

Project : Data cleaning

Name : Akshay Gaur

Email: akshay.gaur028@gmail.com

Batch: 6

Phone: 9311211182

◆ STEP 0: Inspect Raw Data

```
SELECT *
FROM customer_orders
LIMIT 10;
```

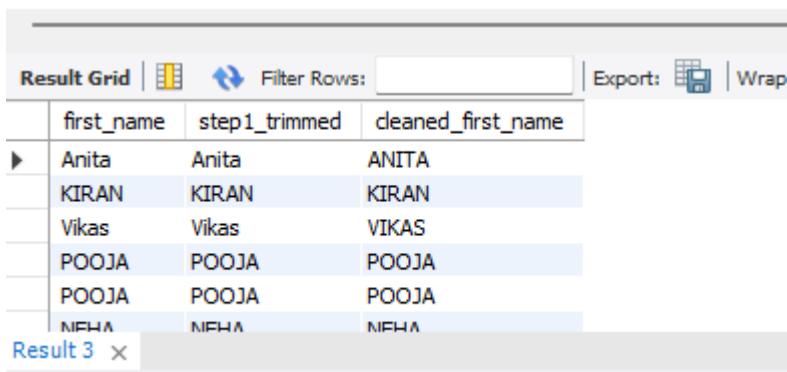
```
3 •  select *
4   from customer_orders
5   limit 10;
```

Result Grid												
<input type="checkbox"/> Filter Rows: <input type="button"/> Export: <input type="button"/> Wrap Cell Content: <input type="checkbox"/> Fetch rows: <input type="button"/>												
customer_id	first_name	last_name	email	mobile_number	order_id	order_date	delivery_date	order_amount	city	signup_date	rating	
1001	Anita	SHARMA	anita1@GMAIL.COM	919110053353	ORD-2022-0001	2022-07-18	2022-07-20	3662.377	PUNE	2021-11-22	1.882	
1002	KIRAN	Gupta	kiran2@GMAIL.COM	0091-9749621470	ORD-2021-0002	2021-01-14	2021-01-21	10669.923	delhi	2018-10-02	4.892	
1003	Vikas	PATEL	vikas3@yahoo.com	0091-9664130526	ORD-2024-0003	2024-02-05	2024-02-08	23175.878	bangalore	2021-09-29	1.651	
1004	POOJA	SINGH	pooja4@yahoo.com	919654049436	ORD-2022-0004	2022-07-01	2022-07-07	38255.816	MUMBAI	2020-07-22	4.287	
1005	POOJA	VERMA	pooja5@yahoo.com	919940992571	ORD-2022-0005	2022-07-03	2022-07-04	91497.485	hyderabad	2021-11-19	2.598	
1006	NEHA	Kumar	neha6@GMAIL.COM	9010811514014	ORD-2023-0006	2023-11-10	2023-11-14	50847.600	CHENNAI	2022-07-30	3.883	

◆ STEP 1: Clean `first_name` (Spaces + Case)

```
SELECT
    first_name,
    TRIM(first_name) AS step1_trimmed,
    UPPER(TRIM(first_name)) AS cleaned_first_name
FROM customer_orders;
```

```
7 •   select
8     first_name,
9     trim(first_name) as step1_trimmed,
10    upper(trim(first_name)) as cleaned_first_name
11  from customer_orders;
```

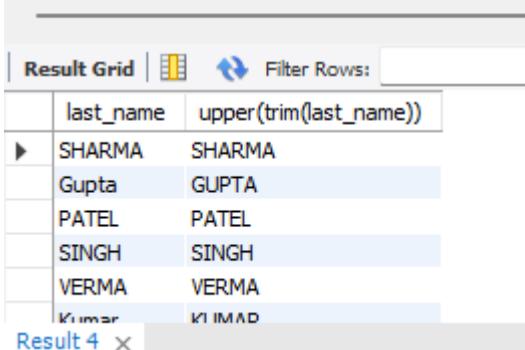


	first_name	step1_trimmed	cleaned_first_name
▶	Anita	Anita	ANITA
	KIRAN	KIRAN	KIRAN
	Vikas	Vikas	VIKAS
	POOJA	POOJA	POOJA
	POOJA	POOJA	POOJA
	NFHA	NFHA	NFHA

◆ STEP 2: Clean last_name

```
SELECT
  last_name,
  UPPER(TRIM(last_name)) AS cleaned_last_name
FROM customer_orders;
```

```
13 •   select
14     last_name,
15     upper(trim(last_name))
16  from customer_orders;
```



	last_name	upper(trim(last_name))
▶	SHARMA	SHARMA
	Gupta	GUPTA
	PATEL	PATEL
	SINGH	SINGH
	VERMA	VERMA
	Kumar	KUMAR

