

CASE SQL - IFOOD

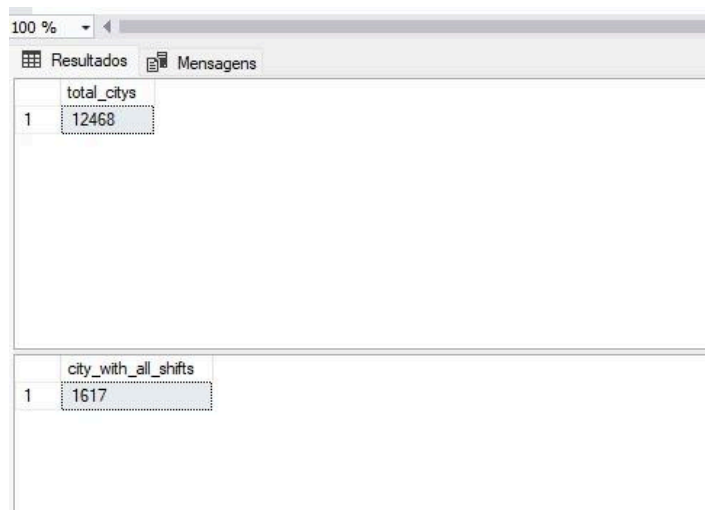
1 - Quantidade de Cidades do Brasil com Pedidos em Todos os Turnos

Escreva uma consulta que retorne a quantidade de cidades do Brasil que possuem pedidos realizados em todos os turnos disponíveis na base.

COMANDO:

```
SELECT COUNT(city_unique_id) AS city_with_all_shifts FROM [Growth & Business Intelligence _ Business Case Dataset] WHERE week_shift='all shifts combined'
```

RESULTADO:



	total_citys
1	12468

	city_with_all_shifts
1	1617

EXPLICAÇÃO:

Este código utiliza o comando **SELECT COUNT(city unique id)** para realizar uma contagem do número de cidades que possuem todos os turnos combinados na semana. Através da cláusula **WHERE**, ele filtra os registros onde o turno da semana é especificamente "all shifts combined" no conjunto de dados denominado "Growth & Business Intelligence _ Business Case Dataset". O resultado da contagem é nomeado como "city_with_all_shifts" a partir do comando **AS** após a contagem, que resultou no total de cidades que tenham registros em todos os shifts.

2 - Agrupamento de Turnos e Total de Pedidos

Escreva uma consulta que retorne o total de pedidos nos 5 turnos possíveis, dawn, breakfast, lunch, snack e dinner, com um campo que separe o weekday/weekend à parte do turno de shift atual.

COMANDO:

```
SELECT
    week_type,
    shift,
```

```

SUM(total_orders) AS total_orders
FROM [Growth & Business Intelligence _ Business Case Dataset]
WHERE week_shift <> 'all shifts combined'
GROUP BY shift, week_type
ORDER BY week_type DESC, total_orders DESC

```

RESULTADO:

	week_type	shift	total_orders
1	weekend	dinner	11727474
2	weekend	lunch	3561084
3	weekend	snack	888159
4	weekend	dawn	488674
5	weekend	breakfast	95073
6	weekday	dinner	11146359
7	weekday	lunch	8935389
8	weekday	snack	1707360
9	weekday	dawn	401820
10	weekday	breakfast	230624

EXPLICAÇÃO:

Este código realiza uma consulta em um conjunto de dados chamado "Growth & Business Intelligence _ Business Case Dataset". Ele seleciona três colunas específicas a partir do comando **SELECT**, sendo elas: **week_type**, **shift** e uma nova coluna gerada a partir do comando **SUM total_orders**, que posteriormente é nomeada como **total_orders**.

Logo após, foi utilizada a cláusula **WHERE** para filtrar os resultados onde **week_shift** é diferente (<>) de 'all shifts combined'.

Seguindo, foi utilizada o comando **GROUP BY** para organizar os resultados e agrupá-los por **shift** e **week_type**, ou seja, agrupa as informações de acordo com os diferentes turnos e tipos de dias da semana(**weekday** e **weekend**)..

