

Projeto 1 - Análise Delivery Center

Projeto sugerido após módulo 5 sobre agregação de dados.

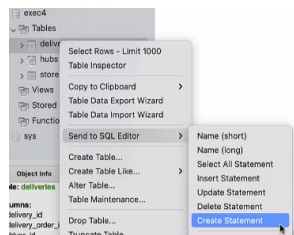
Exercício de Análise de Dados



Dicas: Facilitar a criação da query com arquivos gigantescos, com muitas colunas, e evitando erros de criação/

1. Table Import Wizard
2. Importa o arquivo, como o arquivo é muito grande ele vai demorar, então cancele.
3. Dê refresh e identifique que a tabela foi criada, então vá em Sqnd to SQL Editor > Create Statement
4. Dessa maneira você terá uma query completa da criação da table, pronto para dar LOAD DATA pelo prompt onde é mais rápido.

Observação: Lembre de dar Truncate na tabela para deixá-la limpa.



```
1 CREATE TABLE `deliveries` (  
2   `delivery_id` int DEFAULT NULL,  
3   `delivery_order_id` int DEFAULT NULL,  
4   `driver_id` int DEFAULT NULL,  
5   `delivery_distance_meters` int DEFAULT NULL,  
6   `delivery_status` text  
7 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

PROJETO 1 - Sumário

PROJETO 1 - Sumário

- 1- Qual o número de hubs por cidade?
- 2- Qual o número de pedidos (orders) por status?
- 3- Qual o número de lojas (stores) por cidade dos hubs?
- 4- Qual o maior e o menor valor de pagamento (payment_amount) registrado?
- 5- Qual tipo de driver (driver_type) fez o maior número de entregas?
- 6- Qual a distância média das entregas por tipo de driver (driver_modal)?
- 7- Qual a média de valor de pedido (order_amount) por loja, em ordem decrescente?
- 8- Existem pedidos que não estão associados a lojas? Se caso positivo, quantos?
→ Correção / Método Alternativo
- 9- Qual o valor total de pedido (order_amount) no channel 'FOOD PLACE'?
- 10- Quantos pagamentos foram cancelados (chargeback)?
→ Correção
- 11- Qual foi o valor médio dos pagamentos cancelados (chargeback)?
→ Correção
- 12- Qual a média do valor de pagamento por método de pagamento (payment_method) em ordem decrescente?
- 13- Quais métodos de pagamento tiveram valor médio superior a 100?
- 14- Qual a média de valor de pedido (order_amount) por estado do hub (hub_state), segmento da loja (store_segment) e tipo de canal (channel_type)?
- 15- Qual estado do hub (hub_state), segmento da loja (store_segment) e tipo de canal(channel_type) teve média de valor de pedido (order_amount) maior que 450?

16- Qual o valor total de pedido (order_amount) por estado do hub (hub_state), segmento da loja (store_segment) e tipo de canal (channel_type)

→ Correção

17- Quando o pedido era do Hub do Rio de Janeiro (hub_state), segmento de loja 'FOOD', tipo de canal Marketplace e foi cancelado, qual foi a média de valor do pedido(order_amount)?

18- Quando o pedido era do segmento de loja 'GOOD', tipo de canal Marketplace e foi cancelado, algum hub_state teve total de valor do pedido superior a 100.000?

19- Em que data houve a maior média de valor do pedido (order_amount)? Dica:

→ Correção

20- Em quais datas o valor do pedido foi igual a zero (ou seja, não houve venda)?

1- Qual o número de hubs por cidade?

```
SELECT COUNT(hub_city) as quantidade, hub_city from tb_hubs  
GROUP BY hub_city
```

quantidade	hub_city
4	PORTO ALEGRE
9	RIO DE JANEIRO
15	SAO PAULO
4	CURITIBA

2- Qual o número de pedidos (orders) por status?

```
SELECT COUNT(order_status) AS amount_by_status, order_status  
FROM tb_orders  
GROUP BY order_status
```

amount_by_status	order_status
16979	CANCELED
352020	FINISHED

3- Qual o número de lojas (stores) por cidade dos hubs?