- ◆ **STEP 3: Create `full_name` (CONCAT)**

```
SELECT
    CONCAT(
        UPPER(TRIM(first_name)),
        ' ',
        UPPER(TRIM(last_name))
    ) AS full_name
FROM customer_orders;
```

```
18 •   select
19   ◉ concat(
20     upper(trim(first_name)),
21     ' ',
22     upper(trim(last_name))
23   ) as full_name
24   from customer_orders;
```

The screenshot shows a database interface with a code editor at the top and a result grid below. The code editor contains the SQL query for creating a full name column. The result grid, titled 'Result Grid', displays a single column named 'full_name' with six rows of data: ANITA SHARMA, KIRAN GUPTA, VIKAS PATEL, POOJA SINGH, POOJA VERMA, and NEHA KHIMAD.

full_name
ANITA SHARMA
KIRAN GUPTA
VIKAS PATEL
POOJA SINGH
POOJA VERMA
NEHA KHIMAD

- ◆ **STEP 4: Clean `email` (Standardization)**

```
SELECT
    email,
    LOWER(email) AS cleaned_email
FROM customer_orders;
```

```
26 •   select
27     email,
28     lower(email) as cleaned_email
29     from customer_orders;
```

Result Grid	
	email
▶	anita1@GMAIL.COM
	kiran2@GMAIL.COM
	vikas3@yahoo.com
	pooja4@yahoo.com
	pooja5@yahoo.com
	neha6@GMAIL.COM

Result 7 x

◆ STEP 5: Clean mobile_number (Extract last 10 digits)

```
SELECT
    mobile_number,
    SUBSTR(mobile_number, LENGTH(mobile_number) - 9, 10) AS cleaned_mobile
FROM customer_orders;
```

```
31 •   select
32     mobile_number,
33     substr(mobile_number, length(mobile_number) - 9, 10) as cleaned_mobile_number
34     from customer_orders;
```

Result Grid	
	mobile_number
▶	919110053353
	0091-9749621470
	0091-9664130526
	919654049436
	919940992571
	+91-9811514014

customer_orders 8 Result 9 x

◆ STEP 6: Extract Year from order_id

```
SELECT
```

```
order_id,  
SUBSTR(order_id, 5, 4) AS order_year  
FROM customer_orders;
```

```
36 •  select  
37      order_id,  
38      substr(order_id, 5, 4) as order_year  
39      from customer_orders;
```

Result Grid		Filter Rows:	Export:
	order_id	order_year	
▶	ORD-2022-0001	2022	
	ORD-2021-0002	2021	
	ORD-2024-0003	2024	
	ORD-2022-0004	2022	
	ORD-2022-0005	2022	
	ORD-2023-0006	2023	
	customer_orders 8	Result 10 ×	

◆ STEP 7: Round order_amount

```
SELECT  
      order_amount,  
      ROUND(order_amount, 2) AS cleaned_order_amount  
FROM customer_orders;
```

```
41 •  select  
42      order_amount,  
43      round(order_amount, 2) as cleaned_order_amount  
44      from customer_orders;
```

Result Grid		Filter Rows:	Export:
	order_amount	cleaned_order_amount	
▶	3662.377	3662.38	
	10669.923	10669.92	
	23175.878	23175.88	
	38255.816	38255.82	
	91497.485	91497.48	
	50847.600	50847.7	
	customer_orders 8	Result 11 ×	

- ◆ **STEP 8: Round rating**

```
SELECT
    rating,
    ROUND(rating, 1) AS cleaned_rating
FROM customer_orders;
```

```
46 •  select
47    rating,
48    round(rating, 1) as cleaned_rating
49    from customer_orders;
```

Result Grid | Filter Rows: Export:

	rating	cleaned_rating
▶	1.882	1.9
	4.892	4.9
	1.651	1.7
	4.287	4.3
	2.598	2.6
	3.883	3.9
	customer_orders	8

Result 12 ×

- ◆ **STEP 9: Standardize city**

```
SELECT
    city,
    UPPER(city) AS cleaned_city
FROM customer_orders;
```

```
51 •   select
52     city,
53     upper(city) as cleaned_city
54   from customer_orders;
```

Result Grid		
	city	cleaned_city
▶	PUNE	PUNE
	delhi	DELHI
	bangalore	BANGALORE
	MUMBAI	MUMBAI
	hyderabad	HYDERABAD
	CHEMNAT	CHEMNAT

customer_orders 8 Result 13 ×

◆ STEP 10: Delivery Time Calculation (DATEDIFF)

```
SELECT
    order_date,
    delivery_date,
    DATEDIFF(delivery_date, order_date) AS delivery_days
FROM customer_orders;
```

```
56 •   select
57     order_date,
58     delivery_date,
59     datediff(delivery_date, order_date) as delivery_days
60   from customer_orders;
```

