

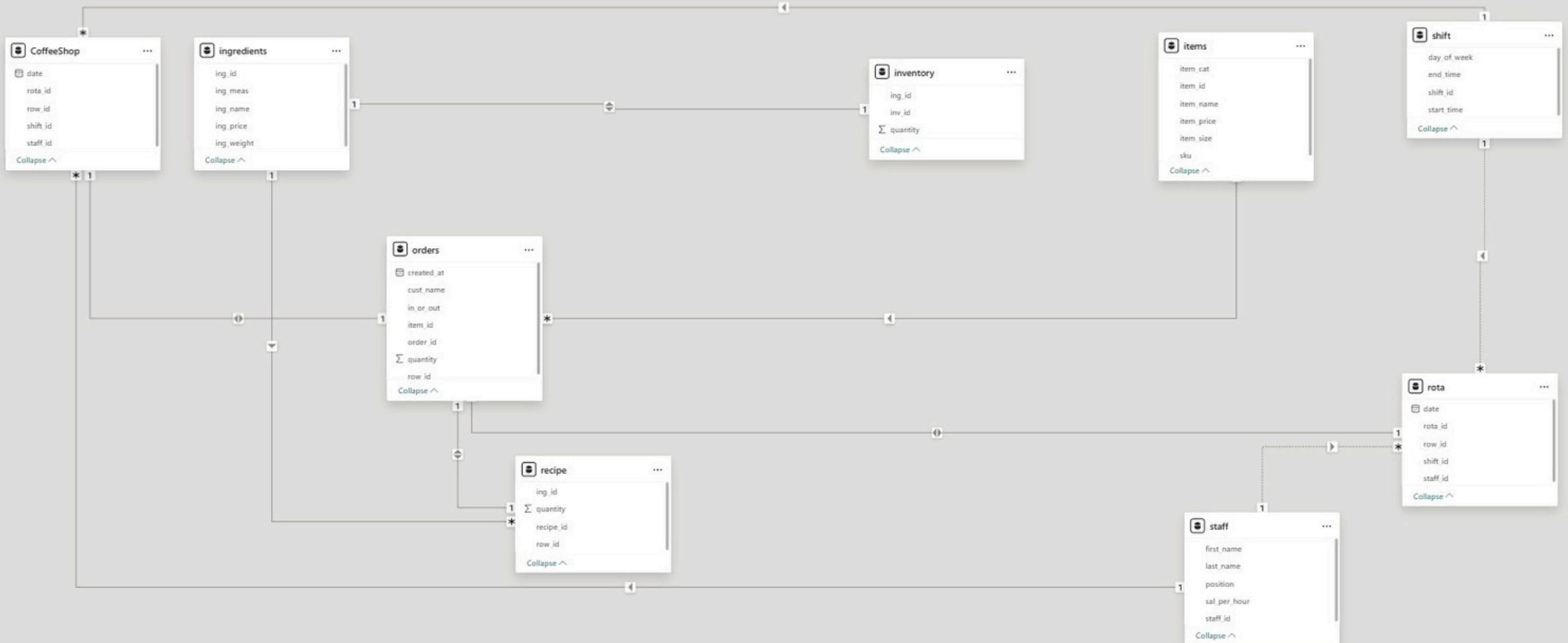
MY SQL COFFEE CHALLENGE



in www.linkedin.com/in/subhapriya-sanki



CARDINALITY



PROBLEM STATEMENT TO ANALYSIS

- Total Orders: Counted all customer orders to gauge business activity.
- Total Sales: Calculated the total revenue generated.
- Total Items: Summarized the variety and number of items sold.
- Average Order Value: Determined the average revenue per order.
- Sales by Category: Analyzed revenue generation by item category.
- Top Selling Items: Identified the most popular items.
- Orders by Hour: Examined the distribution of orders throughout the day.
- Sales by Hour: Analyzed hourly revenue trends.
- Orders In or Out: Differentiated between dine-in and takeout orders.
- Total Quantity by Ingredient: Calculate the total usage of each ingredient.

- **Total Cost of Ingredients:** Estimated the overall cost of ingredients used.
- **Calculate Cost of Coffee:** Determined the cost to produce each coffee item.
- **Percentage Stock Remaining by Ingredients:** Assessed stock levels as a percentage of total capacity.
- **List of Ingredients to Re-order:** Identified ingredients needing replenishment based on inventory levels.
- **Total Staff Cost:** Calculated the total expenditure on staff salaries.
- **Total Hours Worked:** Summed up the hours staff worked.
- **Hours Worked by Staff Member:** Broke down hours worked by individual employees.
- **Cost per Staff Member:** Analysed salary expenses per employee.



