

## 1. Before meeting

### a. Gaspar suggestion

i.

Hey João! Depois no fim de semana o vosso grupo quer trocar respostas sobre o trabalho de BDMM

14:53

Para ver se bate certo ? 14:53

ii. **Todos concordam** --> Eu digo-lhe respostas 3, 4, 5

## 2. During meeting

### a. Data Exploration

i. To do

1) Confirmar valores das tabelas - (Umi)

a)

	Node	Property	Count	Distinct Count	Mean	Min	Max	Missing Count	Missing Count (%)
BEERS	abv	372718	939	6.53	0.01	100	45028	12.08	
	availability	417746	20	-	-	-	0	0	
	brewery_id	417746	16569	24 592.84	1	54 144	0	0	
	id	417746	358673	189 196.88	3	374 406	0	0	
	name	417746	298567	-	-	-	0	0	
	notes	417746	48313	-	-	-	55	0.01	
	retired	417746	2	-	-	-	0	0	
	state	417746	68	-	-	-	73831	16.96	
BREWERIES	id	100694	50347	27 870.51	1	54 156	0	0	
	name	100694	45245	-	-	-	0	0	
	notes	100694	3271	-	-	-	170	0.17	
	state	100694	68	-	-	-	22542	22.39	
	types	100694	30	-	-	-	0	0	
CITIES	name	23330	11665	-	-	-	2	0.01	
COUNTRIES	name	400	200	-	-	-	2	0.50	
REVIEWS	beer_id	2549252	189645	77 459.01	3	373 128	0	0	
	date	2549252	6379	-	-	-	19	0.00	
	feel	1484819	17	3.89	1	5	1064433	71.69	
	id	2549252	2546141	4 517 442.86	1	9 073 127	0	0	
	look	1484819	17	3.95	1	5	1064433	71.69	
	overall	1484819	17	3.92	1	5	1064433	71.69	
	score	2549252	401	3.89	1	5	0	0	
	small	1484819	17	3.89	1	5	1064433	71.69	
	taste	1484819	17	3.92	1	5	1064433	71.69	
	text	2549252	814333	-	-	-	19	0.00	
STYLE	name	113	113	-	-	-	1	0.88	
USER	name	123935	123935	-	-	-	1	0.00	

b) Confirmar valores com tabela em 2.1

2) Cofirmar valores (Umi)

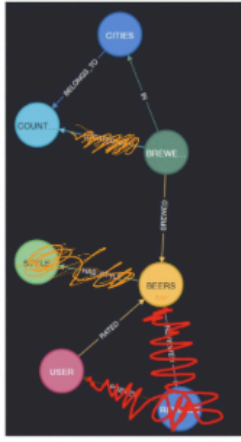
a)

BEERS		<ul style="list-style-type: none"><li><b>abv</b>: According to <i>Hops &amp; Hops</i> (2025) the world's strongest beer is <b>Snake Venom</b> with <b>67.5% ABV</b>. We can see that the <b>maximum</b> value in the database is <b>100</b>, so we will need to verify the beers with more than <b>67.5% ABV</b>.</li><li><b>availability</b>: have 20 different values. We will check what are the possible values.</li><li><b>brewery_id</b>: In total we have 417 746 brewery_ids, but 16 569 distinct brewery_ids. It is possible that some breweries have more than one beer.</li><li><b>id</b>: In total we have 417 746 ids, but 358 873 distinct ids. We need to check the duplicates.</li><li><b>name</b>: In total we have 417 746 names, but 298 567 distinct names. It is possible that some beers have the same name.</li><li><b>notes</b>: In total we have 417 746 notes, but 48 313 distinct notes. Same as the <b>name</b> property, it is possible that some beers have the same notes.</li><li><b>retired</b> have 2 different values - <b>True</b> and <b>False</b>.</li><li><b>state</b> have 68 different values. We will check what are the possible values.</li></ul>
BREWERIES		<ul style="list-style-type: none"><li><b>id</b>: In total we have 100 694 ids, but 50 347 distinct ids. As in <b>BEERS</b>, it is possible that some breweries have more than one beer.</li><li><b>name</b>: In total we have 100 694 names, but 45 245 distinct names. We need to check the duplicates.</li><li><b>notes</b>: In total we have 100 694 notes, but 3 271 distinct notes. Same as the <b>name</b> property, it is possible that some breweries have the same notes.</li><li><b>state</b> have 68 different values (the same as in <b>BEERS</b>). We will check what are the possible values.</li><li><b>types</b> have 30 different values. We will check what are the possible values.</li></ul>
CITIES		<ul style="list-style-type: none"><li><b>name</b>: In total we have 23 330 names, but 11 665 distinct names. We need to check the duplicates.</li></ul>
COUNTRIES		<ul style="list-style-type: none"><li><b>name</b>: In total we have 400 names, but 200 distinct names. We need to check the duplicates.</li></ul>
REVIEWS		<ul style="list-style-type: none"><li><b>beer_id</b>: In total we have 2 549 252 beer_ids, but 189 645 distinct beer_ids. It is possible that some beers have more than one review.</li><li><b>date</b>: We need to convert this property to <b>date</b> format.</li><li><b>feel</b>, <b>look</b>, <b>overall</b>, <b>small</b>, <b>taste</b> and <b>score</b>: We need to convert these properties to <b>float</b> format, but we can see that the values are already clean, because the <b>Min</b> and <b>Max</b> values are between 1 and 5.</li><li><b>id</b>: In total we have 2 549 252 ids, but 2 546 141 distinct ids. We need to check the duplicates.</li><li><b>text</b>: In total we have 2 549 252 texts, but 814 333 distinct texts. It is possible that some reviews have the same text.</li></ul>
STYLE		<ul style="list-style-type: none"><li><b>name</b>: In total we have 113 names and distinct names, so we don't need to check the duplicates.</li></ul>

