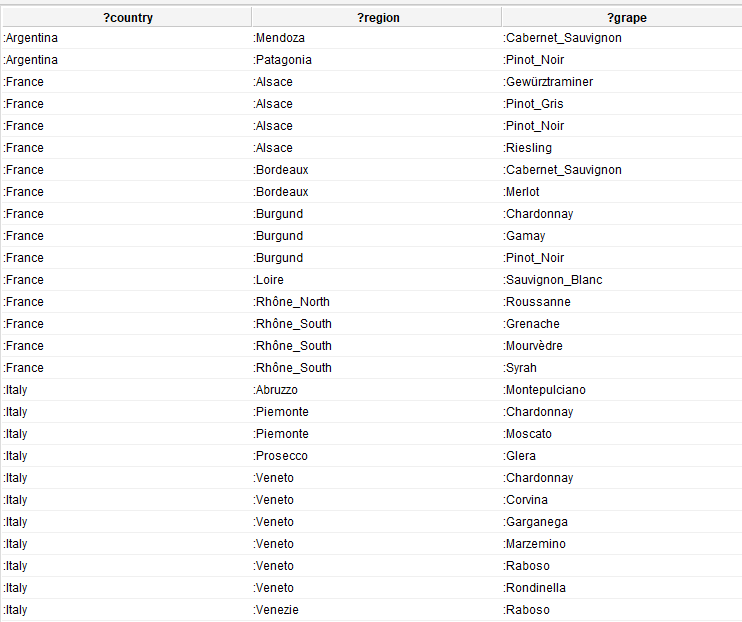
:isRegionOfWine ?wine.

?wine :hasGrape **?grape**.

}

ORDER BY **?country ?region ?grape**

Some of the results:



**5. I want to try a wine from Argentina. However, it should be possible to store and it should not be a Pinot Noir.**

SELECT **?wine** **?region** **?grape**

WHERE

{

**?wine** a :Wine;

:hasStorageExtent ?storageExtent;

:hasGrape **?grape**;

:isWineFromRegion **?region**.

**?region** a:Argentine\_Region.

FILTER(?storageExtent = :Should\_Store || ?storageExtent = :Can\_Store)

FILTER(**?grape** != :Pinot\_Noir)

}

https://lh3.googleusercontent.com/N4tFAtBe7mATeMYPbSw3EIwrm_fu-_GrRMKpqI5-A_IqdxITRdOWpRw06FBeaKbs2yS-I_86tdSRR_HBtbRTSg0de-zRzmwj01nnyyr_FbTM70zVu6mu1jvK9a215ez52zK44lkZ

**6. I once tried this wonderful wine, but I can’t remember its name! It was a wine from Italy, more specifically the region Abruzzo. It had a medium freshness and a medium body. It had an alcohol percentage above 12 and it had been stored without losing its exquisiteness. What could this wine have been?**

SELECT **?wine** **?alcoholPercentage** **?storageExtent**

WHERE

{

**?wine** a :Wine;

:hasStorageExtent **?storageExtent**;

:isWineFromRegion :Abruzzo;

:hasFreshness :Medium\_Freshness;

:hasBody :Medium\_Body;

:hasAlcoholPercentage **?alcoholPercentage**.

FILTER(**?storageExtent** != :No\_Storage)

FILTER(**?alcoholPercentage** > 12.0)

}



**7. Show me the wine profile of the different Burgundy red wines.**

SELECT **?wine** (GROUP\_CONCAT(?wpLabel; separator=", ")AS **?wineProfile**)

WHERE

{

**?wine** a :Red\_Wine;

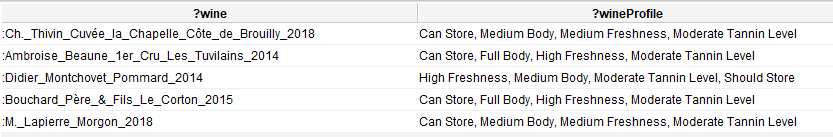
:isWineFromRegion :Burgund;

:hasWineProfile ?wp.

?wp rdfs:label ?wpLabel.

}

GROUP BY **?wine**



**8. I usually only drink sparkling wine. However, I have recently gotten more economically conscious and wonder therefore now if this is the best economic choice. What is the average price of the different types of wine?**

SELECT **?wineLabel** (AVG(?price) AS **?avgPrice**)

WHERE

{

?wineType rdfs:subClassOf :Wine;

rdfs:label **?wineLabel**.

?wine a ?wineType;

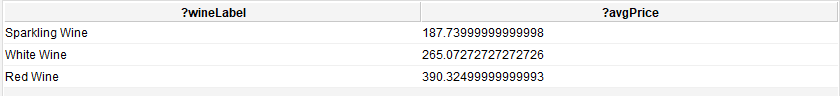
:hasPrice ?price.

FILTER(?wineType != :Wine)

}

GROUP BY **?wineLabel**

ORDERBY **?avgPrice**



**9. What is the cheapest wine I can get for under 200kr with the highest alcohol content?**

The following query did not work with Snap SPARQL Query as this plugin does not support subqueries. It did, however, work in the regular SPARQL Query tab. This plugin has the shortcoming of not being able to reason with inferred facts, such as ?wine being an instance of Wine if it is an instance of a subclass of Wine.

SELECT ?wine ?maxAlcoholPercentage ?price

WHERE

{

?wineType rdfs:subClassOf :Wine.

?wine a ?wineType;

:hasPrice ?price;

:hasAlcoholPercentage ?maxAlcoholPercentage;

{

SELECT (MAX(?alcoholPercentage) AS ?maxAlcoholPercentage)

WHERE

{

?wine :hasAlcoholPercentage ?alcoholPercentage;

:hasPrice ?somePrice.

FILTER (?somePrice < 200.0)

}

}

FILTER(?price < 200.0)

}

ORDER BY ?price

LIMIT 1

https://lh3.googleusercontent.com/s_B0cLOppjYLBE4IFmZ0qkAM3_qjovoK7hKp_Bb7MRR1LE73HM2XuaPfjvpHb5CPNmOC7-gD7kK3fyxbwKix5__PW0y0vPDU16H027tVNtDT-GjIKat0K7aBBsGeJ_YO653aVw_8

**10. I love dry white wines. Which five wineries should I visit based on how many dry whites they produce?**

This query too had to be run in the regular SPARQL Query tab as Snap SPARQL seems to be unable to handle LIMIT.

SELECT ?winery (COUNT(?dryWine) as ?numberOfDryWines)

WHERE

{

?dryWine a :White\_Wine;

:hasSweetness :Dry;

:isProducedBy ?winery.

}

GROUP BY ?winery

ORDER BY DESC (?numberOfDryWines)

LIMIT 5