```
SELECT COUNT(store_name) as store_amount, hub_city  
FROM tb_stores S INNER JOIN tb_hubs H  
ON S.hub_id = H.hub_id  
GROUP BY hub_city
```

store_amount	hub_city
48	PORTO ALEGRE
326	RIO DE JANEIRO
460	SAO PAULO
117	CURITIBA

4- Qual o maior e o menor valor de pagamento (payment_amount) registrado?

```
SELECT MAX(payment_amount), MIN(payment_amount) FROM tb_payments
```

5- Qual tipo de driver (driver_type) fez o maior número de entregas?

```
# Quantidade de entregas por ID

SELECT COUNT(delivery_status) amount_delivered, V.driver_id
FROM tb_deliveries D INNER JOIN tb_drivers V
WHERE D.driver_id = V.driver_id AND delivery_status = "DELIVERED"
GROUP BY V.driver_id
ORDER BY amount_delivered DESC;

#####
#
#
# Na correção do exercício o professor não cogitou o uso do "Cancelled" e "Delivered". Na minha interpretação, ordens canceladas não são en
# Então é sim necessário filtrar não só pela quantidade com IDs, mas também com uma condicional de "DELIVERED"
#
#
#####

# Qual drive_type fez maior numero de entregas:

SELECT COUNT(delivery_status) amount_delivered, V.driver_type
FROM tb_deliveries D INNER JOIN tb_drivers V
WHERE D.driver_id = V.driver_id AND delivery_status = "DELIVERED"
GROUP BY V.driver_type
ORDER BY amount_delivered DESC;
```

amount_delivered	driver_type
259249	FREELANCE
103517	LOGISTIC OPERATOR

6- Qual a distância média das entregas por tipo de driver (driver_modal)?

```
SELECT ROUND(AVG(delivery_distance_meters), 2) as average_distance, V.driver_modal
FROM tb_deliveries D INNER JOIN tb_drivers V
WHERE D.driver_id = V.driver_id AND delivery_status = "DELIVERED"
GROUP BY V.driver_modal
ORDER BY average_distance DESC;
```

average_distance	driver_modal
3446.90	MOTOBOY
1109.20	BIKER

7- Qual a média de valor de pedido (order_amount) por loja, em ordem decrescente?

```
SELECT store_name, ROUND(AVG(order_amount), 2) AS average_order_amount
FROM tb_orders orders INNER JOIN tb_stores stores
WHERE orders.store_id = stores.store_id
GROUP BY store_name
ORDER BY average_order_amount DESC
LIMIT 10;
```

store_name	average_order_amount
ZAMZIMU	5266.63
OIGAMA	2412.67
GRIME CRU	2186.79
SIRSUMG	2096.59
CEI	1764.15

store_name	average_order_amount
VACMIR ZUGI	1518
CAI RIRAMARI	1404.61
GRUGIRO	1350.67
PIMOURI	1100.76
AMACI RAMIMA	1083.92

8- Existem pedidos que não estão associados a lojas? Se caso positivo, quantos?

```
SELECT store_id from tb_orders
WHERE store_id = "" OR store_id = NULL
```

#Não temos pedidos sem associação com alguma loja

→ Correção / Método Alternativo



Aqui o professor utilizou outro método para chegar no mesmo resultado.

Utilizou a função COALESCE que funciona como um case, assim preenchendo valores nulos com um determinado valor.

Ele também utilizou do Left Join, pra caso haja de fato algum pedido sem loja atrelado, ele retornaria "Sem Loja"

```
SELECT COALESCE(store_name, "Sem Loja"), COUNT(order_id) AS contagem
FROM tb_orders orders LEFT JOIN tb_stores stores
ON orders.store_id = stores.store_id
GROUP BY store_name
```

Chegou no "Mesmo" resultado. Porém aqui ele retornou todos os pedidos de todas as lojas, e se CASO houvesse algum pedido sem loja atrelado
Minha query é mais simples e chegou no mesmo resultado, porém a dele é de melhor visualização, mesmo tendo de carregar todos os dados, et

9- Qual o valor total de pedido (order_amount) no channel 'FOOD PLACE'?

Quantidade de Pedidos

```
SELECT ROUND(COUNT(order_amount), 2) AS order_amount, C.channel_name
FROM tb_orders O INNER JOIN tb_channels C
ON C.channel_name = "FOOD PLACE" AND O.channel_id = C.channel_id
GROUP BY channel_name
```

Valor total somado de todos os pedidos

```
SELECT ROUND(SUM(order_amount), 2) AS order_amount, C.channel_name
FROM tb_orders O INNER JOIN tb_channels C
ON C.channel_name = "FOOD PLACE" AND O.channel_id = C.channel_id
GROUP BY channel_name
```

total_orders	channel_name
24703719.64	FOOD PLACE

10- Quantos pagamentos foram cancelados (chargeback)?