## b. Database schema

### i. Decisão final

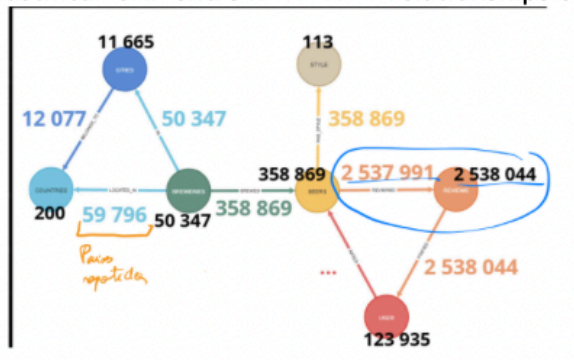
1)



### ii. To do list

- 1) Alterar nome nodes: Não mexer com código. Apenas sugestão. Colocar tudo no Singular
- 2) Justificar
  - a) Remover Style, Has\_Style - Done
  - b) Remoção de REVIEWS, POSTED + REVIEWED - Done
  - c) Não implementação de ligação BREWERIES to COUNTRIES - Done
  - d) Justificar GAP entre REVIEWED relationships e o REVIEWS - Done

e)



### f) Property State na BREWERIES e BEERS

i)

```
1 MATCH (brewery:BREWERIES)-[:BREWED]->(beer:BEERS)
2 WHERE brewery.state <> beer.state
3 RETURN brewery.name AS Brewery, brewery.state AS BreweryState,
4         beer.name AS Beer, beer.state AS BeerState
5 LIMIT 10
```

(no changes, no records)

### ii) Solução:

Um. Eliminar state de BEERS e manter apenas na de BREWERIES - Código (André)

Dois. Justificação para o porquê de não podermos colocar a property state na relação belongs\_to (Filipa)

1. Queríamos colocar a property state na relação Belongs\_to the city country mas não podemos porque.....
2. porque..... há cidades associadas a mais do que um país
3. Escrever aqui

1.

```
total COUNTRIES with CITIES relationship: 12077
AAAAAAAAAAAAAAAAAAAA
ESCREVER
// verify if "every property 'state' of 'BEERS' is also a property of 'BREWERIES'"
query = ""
// collect distinct states from BEERS (including null as a value)
nodes (BEERS)
with collect(distinct &.state) as beerStates
// collect distinct states from BREWERIES (including null as a value)
nodes (BREWERIES)
```

### g) Justificar parte da remoção do Style (João)

i) Fazer na minha versao e nao na minha original



c. Query optimization

- i. Comentários com base nas imagens (Alex)

d. Graph algorithms

- i. 5.1 - comentário (João)

- 1) Assumptions
- 2) Results

3)

Applying Node Similarity Algorithm

1. Filters out countries with fewer than 5 beer styles
  - Only includes countries with at least 5 distinct beer styles in the graph and have c.name different from NULL
  - This ensures that similarity is only computed for countries with a sufficiently diverse beer production.
2. Creates a weighted graph between countries and beer styles
  - Each country node (COUNTRIES) is connected to its top 5 most produced beer styles (STYLE).
  - The relationship weight is the number of beers in that country belonging to each style.
3. Runs the Node Similarity algorithm using these weighted connections
  - Instead of just checking if two countries share beer styles, the algorithm considers how much beer is produced in each style.
  - If two countries produce the same styles but in very different proportions, they will have lower similarity.
  - If two countries produce the same styles in similar proportions, they will have higher similarity.

ALTERAR COMMENT

- ii. 5.2 - comentários (André)

3. **Task geral para todos**

- a. Rerer notebook inteiro
- b. Criar container novo + Correr tudo
- c. Até meeting 16.03 - 18:30