

Project : Data cleaning

Name : AMARJEET KUMAR

Email: amarjeetbilla63@gmail.com

Batch: DA - BATCH ^6

Phone: 9871434547

◆ STEP 0: Inspect Raw Data

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

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying a SQL query: `SELECT * FROM customer_orders LIMIT 9;`. The 'Results' tab shows the output of the query, which is a table with 10 columns: `customer_id`, `first_name`, `last_name`, `email`, `mobile_number`, `order_id`, `order_date`, `delivery_date`, `order_amount`, `city`, `signup_date`, and `rating`. The table contains 10 rows of data. The 'Output' tab at the bottom shows the execution log, including the query execution time and the number of rows returned.

| 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 | kirant2@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 | +91-9811514914 | ORD-2023-0006 | 2023-11-10 | 2023-11-14 | 59842.699 | CHENNAI | 2023-07-30 | 3.883 |
| 1007 | Arjun | GUPTA | arjun7@yahoo.com | +91-9466825638 | ORD-2022-0007 | 2022-11-28 | 2022-11-29 | 50308.318 | MUMBAI | 2022-01-22 | 2.496 |
| 1008 | ROHIT | Sharma | rohit8@GMAIL.COM | 0091-9997612044 | ORD-2022-0008 | 2022-10-26 | 2022-10-30 | 8336.526 | hyderabad | 2022-09-27 | 1.661 |

◆ 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;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'data_management' selected. The main editor shows a SQL query in 'Query 1' window:

```
9
10 -- q2 :::----->>
11 • select
12     first_name,
13     trim(first_name) as trim_fname,
14     upper(trim(first_name)) as clean_fname
15 from customer_orders
16 limit 15;
```

Below the query editor, the 'Result Grid' shows the results of the query:

| first_name | trim_fname | clean_fname |
|------------|------------|-------------|
| Anita | Anita | ANITA |
| KIRAN | KIRAN | KIRAN |
| Vikas | Vikas | VIKAS |
| POOJA | POOJA | POOJA |
| POOJA | POOJA | POOJA |
| NEHA | NEHA | NEHA |

The bottom panel shows the 'Action Output' log with the following entries:

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|---------------------|-----------------------|
| 7 | 13:41:55 | DEALLOCATE PREPARE stmt | OK | 0.000 sec |
| 8 | 13:42:38 | select * from customer_orders LIMIT 0, 50000 | 350 row(s) returned | 0.000 sec / 0.000 sec |
| 9 | 13:42:51 | select * from customer_orders limit 5 | 5 row(s) returned | 0.000 sec / 0.000 sec |
| 10 | 13:44:08 | select * from customer_orders limit 9 | 9 row(s) returned | 0.000 sec / 0.000 sec |
| 11 | 13:49:37 | select first_name, trim(first_name) as trim_fname... | 350 row(s) returned | 0.015 sec / 0.000 sec |
| 12 | 13:49:58 | select * from customer_orders limit 9 | 9 row(s) returned | 0.000 sec / 0.000 sec |
| 13 | 13:49:58 | select first_name, trim(first_name) as trim_fname... | 6 row(s) returned | 0.000 sec / 0.000 sec |

◆ STEP 2: Clean last_name

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

MySQL Workbench

data cleaning x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees_salary.customer_ord...

Limit to 50000 rows

```

17
18 -- step 2 --> Clean last_name
19 • SELECT
20     last_name,
21     UPPER(TRIM(last_name)) AS cleaned_last_name
22 FROM customer_orders
23 limit 10;
24

```

Result Grid

| last_name | cleaned_last_name |
|-----------|-------------------|
| SHARMA | SHARMA |
| Gupta | GUPTA |
| PATEL | PATEL |
| SINGH | SINGH |
| VERMA | VERMA |
| Kumar | KUMAR |
| GUPTA | GUPTA |
| Sharma | SHARMA |

