

1. Importar o arquivo .csv para o banco de dados Phpmyadmin.

2. Alterar o nome das colunas.

```
ALTER TABLE `table 1` CHANGE `COL 1` `timestamp` VARCHAR(19)  
CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL;
```

```
ALTER TABLE `table 1` CHANGE `COL 2` `hours_sleepLN` VARCHAR(40)  
CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL;
```

```
ALTER TABLE `table 1` CHANGE `COL 3` `recognition_score` VARCHAR(20)  
CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL;
```

```
ALTER TABLE `table 1` CHANGE `COL 4` `temporal_memory` VARCHAR(21)  
CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL;
```

3. Renomeando tabela:

```
RENAME TABLE `table 1` TO `dados_limpos`;
```

4. Removendo títulos do csv:

```
DELETE FROM dados_limpos WHERE timestamp ='Timestamp'
```

5. Verificar se existem valores maiores que 24 na tabela hours_sleepLN:

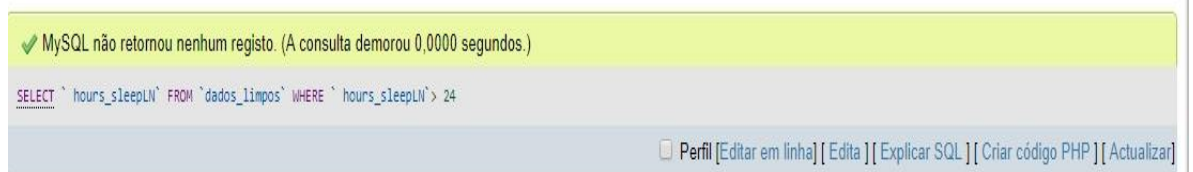


Figura 1- Como é possível notar na imagem, não retornou nenhum resultado.

6. Contar o número de campos vazios:

Coluna hours_sleepLN: SELECT COUNT(*) as 'Número de campos vazios'
FROM dados_limpos WHERE (`hours_sleepLN` IS NULL or `hours_sleepLN`
= '')

Resultado: 2 campos vazios.

Coluna recognition_score: SELECT COUNT(*) as 'Número de campos
vazios' FROM dados_limpos WHERE (`recognition_score` IS NULL or
`recognition_score` = '')

Resultado: 58 campos vazios.

Coluna temporal_memory: SELECT COUNT(*) as 'Número de campos
vazios' FROM dados_limpos WHERE (`temporal_memory` IS NULL or
`temporal_memory` = '')

Resultado: 72 campos vazios.

7. Coloquei NULL em todos os espaços vazios da coluna hours_sleepLN:

```
UPDATE dados_limpos SET `hours_sleepLN` = NULL WHERE `hours_sleepLN` = ''
```

8. Tirei a média da coluna hours_sleepLN:

```
SELECT AVG(`hours_sleepLN`) FROM dados_limpos
```

Resultado arredondado: 6,70 como não existe tal hora irei arredondar para 7 horas.

9. Decidi substituir os valores nulos pela média de horas dormidas que no caso serão 7 horas:

```
UPDATE dados_limpos SET `hours_sleepLN` = 7 WHERE `hours_sleepLN` is NULL
```

10. Convertendo dados da coluna recognition_score do tipo string em double. Várias advertências ocorreram, indicando problemas de conversão. Além de campos vazios, havia uma strings chamadas 'i dunno', 'BBC link returns 404', 'ee', 'n/a', '183(com formatação diferente)' e símbolos de '%' e '&' o sql converteu corretamente os símbolos de '%' e '&'.

Obs: Após fazer a limpeza dos erros pude convertes com sucesso de varchar para double, abaixo estão os comandos utilizados:

```
UPDATE dados_limpos SET `recognition_score` = NULL WHERE  
`recognition_score` = 'i dunno';
```

```
UPDATE dados_limpos SET `recognition_score` = NULL WHERE  
`recognition_score` = '';
```

```
UPDATE dados_limpos SET `recognition_score` = NULL WHERE  
`recognition_score` = 'ee';
```

```
UPDATE dados_limpos SET `recognition_score` = NULL WHERE  
`recognition_score` = 'n/a';
```

```
UPDATE dados_limpos SET `recognition_score` = NULL WHERE  
`recognition_score` = '1 8 3 ';
```

```
UPDATE dados_limpos SET `recognition_score` = NULL WHERE  
`recognition_score` = 'BBC link returns 404';
```

```
SELECT AVG(`recognition_score`) FROM dados_limpos;
```

```
UPDATE dados_limpos SET `recognition_score` = 92.17 WHERE  
`recognition_score` is NULL
```

Obs2: Todos os nulos foram substituídos pela média 92.17, porém achei melhor não deixar números quebrados então usei o seguinte comando:

```
update dados_limpos set `recognition_score` = 93 where `recognition_score` is  
NULL
```

Obs3: Optei por deixar os números que estavam com valor zero.

- 11. Todos os valores maiores que 100 foram substituídos por 100, ou seja, 100 é o valor máximo:**

```
UPDATE dados_limpos SET `recognition_score` = 100 WHERE  
`recognition_score` > 100
```

Resultado: 288 linhas alteradas

- 12. Convertendo dados da coluna temporal_memory do tipo string em double. Várias advertências ocorreram, indicando problemas de conversão. Além de campos vazios, havia uma strings chamadas 'i dunno', 'BBC link returns 404', 'ee', 'n/a', 'Tp', '81 (com formatação diferente)' e símbolos de '%' e '&' o sql converteu corretamente os símbolos de '%' e '&'.**

Obs: Após fazer a limpeza dos erros pude convertes com sucesso de varchar para double, abaixo estão os comandos utilizados:

```
UPDATE dados_limpos SET `temporal_memory` = NULL WHERE  
`temporal_memory` = 'i dunno';  
UPDATE dados_limpos SET `temporal_memory` = NULL WHERE  
`temporal_memory` = '';  
UPDATE dados_limpos SET `temporal_memory` = NULL WHERE  
`temporal_memory` = 'ee';  
UPDATE dados_limpos SET `temporal_memory` = NULL WHERE  
`temporal_memory` = 'n/a';  
UPDATE dados_limpos SET `temporal_memory` = NULL WHERE  
`temporal_memory` = '8 1';  
UPDATE dados_limpos SET `temporal_memory` = NULL WHERE  
`temporal_memory` = 'BBC link returns 404';  
UPDATE dados_limpos SET `temporal_memory` = NULL WHERE  
`temporal_memory` = 'Tp';
```

Obs2: Tirei a média para substituir os números nulos, seu resultado arredondado foi: 78,55 que optei por deixar 79.

```
SELECT AVG(`temporal_memory`) FROM dados_limpos;
```

Obs3: Por fim alterei todos os nulos pela média:

```
UPDATE dados_limpos SET `temporal_memory` = 79 WHERE  
`temporal_memory` is NULL
```

- 13. Todos os valores maiores que 100 foram substituídos por 100, ou seja, 100 é o valor máximo:**

```
UPDATE dados_limpos SET `temporal_memory` = 100 WHERE  
`temporal_memory` > 100
```

Resultado: 3 linhas alteradas.