```
SELECT COUNT(delivery_status) AS status_amount, delivery_status
FROM tb_deliveries
WHERE delivery_status = "CANCELLED"
GROUP BY delivery_status
```

```
# status_amount | delivery_status
# 7253          | CANCELLED
```

QUERY ERRADA - NÃO CONDIZ COM O QUE FOI PEDIDO COM CHARGEBACK, CORREÇÃO ABAIXO.

→ Correção

Por não lembrar das definições completas dadas no documento do dataset, os pagamentos **CANCELADOS** não são achados através do delivery_status = "CANCELLED" como fiz acima,

e sim através do payment_status = "CHARGEBACK".

Fica aqui uma chamada de atenção para ler e se precisar transpor em outro local as definições da documentação.

```
SELECT payment_status, COUNT(payment_status) AS contagem
FROM tb_payments payments
WHERE payment_status = "CHARGEBACK"
GROUP BY payment_status;
```

```
# payment status | contagem
# CHARGEBACK | 438
```

11- Qual foi o valor médio dos pagamentos cancelados (chargeback)?

```
SELECT ROUND(AVG(order_amount), 2) AS avg_chargeback, delivery_status
FROM tb_deliveries D INNER JOIN tb_orders O
WHERE delivery_status = "CANCELLED" AND D.delivery_order_id = O.delivery_order_id
GROUP BY delivery_status
```

```
# average | status
# 90.06 | CANCELLED
```

→ Correção

Pelo mesmo motivo do item 10, como o filtro foi feito pelo status **CANCELLED** e não do payment status **CHARGEBACK** os resultados não serão o esperado.

```
SELECT payment_status, ROUND(AVG(payment_amount), 2) AS contagem
FROM tb_payments payments
WHERE payment_status = "CHARGEBACK"
GROUP BY payment_status;
```

```
# payment_status | contagem
# CHARGEBACK | 16.35
```

12- Qual a média do valor de pagamento por método de pagamento (payment_method) em ordem decrescente?

```
SELECT payment_method, ROUND(AVG(payment_amount), 2) AS average_value
FROM tb_payments
GROUP BY payment_method
ORDER BY average_value DESC;
```

payment_method	average_value
INSTALLMENT_CREDIT_S	499.89
STORE_DIRECT_PAYMENT	281.78
VOUCHER_STORE	161.69
CREDIT_STORE	134.46
PAYMENT_LINK	122.76
ONLINE	101.98
CREDIT	97.12

payment_method	average_value
DEBIT_STORE	90.21
VOUCHER_DC	83.9
VOUCHER_OL	83.45
BANK_TRANSFER_DC	76.27
DEBIT	66.67
MEAL_BENEFIT	63.47
MONEY	33.89
VOUCHER	14.37

13- Quais métodos de pagamento tiveram valor médio superior a 100?

```
SELECT payment_method, ROUND(AVG(payment_amount), 2) AS average_value
FROM tb_payments
GROUP BY payment_method
HAVING average_value > 100
ORDER BY average_value DESC;
```

payment_method	average_value
INSTALLMENT_CREDIT_S	499.89
STORE_DIRECT_PAYMENT	281.78
VOUCHER_STORE	161.69
CREDIT_STORE	134.46
PAYMENT_LINK	122.76
ONLINE	101.98

14- Qual a média de valor de pedido (order_amount) por estado do hub (hub_state), segmento da loja (store_segment) e tipo de canal (channel_type)?