Por fim, a cláusula **ORDER BY** faz uma classificação dos resultados, utilizando, primeiramente a coluna **week_type** em ordem decrescente e depois pela coluna **total_orders** em ordem decrescente. Isso significa que os resultados serão organizados primeiro de acordo com o tipo de semana e dentro de cada tipo de semana, os totais de pedidos serão apresentados em ordem decrescente, do maior para o menor.

3 - Ranqueamento de Cidades com Mais Pedidos e Turnos com Mais Pedidos

Escreva uma consulta que retorne uma lista com um ranking de cidades com mais pedidos realizados, além de um ranqueamento dentro das cidades em relação aos seus business_models. Filtre as top 5 cidades.

COMANDO:

```
SELECT
    city_unique_id,
    business_model,
    SUM(total_orders) AS total_orders,
    RANK() OVER (PARTITION BY business_model ORDER BY SUM(total_orders) DESC)
AS ranking_pedidos,
    RANK() OVER (PARTITION BY city_unique_id ORDER BY SUM(total_orders) DESC) AS
ranking_bm_interno
FROM
    [Growth & Business Intelligence _ Business Case Dataset]
GROUP BY
    city_unique_id, business_model
HAVING
    city_unique_id IN (
        SELECT TOP 5 city_unique_id
        FROM [Growth & Business Intelligence _ Business Case Dataset]
        GROUP BY city_unique_id
        ORDER BY SUM(total_orders) DESC
    )
ORDER BY
    ranking_pedidos ASC, ranking_bm_interno ASC
```

RESULTADO:

	city_unique_id	business_model	total_orders	ranking_pedidos	ranking_bm_interno
1	3550308	MARKETPLACE	6942230	1	1
2	3550308	FULL_SERVICE	5709464	1	2
3	3550308	HYBRID	233976	1	3
4	3304557	MARKETPLACE	4639852	2	1
5	3304557	FULL_SERVICE	1911158	2	2
6	3304557	HYBRID	77266	2	3
7	5300108	FULL_SERVICE	1540718	3	1
8	2304400	MARKETPLACE	1413130	3	1
9	3106200	HYBRID	35778	3	3
10	3106200	MARKETPLACE	1256220	4	1
11	3106200	FULL_SERVICE	1225628	4	2
12	5300108	HYBRID	34034	4	3
13	2304400	FULL_SERVICE	807830	5	2
14	5300108	MARKETPLACE	1224564	5	2
15	2304400	HYBRID	23184	5	3

EXPLICAÇÃO:

Este código faz uma análise do conjunto de dados "Growth & Business Intelligence _ Business Case Dataset". A consulta começa selecionando três colunas a partir do comando **SELECT**, sendo elas: **city_unique_id**, **business_model** e a soma de **total_orders**, nomeada como **total_orders**.

Em seguida, são calculados dois rankings utilizando a função **RANK() OVER()**. O primeiro ranking, **ranking_pedidos**, é determinado pela soma total de pedidos agrupados por **business_model**, classificando esses valores de forma descendente. O segundo ranking, **ranking_bm_interno**, é calculado pela soma total de pedidos agrupados pela **city_unique_id**, também classificando esses valores de forma descendente.

A cláusula **GROUP BY** foi utilizada para agrupar os resultados por **city_unique_id** e **business_model**, o que significa que os valores são calculados por cada combinação de ID's de cidade e modelo de negócio.

Logo após, foi utilizada a cláusula **HAVING** para filtrar apenas as top 5 cidades com base na soma total de pedidos. Isto é feito utilizando uma subconsulta que utiliza um **SELECT** com as top 5 **city_unique_id** ordenados pela soma (**SUM**) do total de pedidos.

Por fim, os resultados são ordenados pelo **ranking_pedidos** e pelo **ranking_bm_interno**, ambos de forma ascendente. Isso permite uma visualização ordenada e compreensível dos dados, destacando as cidades com o maior volume de pedidos e os modelos de negócio mais requisitados dentro de cada uma delas.