Coff... ▾ C +

↔ Query #1 ●

↔ Query #2 ●

+ ▾

★ Upgrade

Filter



ENTITIES 9



items

item_id	text
sku	text
item_name	text
item_cat	text
item_size	text
item_price	do...

orders

row_id	int
order_id	text
created_at	text
item_id	text
quantity	int
cust_name	text
in_or_out	text

recipe

row_id	int
recipe_id	text
ing_id	text

```
1 #Q1 • Total Orders: Counted all customer orders to gauge business activity.  
2  
3     Select Count(*)as TOTAL_ORDERS from orders  
4 where order_id is not null;  
5 # ANS : 521  
6  
7     #Q2 Total Sales: Calculated the total revenue generated.  
8  
9 with Cte1 as ( select A.item_id ,A.item_price , B.quantity  
10                 from items A JOIN orders B  
11                   ON A.item_id = B.item_id )  
12  
13 select ROUND(SUM(CTE1.item_price * CTE1.quantity),2) as TOTAL_REV from CTE1;  
14
```

Save

Run ▾

TOTAL_REV

1 1857.45



Coff... ▾

Filter

ENTITIES 9

order_id text

quantity int

items ▾

item_id text

sku text

item_name text

item_cat text

item_size text

item_price do...

orders ▾

row_id int

order_id text

created_at text

item_id text

quantity int

cust_name text

in_or_out text

recipe ▾

row_id int

↔ Query #1 • ↔ Query #2 • ↔ Query #3 • + ▾

★ Upgrade

```
1 # 3. • Average Order Value: Determined the average revenue per order.  
2  
3 with CTE1 as ( select A.order_id , AVG(A.quantity * B.item_price) as REVENUE  
4             from orders A JOIN items B  
5                 ON A.item_id = B.item_id  
6             GROUP BY order_id )  
7 Select * from CTE1  
8
```

Save Run ▾

	order_id	REVENUE
1	ORD001	2.15
2	ORD002	4.6
3	ORD003	2.15
4	ORD004	3.25
5	ORD005	5.5
6	ORD006	3.6
7	ORD007	3.35
8	ORD008	4.4
9	ORD009	5.05





Coff... ▾



Filter



ENTITIES 9

my_db text

quantity int

items

item_id text

sku text

item_name text

item_cat text

item_size text

item_price do...

orders

row_id int

order_id text

created_at text

item_id text

quantity int

cust_name text

in_or_out text

recipe

row_id int

recipe_id text

↔ Query #1 ● ↔ Query #2 ● ↔ Query #3 ● ↔ Query #4 ● ↔ Query #5

★ Upgrade

```
1 #4.Sales by Category: Analyzed revenue generation by item category.
2 #5• Top Selling Items: Identified the most popular items.
3 With CTE1 as ( select A.item_name , SUM(A.item_price * B.quantity) as REVENUE_GEN
4   FROM items A JOIN orders B
5     ON A.item_id = B.item_id
6     GROUP BY item_name
7       ORDER BY REVENUE_GEN DESC )
8
9 SELECT CTE1.item_name , ROUND(CTE1.REVENUE_GEN,2) FROM CTE1;
```

Save

Run

	item_name	ROUND(CTE1.REVENUE_GEN,2)
--	-----------	---------------------------

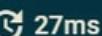
1	Cold Mocha	204.2
2	White Mocha	197.5
3	Caramel Macchiato	162.2
4	Mocha	159.2
5	Hot Chocolate	155.8
6	Cappuccino	143.7
7	Lemonade	142.4



localhost:3306/CoffeeSh



14 0 affected



27ms

Download ▾





Coff... ▾ C +

Filter ▾



ENTITIES 9

" " " " " "

inventory

inv_id text

ing_id text

quantity int

items

item_id text

sku text

item_name text

item_cat text

item_size text

item_price do...