Result 7 x

Read Only

Schema: data_management

Output

| # | Time | Action | Message | Duration / Fetch |
|------|----------|---|---------------------|-----------------------|
| ✓ 8 | 13:42:38 | select * from customer_orders LIMIT 0, 50000 | 350 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ 9 | 13:42:51 | select * from customer_orders limit 5 | 5 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ 10 | 13:44:08 | select * from customer_orders limit 9 | 9 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ 11 | 13:49:37 | select first_name, trim(first_name) as trim_fnam... | 350 row(s) returned | 0.015 sec / 0.000 sec |
| ✓ 12 | 13:49:58 | select * from customer_orders limit 9 | 9 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ 13 | 13:49:58 | select first_name, trim(first_name) as trim_fnam... | 6 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ 14 | 13:54:00 | SELECT last_name, UPPER(TRIM(last_name)) A... | 10 row(s) returned | 0.000 sec / 0.000 sec |

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

```

SELECT
    CONCAT(
        UPPER(TRIM(first_name)),
        ' ',
        UPPER(TRIM(last_name))
    ) AS full_name

```

```
FROM customer_orders;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'data_management' selected. The main editor shows a SQL query for 'Query 1' in the 'employees_salary.customer_ord...' database. The query is as follows:

```
25 -- STEP 3: Create full_name (CONCAT)
26 • SELECT
27     CONCAT( UPPER(TRIM(first_name)), ' ', UPPER(TRIM(last_name))
28     ) AS full_name
29   from customer_orders ;
30
31
32
```

The 'Result Grid' shows the output of the query, displaying a list of full names:

| full_name |
|--------------|
| ANITA SHARMA |
| KIRAN GUPTA |
| VIKAS PATEL |
| POOJA SINGH |
| POOJA VERMA |
| NEHA KUMAR |
| ARJUN GUPTA |
| ROHIT SHARMA |

The 'Output' pane at the bottom shows the 'Action Output' table, which includes columns for '#', 'Time', 'Action', 'Message', and 'Duration / Fetch'. It lists the execution of various SQL statements and their results.

◆ STEP 4: Clean email (Standardization)

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

MySQL Workbench

data cleaning x

File Edit View Query Database Server Tools Scripting Help

Navigator: employees_salary.customer_ord...

Query 1 x

Limit to 50000 rows

```

28 ) AS full_name
29 from customer_orders ;
30
31 -- o STEP 4: Clean email (Standardization)
32 • select email,
33     lower(email) as clear_email
34 from customer_orders ;
35

```

Result Grid

| email | clear_email |
|------------------|------------------|
| anita1@GMAIL.COM | anita1@gmail.com |
| kiran2@GMAIL.COM | kiran2@gmail.com |
| vikas3@yahoo.com | vikas3@yahoo.com |
| pooja4@yahoo.com | pooja4@yahoo.com |
| pooja5@yahoo.com | pooja5@yahoo.com |
| neha6@GMAIL.COM | neha6@gmail.com |
| arjun7@yahoo.com | arjun7@yahoo.com |
| rohit8@GMAIL.COM | rohit8@gmail.com |

Result 9 x

Read Only

Output

Action Output

| # | Time | Action | Message | Duration / Fetch |
|------|----------|--|---------------------|-----------------------|
| ✓ 10 | 13:44:08 | select * from customer_orders limit 9 | 9 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ 11 | 13:49:37 | select first_name, trim(first_name) as trim_fnam... | 350 row(s) returned | 0.015 sec / 0.000 sec |
| ✓ 12 | 13:49:58 | select * from customer_orders limit 9 | 9 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ 13 | 13:49:58 | select first_name, trim(first_name) as trim_fnam... | 6 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ 14 | 13:54:00 | SELECT last_name, UPPER(TRIM(last_name)) A... | 10 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ 15 | 13:57:36 | SELECT CONCAT(UPPER(TRIM(first_name)),"... | 350 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ 16 | 14:00:08 | select email, lower(email) as clear_email from cust... | 350 row(s) returned | 0.000 sec / 0.000 sec |

Schema: data_management

Object Info Session

◆ 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;