Result Grid			
	order_date	delivery_date	delivery_days
▶	2022-07-18	2022-07-20	2
	2021-01-14	2021-01-21	7
	2024-02-05	2024-02-08	3
	2022-07-01	2022-07-07	6
	2022-07-03	2022-07-04	1
	2023-11-10	2023-11-14	4

customer_orders 8 Result 17 ×

◆ STEP 11: Customer Tenure Calculation

```
SELECT
    signup_date,
    DATEDIFF(NOW(), signup_date) AS days_with_company
FROM customer_orders;
```

```
62 •   select
63     signup_date,
64     datediff(now(), signup_date) as days_with_company
65     from customer_orders
66     order by days_with_company desc;
```

The screenshot shows a MySQL Workbench result grid with the following data:

	signup_date	days_with_company
▶	2018-10-02	2644
	2018-10-22	2624
	2018-11-01	2614
	2018-11-17	2598
	2018-12-02	2583
	2018-12-11	2574

customer_orders 8 Result 19 ×

◆ STEP 12: CASE WHEN – Order Value Category

```
SELECT
    order_amount,
    CASE
        WHEN order_amount >= 50000 THEN 'High Value'
        WHEN order_amount >= 20000 THEN 'Medium Value'
        ELSE 'Low Value'
    END AS order_category
FROM customer_orders;
```

```
68 •   select
69     order_amount,
70     case
71         when order_amount >= 50000 then 'High Value'
72         when order_amount >= 20000 then 'Medium Value'
73         else 'Low Value'
74     end as order_category
75     from customer_orders;
```

	order_amount	order_category
▶	3662.377	Low Value
	10669.923	Low Value
	23175.878	Medium Value
	38255.816	Medium Value
	91497.485	High Value
	50847.600	High Value

◆ STEP 13: CASE WHEN – Customer Type

```
SELECT
    signup_date,
    CASE
        WHEN DATEDIFF(NOW(), signup_date) <= 30 THEN 'New'
        WHEN DATEDIFF(NOW(), signup_date) <= 180 THEN 'Regular'
        ELSE 'Loyal'
    END AS customer_type
FROM customer_orders;
```

```

77 •   select
78     signup_date,
79     case
80         when datediff(now(), signup_date) <= 30 then "New"
81         when datediff(now(), signup_date) <= 180 then "Regular"
82         else "Loyal"
83     end as customer_type
84   from customer_orders;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	signup_date	customer_type
▶	2021-11-22	Loyal
	2018-10-02	Loyal
	2021-09-29	Loyal
	2020-07-22	Loyal
	2021-11-19	Loyal
	2023-07-30	Loyal

customer_orders 8 Result 21 ×

◆ STEP 14: FINAL CLEANED VIEW (Industry Practice)

```

CREATE VIEW customer_orders_cleaned AS
SELECT
    customer_id,
    UPPER(TRIM(first_name)) AS first_name,
    UPPER(TRIM(last_name)) AS last_name,
    CONCAT(
        UPPER(TRIM(first_name)), ' ',
        UPPER(TRIM(last_name)))
    ) AS full_name,
    LOWER(email) AS email,
    SUBSTR(mobile_number, LENGTH(mobile_number) - 9, 10) AS mobile_number,
    order_id,
    SUBSTR(order_id, 5, 4) AS order_year,
    order_date,
    delivery_date,

```

```

DATEDIFF(delivery_date, order_date) AS delivery_days,
ROUND(order_amount, 2) AS order_amount,
UPPER(city) AS city,
signup_date,
DATEDIFF(NOW(), signup_date) AS customer_tenure_days,
CASE
    WHEN order_amount >= 50000 THEN 'High Value'
    WHEN order_amount >= 20000 THEN 'Medium Value'
    ELSE 'Low Value'
END AS order_category,
ROUND(rating, 1) AS rating
FROM customer_orders;

```

◆ STEP 15: Validate Cleaned Data

```

SELECT *
FROM customer_orders_cleaned
LIMIT 10;

```

114 • select *
115 from customer_orders_cleaned;

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	customer_id	first_name	last_name	full_name	email	mobile_number	order_id	order_year	order_date	delivery_date	delivery_days	order_amount	city
▶	1001	ANITA	SHARMA	ANITA SHARMA	anita1@gmail.com	9110053353	ORD-2022-0001	2022	2022-07-18	2022-07-20	2	3662.38	PUNE
	1002	KIRAN	GUPTA	KIRAN GUPTA	kiran2@gmail.com	9749621470	ORD-2021-0002	2021	2021-01-14	2021-01-21	7	10669.92	DELHI
	1003	VIKAS	PATEL	VIKAS PATEL	vikas3@yahoo.com	9664130526	ORD-2024-0003	2024	2024-02-05	2024-02-08	3	23175.88	BANGALOR
	1004	POOJA	SINGH	POOJA SINGH	pooja4@yahoo.com	9654049436	ORD-2022-0004	2022	2022-07-01	2022-07-07	6	38255.82	MUMBAI
	1005	RONITA	VERMA	RONITA VERMA	ronita5@yahoo.com	9840002571	ORD-2022-0005	2022	2022-07-03	2022-07-04	1	91407.49	HYDERABAD

customer_orders 8 customer_orders_cleaned 24 ×