orders

row_id int

order_id text

created_at text

item_id text

quantity int

cust_name text

in_or_out text

<> Query #1 ● <> Query #2 ● <> Query #3 ● <> Query #4 ● <> Query #5 ● + ▾

★ Upgrade

```
1 #6• Orders by Hour: Examined the distribution of orders throughout the day.  
2  
3 with CTE1 as (  
4     select count(order_id) as DISTRIBUTION, hour(created_at) as HH from orders  
5         where in_or_out = "out"  
6             group by HH )  
7 select CTE1.HH , CTE1.DISTRIBUTION from CTE1;  
8  
9 with CTE1 as (  
10    select count(order_id) as DISTRIBUTION, hour(created_at) as HH from orders  
11        group by HH )  
12 select CTE1.HH , CTE1.DISTRIBUTION from CTE1;  
13  
14
```

Save

Run ▾

HH ▾ DISTRIBUTION ▾

1	7	33
2	8	30
3	9	28
4	10	12
5	11	11
6	12	35



Coff... ▾

Filter ▾

<> Query # ● + ▾ ★ Upgrade

ENTITIES 9

items

item_id	text
sku	text
item_name	text
item_cat	text
item_size	text
item_price	do...

orders

row_id	int	HH ▲	TRENDS ▲
order_id	text	1	12 290
created_at	text	2	7 280
item_id	text	3	13 279
quantity	int	4	8 262
cust_name	text	5	9 219
in_or_out	text	6	14 159
recipe		7	15 100

Save

Run ▾





CoffeeShop



Filter



ENTITIES 9



item_cat



item_size



item_price



orders

row_id



order_id



created_at



item_id



quantity



cust_name



in_or_out



recipe

row_id



recipe_id



ing_id



quantity



rota

row_id



rota_id



date



shift_id



<> Query 1 <> Query 2 <> Query 3 <> Query 4 <> Query 5 <> Query 6 <> Query 7 <> Query 8 Upgrade

```
1 # 8 Orders In or Out: Differentiated between dine-in and takeout orders.  
2  
3 WITH CTE1 AS ( SELECT order_id,  
4 case  
5     WHEN in_or_out IS NULL THEN "NULL"  
6     WHEN TRIM(in_or_out) ="" THEN "EMPTY"  
7     ELSE in_or_out  
8 END AS RESULTS  
9 FROM orders )  
10  
11 SELECT RESULTS ,COUNT(CTE1.order_id) as ORDERS_CNT  
12 FROM CTE1  
13 GROUP BY RESULTS  
14
```

Save

Run



	RESULTS	ORDERS_CNT
1	out	228
2	in	203
3	EMPTY	90



CoffeeShop ▾ C +

Filter

1 #9 Total Quantity by Ingredient: Calculate the total usage of each ingredient.
2
3 select ing_meas , ing_name ,SUM(ing_weight) as TTL_WEIGHT
4 from ingredients
5 GROUP BY ing_meas , ing_name
6 order by ing_meas

7
8
9
10
11

ENTITIES 9

+

rota_id text

date text

shift_id text

staff_id text

▼ ingredients

ing_id text

ing_name text

ing_weight int

ing_meas text

ing_price double

▼ inventory

inv_id text

ing_id text

quantity int

▼ items

item_id text

sku text

item_name text

item_cat text

item_size text

Save

Run

	ing_meas	ing_name	TTL_WEIGHT
1	grams	Black Tea	1000
2	grams	Cheddar	500
3	grams	Chocolate	1000
4	grams	Cocoa powder	1000
5	grams	Espresso beans	1000
6	grams	Ham	1000
7	grams	Mozzarella	500



CoffeeShop ▾ C +

<> ● <> (●) + ▾ ★ Upgrade

Filter



ENTITIES 9



rota_id

text

date

text

shift_id

text

staff_id

text

▼ ingredients

ing_id

text

ing_name

text

ing_weight

int

ing_meas

text

ing_price

double

▼ inventory

inv_id

text

ing_id

text

quantity

int

▼ items

item_id

text

sku

text

item_name

text

item_cat

text

item_size

text

```
1 #10 Total Cost of Ingredients: Estimated the overall cost of ingredients used.  
2  
3 SELECT  
4     ing_name,  
5     SUM(CASE WHEN ing_meas = 'grams' THEN ing_price ELSE 0 END) AS total_price_grams,  
6     SUM(CASE WHEN ing_meas = 'units' THEN ing_price ELSE 0 END) AS total_price_units,  
7     SUM(CASE WHEN ing_meas = 'ml' THEN ing_price ELSE 0 END) AS total_price_ml  
8 FROM ingredients  
9 GROUP BY ing_name;
```