```

```

35
36 -- ◊ STEP 5: Clean mobile_number (Extract last 10 digits)
37 • select
38     mobile_number,
39     substr(mobile_number,length(mobile_number) -9,10) as cleaned_mobile_number
40 from customer_orders;
41
42

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [↗](#)

| mobile_number | cleaned_mobile_number |
|-----------------|-----------------------|
| 919110053353 | 9110053353 |
| 0091-9749621470 | 9749621470 |
| 0091-9664130526 | 9664130526 |
| 919654049436 | 9654049436 |
| 919940992571 | 9940992571 |
| +91-9811514914 | 9811514914 |
| +91-9466825638 | 9466825638 |
| 0091-9997612044 | 9997612044 |

Result 10 x

◆ STEP 6: Extract Year from **order_id**

```

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

```

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 50000 rows' dropdown. The SQL editor contains the following code:

```
41
42  -- ♦ STEP 6: Extract Year from order_id
43  • SELECT order_id,
44         SUBSTR(order_id, 5, 4) AS ordered_year
45  FROM customer_orders;
46
47
48
```

Below the editor is the 'Result Grid' section, which includes a 'Filter Rows' input and 'Export' and 'Wrap Cell Content' options. The results are displayed in a table:

| | order_id | ordered_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 |
| | ORD-2022-0007 | 2022 |