```
SELECT ROUND(AVG(order_amount),2) avg_order, channel_type, store_segment, hub_state
FROM tb_channels CHAN
JOIN tb_orders ORD ON CHAN.channel_id = ORD.channel_id
JOIN tb_stores STOR ON STOR.store_id = ORD.store_id
JOIN tb_hubs HUB ON HUB.hub_id = STOR.hub_id
GROUP BY channel_type, store_segment, hub_state
ORDER BY avg_order DESC
LIMIT 10;
```

avg_order	channel_type	store_segment	hub_state
421.04	OWN CHANNEL	GOOD	SP
358.66	OWN CHANNEL	GOOD	RS
233.56	OWN CHANNEL	GOOD	PR
211.41	OWN CHANNEL	GOOD	RJ
139.62	MARKETPLACE	GOOD	RJ
138.9	MARKETPLACE	GOOD	SP
135.02	MARKETPLACE	GOOD	PR
95.71	MARKETPLACE	FOOD	SP
86.99	OWN CHANNEL	FOOD	SP
81.55	MARKETPLACE	FOOD	RJ

15- Qual estado do hub (hub_state), segmento da loja (store_segment) e tipo de canal(channel_type) teve média de valor de pedido (order_amount) maior que 450?

Nenhuma.

```
# Adiciona o HAVING na query anterior

SELECT ROUND(AVG(order_amount),2) avg_order, channel_type, store_segment, hub_state
FROM tb_channels CHAN
JOIN tb_orders ORD ON CHAN.channel_id = ORD.channel_id
JOIN tb_stores STOR ON STOR.store_id = ORD.store_id
JOIN tb_hubs HUB ON HUB.hub_id = STOR.hub_id
GROUP BY channel_type, store_segment, hub_state WITH ROLLUP
HAVING avg_order > 450
ORDER BY avg_order DESC;
```

16- Qual o valor total de pedido (order_amount) por estado do hub (hub_state), segmento da loja (store_segment) e tipo de canal (channel_type)

Demonstre os totais intermediários e formate o resultado.

```
SELECT ROUND(SUM(order_amount),2), hub_state, store_segment, channel_type
FROM tb_orders ORDERS
INNER JOIN tb_stores STORES ON ORDERS.store_id = STORES.store_id
RIGHT JOIN tb_hubs HUBS ON HUBS.hub_id = STORES.hub_id
JOIN tb_channels CHANNELS ON ORDERS.channel_id = CHANNELS.channel_id
GROUP BY hub_state, store_segment, channel_type
ORDER BY hub_state;
```

total_order	hub_state	store_segment	channel_type
1335699.48	PR	FOOD	MARKETPLACE
24306.68	PR	FOOD	OWN CHANNEL
118544.08	PR	GOOD	MARKETPLACE
566142.84	PR	GOOD	OWN CHANNEL
9987907.75	RJ	FOOD	MARKETPLACE
349569.35	RJ	FOOD	OWN CHANNEL
444284.4	RJ	GOOD	MARKETPLACE
1532948.3	RJ	GOOD	OWN CHANNEL
2206522.86	RS	FOOD	MARKETPLACE
83950.95	RS	FOOD	OWN CHANNEL
44664.93	RS	GOOD	MARKETPLACE
658861.84	RS	GOOD	OWN CHANNEL
12751358.99	SP	FOOD	MARKETPLACE
287250.68	SP	FOOD	OWN CHANNEL
2298534.46	SP	GOOD	MARKETPLACE
6110183.21	SP	GOOD	OWN CHANNEL

→ Correção

A query está correta, porém em partes. Eu não soube interpretar o **TOTAIS INTERMÉDIOS** que era basicamente utilizar o ROLLUP para me trazer esse resultado de soma para todas as segmentações que estou fazendo.

```
SELECT
CASE
WHEN hub_state IS NULL THEN "Total Hub State" ELSE hub_state
END AS hub_state,
```

```

CASE
  WHEN store_segment IS NULL THEN "Total Segmento" ELSE store_segment
END AS store_segment,
CASE
  WHEN channel_type IS NULL THEN "Total tipo Canal" ELSE channel_type
END AS channel_type,
ROUND(SUM(order_amount),2) AS amount_total
FROM tb_orders ORDERS, tb_stores STORES, tb_hubs HUBS, tb_channels CHANNELS
WHERE ORDERS.store_id = STORES.store_id
AND HUBS.hub_id = STORES.hub_id
AND ORDERS.channel_id = CHANNELS.channel_id
GROUP BY hub_state, store_segment, channel_type WITH ROLLUP
ORDER BY hub_state;

```

17- Quando o pedido era do Hub do Rio de Janeiro (hub_state), segmento de loja 'FOOD', tipo de canal Marketplace e foi cancelado, qual foi a média de valor do pedido(order_amount)?