10
11

Save

Run

	ing_name	total_price_grams	total_price_units	total_price_ml
1	Espresso beans	12	0	0
2	Whole Milk	0	0	1.2
3	Cheddar	7.45	0	0
4	Mozzarella	5	0	0
5	Whipped cream	0	0	1.35
6	Vanilla syrup	0	0	14.52
7	Barista chocolate syrup	0	0	8.49



CoffeeShop ▾ C +

Filter



ENTITIES 9



▼ ingredients

ing_id	text
ing_name	text
ing_weight	int
ing_meas	text
ing_price	double

▼ inventory

inv_id	text
ing_id	text
quantity	int

▼ items

item_id	text
sku	text
item_name	text

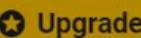
item_cat	text
item_size	text
item_price	double

▼ orders

row_id	int
order_id	text



+ ▾



1 #11 Calculate Cost of Coffee: Determined the cost to produce each coffee item.

2

3

4 select item_name , SUM(item_price) as COST from items
5 group by item_name

6

7

8

Save

Run



item_name ▾ COST ▾

1	White Mocha	9.2
2	Caramel Macchiato	8.8
3	Hot Chocolate	8.8
4	Mocha	8.6
5	Cold Mocha	8.6
6	Cappuccino	7.2
7	Latte	7.2
8	Cold Coffee	7.2
9	Lemonade	7.1



localhost:3306/CoffeeShop my

14

0 affected

27ms

Download





CoffeeShop ▾



Filter



ENTITIES 9



coffeshoptb

row_id

int

rota_id

text

date

text

shift_id

text

staff_id

text

ingredients

ing_id

text

ing_name

text

ing_weight

int

ing_meas

text

ing_price

double

inventory

inv_id

text

ing_id

text

quantity

int

items

item_id

text

sku

text

item_name

text



★ Upgrade

```
1 #12 Percentage Stock Remaining by Ingredients:  
2   #Assessed stock levels as a percentage of total capacity.  
3  
4 select * from ingredients;  
5 select * from inventory;  
6  
7 select A.ing_name , A.ing_meas, A.ing_weight * B.quantity as STOCK_LEFT  
8   from ingredients A JOIN inventory B  
9   ON A.ing_id = B.ing_id
```

	ing_name	ing_meas	STOCK_LEFT
1	Espresso beans	grams	4000
2	Whole Milk	ml	55000
3	Cheddar	grams	500
4	Mozzarella	grams	2000
5	Whipped cream	ml	2100
6	Vanilla syrup	ml	3000
7	Barista chocolate syrup	ml	3000

Save

Run ▾



> coffeeShoptb

✓ ingredients

ing_id	text
ing_name	text
ing_weight	int
ing_meas	text
ing_price	double

✓ inventory

inv_id	text
ing_id	text
quantity	int

> items

> orders

> recipe

> rota

> shift

> staff



```
1 #13 List of Ingredients to Re-order:  
2   #Identified ingredients needing replenishment based on inventory levels.  
3  
4   SELECT * from ingredients; // SELECT * FROM inventory;  
5  
6   with CTE1 as (  
7     SELECT A.ing_name , A.ing_meas ,A.ing_weight * B.quantity as STOCK_LV  
8     FROM ingredients A JOIN inventory B  
9     ON A.ing_id = B.ing_id ),  
10  
11  CTE2 as ( SELECT CTE1.ing_name,CTE1.ing_meas, CTE1.STOCK_LV,  
12    RANK() OVER( PARTITION BY CTE1.ing_meas ORDER BY CTE1.STOCK_LV) as RNK  
13    from CTE1 )  
14  
15  select * from CTE2 where RNK in (1,2);
```

Save

Run

	ing_name	ing_meas	STOCK_LV	RNK
4	Salami	grams	2000	2
5	Black Tea	grams	2000	2
6	Vanilla extract	ml	120	1
7	Barista caramel sauce	ml	1000	2
8	Lemons	units	50	1
9	Panini Bread	units	80	2