| | ORD-2022-0008 | 2022 |

At the bottom, it says 'Result 11' with a close button.

♦ STEP 7: Round **order_amount**

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

```

46
47 -- ◊ STEP 7: Round order_amount
48 • SELECT
49     order_amount,
50     ROUND(order_amount, 2) AS cleaned_order_amount
51 FROM customer_orders;
52
53

```

| Result Grid | | | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|--------------|----------------------|--------------|---------|--------------------|
| | 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 | | | |
| | 59842.699 | 59842.7 | | | |
| | 50308.318 | 50308.32 | | | |
| | 8336.326 | 8336.33 | | | |

◆ STEP 8: Round **rating**

```

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

```



```
52
53  -- ◊ STEP 8: Round rating
54 • SELECT
55     rating,
56     ROUND(rating, 1) AS cleaned_rating
57 FROM customer_orders;
58
59
```

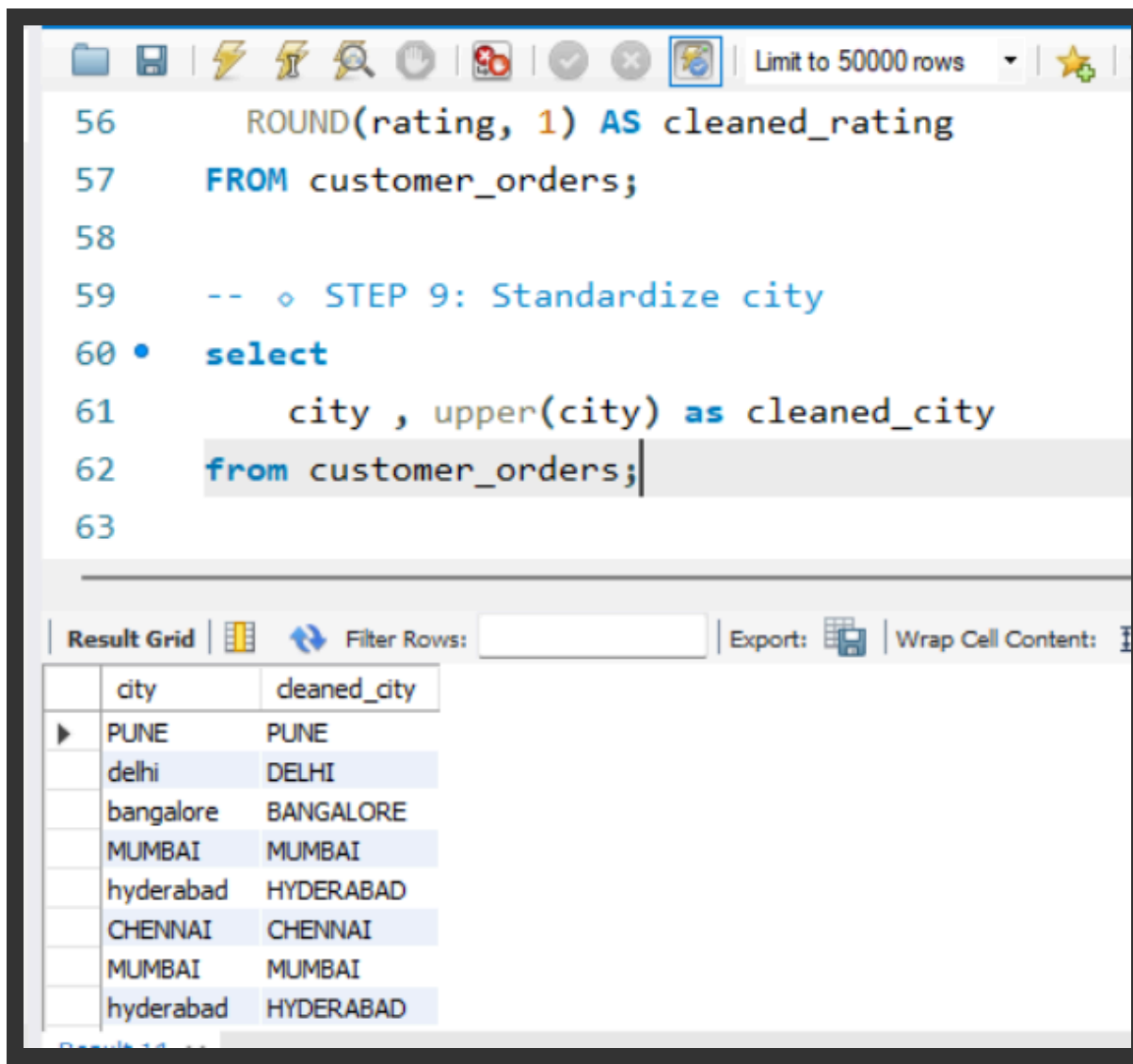
Result Grid | Filter Rows: | Export: | Wrap Cell C

| | 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 |
| | 2.496 | 2.5 |
| | 1.661 | 1.7 |

Result 13 x

◆ STEP 9: Standardize **city**

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



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 50000 rows' dropdown. The SQL editor contains the following code:

```
56     ROUND(rating, 1) AS cleaned_rating
57 FROM customer_orders;
58
59 -- ♦ STEP 9: Standardize city
60 • select
61     city , upper(city) as cleaned_city
62 from customer_orders;
63
```

Below the editor is the 'Result Grid' section, which includes a 'Filter Rows' input and an 'Export' button. The result grid displays the following data:

| city | cleaned_city |
|-----------|--------------|
| PUNE | PUNE |
| delhi | DELHI |
| bangalore | BANGALORE |
| MUMBAI | MUMBAI |
| hyderabad | HYDERABAD |
| CHENNAI | CHENNAI |
| MUMBAI | MUMBAI |
| hyderabad | HYDERABAD |




♦ STEP 10: Delivery Time Calculation (**DATEDIFF**)

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

```

63
64  -- ♦ STEP 10: Delivery Time Calculation (DATEDIFF)
65 • select
66     delivery_date,
67     order_date,
68     datediff(delivery_date,order_date) as delivery_days_remains
69 from customer_orders;
70

```

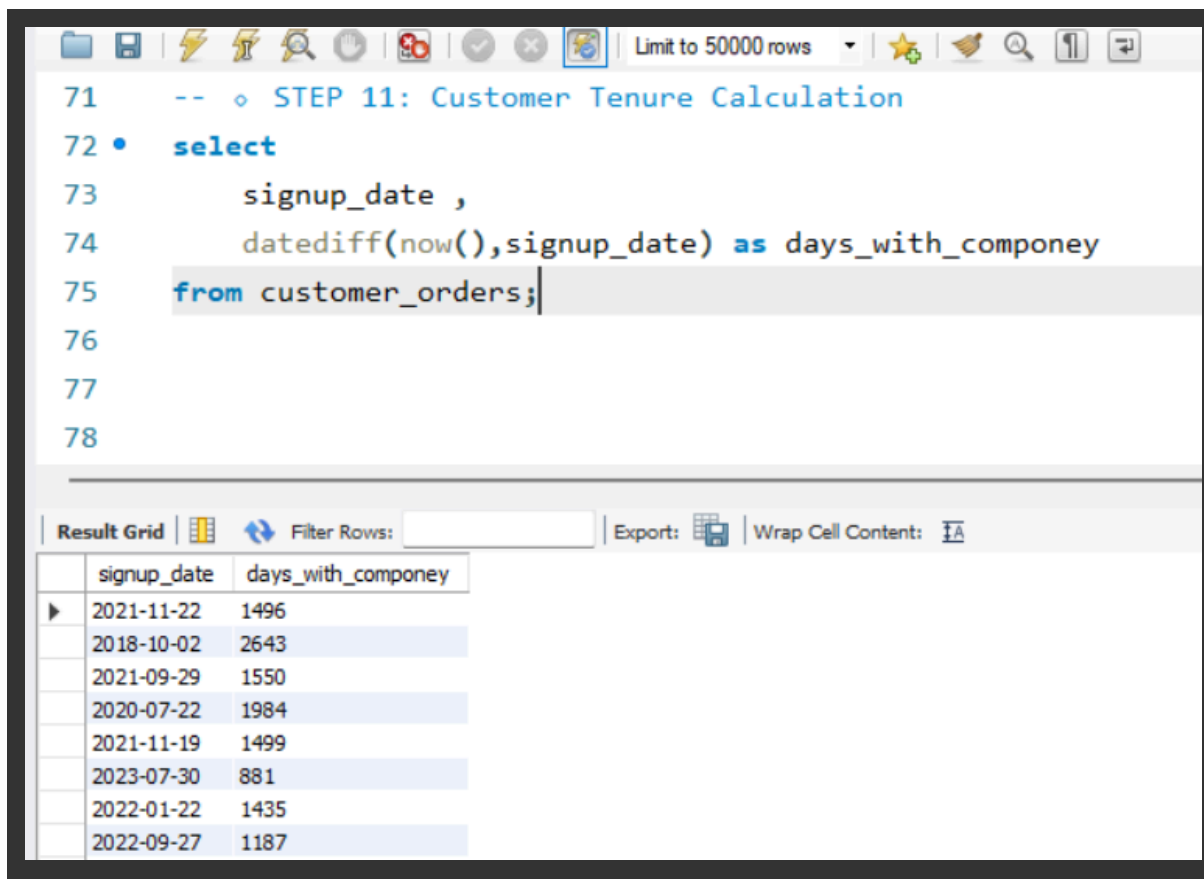
| Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content:  | | | |
|--|---------------|------------|-----------------------|
| | delivery_date | order_date | delivery_days_remains |
| ▶ | 2022-07-20 | 2022-07-18 | 2 |
| | 2021-01-21 | 2021-01-14 | 7 |
| | 2024-02-08 | 2024-02-05 | 3 |
| | 2022-07-07 | 2022-07-01 | 6 |
| | 2022-07-04 | 2022-07-03 | 1 |
| | 2023-11-14 | 2023-11-10 | 4 |
| | 2022-11-29 | 2022-11-28 | 1 |
| | 2022-10-30 | 2022-10-26 | 4 |

♦ STEP 11: Customer Tenure Calculation

```

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

```



```
71 -- ◊ STEP 11: Customer Tenure Calculation
72 • select
73     signup_date ,
74     datediff(now(),signup_date) as days_with_componney
75 from customer_orders;
76
77
78
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

| | signup_date | days_with_componney |
|---|-------------|---------------------|
| ▶ | 2021-11-22 | 1496 |
| | 2018-10-02 | 2643 |
| | 2021-09-29 | 1550 |
| | 2020-07-22 | 1984 |
| | 2021-11-19 | 1499 |
| | 2023-07-30 | 881 |
| | 2022-01-22 | 1435 |
| | 2022-09-27 | 1187 |