```

SELECT ROUND(AVG(order_amount),2) canceled_amount, hub_state, store_segment, channel_type
FROM tb_orders ORDERS
INNER JOIN tb_stores STORES ON ORDERS.store_id = STORES.store_id AND ORDERS.order_status = "CANCELED"
RIGHT JOIN tb_hubs HUBS ON HUBS.hub_id = STORES.hub_id AND HUBS.hub_state = "RJ" AND STORES.store_segment = "FOOD"
JOIN tb_channels CHANNELS ON ORDERS.channel_id = CHANNELS.channel_id AND CHANNELS.channel_type = "MARKETPLACE"
GROUP BY hub_state, store_segment, channel_type

```

canceled_amount	hub_state	store_segment	channel_type
66.49	RJ	FOOD	MARKETPLACE

18- Quando o pedido era do segmento de loja 'GOOD', tipo de canal Marketplace e foi cancelado, algum hub_state teve total de valor do pedido superior a 100.000?

```

SELECT ROUND(SUM(order_amount),2) canceled_amount, hub_state, store_segment, channel_type
FROM tb_orders ORDERS
INNER JOIN tb_stores STORES ON ORDERS.store_id = STORES.store_id AND ORDERS.order_status = "CANCELED"
RIGHT JOIN tb_hubs HUBS ON HUBS.hub_id = STORES.hub_id AND STORES.store_segment = "GOOD"
JOIN tb_channels CHANNELS ON ORDERS.channel_id = CHANNELS.channel_id AND CHANNELS.channel_type = "MARKETPLACE"
GROUP BY hub_state, store_segment, channel_type
HAVING canceled_amount > 100000

```

canceled_amount	hub_state	store_segment	channel_type
496669.48	SP	GOOD	MARKETPLACE

19- Em que data houve a maior média de valor do pedido (order_amount)? Dica:

Pesquise e use a função SUBSTRING().

Dados não estão completos, muitos valores NULL e incongruentes

```

SELECT ROUND(AVG(O.order_amount), 2) average_amount, SUBSTRING(order_moment_finished, 1, 9) order_date
FROM tb_orders_timers T, tb_orders O
WHERE t.order_id = O.order_id
GROUP BY order_date
ORDER BY average_amount DESC;

```

average_order	order_date
135.82	3/31/2021
127.86	3/13/2021
126.52	3/29/2021
124.41	4/3/2021

average_order	order_date
122.51	4/1/2021
122.3	4/30/2021
120.94	3/30/2021
120.28	3/27/2021
117.2	3/17/2021

→ Correção

No exercício 19 e 20 eu tive de criar uma table chamada **orders_timers** por conta de na importação inicial eu não ter identificado a necessidade dos horários e datas para a importação dos dados. Mas que acabou sendo necessário nestas ultimas duas questões. Me pareceu não muito eficiente importar 100% das colunas, e fiquei muito sujeito à erro por conta de Não trabalhar ainda com datas e também não conhecer a dica que deixei escrita no início do projeto em usar o Import Table Wizard para criar para mim as tables.

Minha query ela ficou correta, mas os timers não estão 100%.

20- Em quais datas o valor do pedido foi igual a zero (ou seja, não houve venda)?

```
SELECT SUBSTRING(order_moment_finished, 1, 9) order_date, MIN(O.order_amount) AS min_order
FROM tb_orders_timers T, tb_orders O
WHERE t.order_id = O.order_id
GROUP BY order_date
HAVING min_order = 0
ORDER BY order_date ASC;
```

order_date	min_order
2/2/2021	0
2/5/2021	0
2/6/2021	0
2/7/2021	0
2/13/2021	0
2/14/2021	0
2/15/2021	0
2/18/2021	0
2/20/2021	0
2/21/2021	0