CoffeeShop

▼ C +

<> Query #1 ● <> Query #2 <> Query #3 ● <> Query #4 ● <> Query #5 ● + ▾

Upgrade

Filter



ENTITIES 9



- shift_id
- staff_id
- > ingredients
- > inventory
- > items
- > orders
- > recipe
- > rota
- ▼ shift

text

double

```
1 # 14 Total Staff Cost: Calculated the total expenditure on staff salaries.
2 # 15 Total Hours Worked: Summed up the hours staff worked.
3 #17 Cost per Staff Member: Analysed salary expenses per employee.
4
5 select * from staff; // select * from shift; // select * from coffeeshoptb;
6
7 WITH CTE1 as ( SELECT A.staff_id , A.first_name , A.last_name ,A.position , A.sal_per_hour , B.shift_id
8
9         from staff A JOIN coffeeshoptb B
10        ON A.staff_id = B.staff_id ),
11      cte2 as ( SELECT CTE1.first_name , CTE1.last_name , CTE1.sal_per_hour ,   C.end_time - C.start_time as T_TIME
12
13         from CTE1 JOIN shift C
14        ON CTE1.shift_id = C.shift_id )
15
16 select Cte2.first_name , CTE2.last_name , sum(CTE2.sal_per_hour * CTE2.T_TIME) from cte2
17 group by Cte2.first_name , CTE2.last_name
```

Save

Run ▾

	first_name	last_name	sum(CTE2.sal_per_hour * CTE2.T_TIME)
1	Olivia	Williams	220
2	Emma	Johnson	220
3	Noah	Brown	260
4	Liam	Smith	260

localhost:3306/CoffeeShop - Beekeeper Studio - Free Version

File Edit View Tools Help

CoffeeShop C +

Filter

ENTITIES 9

- shift_id
- staff_id
- ingredients
- inventory
- items
- orders
- recipe
- rota
- shift
 - shift_id
 - day_of_week
 - start_time
 - end_time
- staff
 - staff_id
 - first_name
 - last_name
 - position
 - sal_per_hour

<> Query #1 ● <> Query #2 ● <> Query #3 ● <> Query #4 ● + ▾

14 (BACK TESTING) Total Staff Cost: Calculated the total expenditure on staff salaries.

```
1 # 14 (BACK TESTING ) Total Staff Cost: Calculated the total expenditure on staff salaries.
2
3
4 WITH CTE1 as ( SELECT A.staff_id , A.first_name , A.last_name ,A.position , A.sal_per_hour , B.shift_id
5                               from staff A JOIN coffeeshoptb B
6                               ON A.staff_id = B.staff_id ),
7
8     cte2 as ( SELECT CTE1.first_name , CTE1.last_name , CTE1.sal_per_hour ,    C.end_time - C.start_time as T_TIME
9                               from CTE1 JOIN shift C
10                              ON CTE1.shift_id = C.shift_id )
11
12 select Cte2.first_name , CTE2.last_name , CTE2.sal_per_hour * CTE2.T_TIME from cte2
13 where Cte2.first_name = "Emma" and CTE2.last_name = "Johnson";
14
15
```

Save Run ▾

	first_name	last_name	CTE2.sal_per_hour * CTE2.T_TIME
1	Emma	Johnson	60
2	Emma	Johnson	40
3	Emma	Johnson	60
4	Emma	Johnson	60

localhost:3306/CoffeeShop mysql 4 0 affected 11ms Download ▾



CoffeeShop Filter

ENTITIES 9

- shift_id text
- staff_id text
- ingredients
- inventory
- items
- orders
- recipe
- rota
- shift
 - shift_id text
 - day_of_week text
 - start_time text
 - end_time text
- staff
 - staff_id text
 - first_name text
 - last_name text
 - position text
 - sal_per_hour double

<> Query #1 ● <> Query #2 <> Query #3 ● <> Query #4 ● <> Query #5 ● + ▾

Upgrade

```
1 #16 Hours Worked by Staff Member: Broke down hours worked by individual employees.
2 WITH CTE1 as ( SELECT A.staff_id , A.first_name , A.last_name ,A.position , A.sal_per_hour , B.shift_id
3
4
5
6     cte2 as ( SELECT CTE1.first_name , CTE1.last_name , CTE1.sal_per_hour ,    C.end_time - C.start_time as T_TIME
7
8         from CTE1 JOIN shift C
9
10        ON CTE1.shift_id = C.shift_id )
11
12
13 select Cte2.first_name , CTE2.last_name ,sum(CTE2.T_TIME) as TTL_time from cte2
14 group by Cte2.first_name , CTE2.last_name ;
```

Save

Run

	first_name	last_name	TTL_time
1	Olivia	Williams	22
2	Emma	Johnson	22
3	Noah	Brown	26
4	Liam	Smith	26

THANKS FOR WATCHING

We will grow together



in www.linkedin.com/in/subhapriya-sanki