◆ 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;
```

```

77 -- STEP 12: CASE WHEN - Order Value Category
78 • select
79     CONCAT( UPPER(TRIM(first_name)), ' ', UPPER(TRIM(last_name))) AS full_name , order_amount,
80     case
81         when order_amount >=50000 then 'ambani mere ____ pe '
82         when order_amount >=30000 then 'bss itni aukat hai teri ,so you arre ambani ,hehehe'
83         when order_amount >=20000 then 'teri aukat nhi hai order_dene ki toh apni ____ mra'
84         else 'bhai tu ekk km kr , tel__chatai ka dhndha start kr de '
85     end as aukat_ka_mirror
86 from customer_orders;

```

| full_name | order_amount | aukat_ka_mirror |
|--------------|--------------|--|
| ANITA SHARMA | 3662.377 | bhai tu ekk km kr , tel__chatai ka dhndha start k... |
| KIRAN GUPTA | 10669.923 | bhai tu ekk km kr , tel__chatai ka dhndha start k... |
| VIKAS PATEL | 23175.878 | teri aukat nhi hai order_dene ki toh apni ____ mra |
| POOJA SINGH | 38255.816 | bss itni aukat hai teri ,so you arre ambani ,heh... |
| POOJA VERMA | 91497.485 | ambani mere ____ pe |

◆ 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;

```

```

87
88 -- ♦ STEP 13: CASE WHEN - Customer Type
89 • SELECT signup_date,
90     CASE
91         WHEN DATEDIFF(NOW(), signup_date) <= 30 THEN 'New'
92         WHEN DATEDIFF(NOW(), signup_date) <= 180 THEN 'Regular'
93         ELSE 'Loyal'
94     END AS customer_type
95 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 | | |
| 2022-01-22 | Loyal | | |

Result 20 x

♦ 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;

```

```

103  -- STEP 14: FINAL CLEANED VIEW (Industry Practice)
104 • create view vw_for_cleaned_customer_orders as
105 select
106     customer_id,
107     upper(trim(first_name)) as clean_fname,
108     UPPER(TRIM(last_name)) AS cleaned_last_name,
109     CONCAT( UPPER(TRIM(first_name)), ' ',UPPER(TRIM(last_name))) AS full_name ,
110     lower(email) as clear_email ,
111     substr(mobile_number,length(mobile_number) -9,10) as cleaned_mobile_number ,
112     order_id,
113     SUBSTR(order_id, 5, 4) AS ordered_year,
114     ROUND(order_amount, 2) AS cleaned_order_amount,
115     delivery_date,
116     upper(city) as cleaned_city ,
117     order_date,
118     datediff(delivery_date,order_date) as delivery_days_remains,
119     signup_date ,
120     datediff(now(),signup_date) as days_with_componey,
121     case
122         when order_amount >=50000 then 'ambani mere ____ pe '
123         when order_amount >=30000 then 'bss itni aukat hai teri ,so you  arre ambani ,hehehe'
124         when order_amount >=20000 then 'teri aukat nhi hai order_dene ki toh apni ____ mra'
125         else 'bhai tu ekk km kr , tel__chatai ka dhndha start kr de '
126     end as order_category
127 from customer_orders;

```

◆ STEP 15: Validate Cleaned Data

```
SELECT *  
FROM customer_orders_cleaned  
LIMIT 10;
```

```
-- STEP 15: Validate Cleaned Data
```

```
• select *  
  from vw_for_cleaned_customer_orders  
 limit 20;
```

| cleaned_mobile_number | order_id | ordered_year | cleaned_order_amount | delivery_date | cleaned_city | order_date | delivery_days_remains | signup_date | days_with_componey | order_category |
|-----------------------|---------------|--------------|----------------------|---------------|--------------|------------|-----------------------|-------------|--------------------|--|
| 9110053353 | ORD-2022-0001 | 2022 | 3662.38 | 2022-07-20 | PUNE | 2022-07-18 | 2 | 2021-11-22 | 1496 | bhai tu ekk km kr , tel_chatai ka dhndha stz |
| 9749621470 | ORD-2021-0002 | 2021 | 10669.92 | 2021-01-21 | DELHI | 2021-01-14 | 7 | 2018-10-02 | 2643 | bhai tu ekk km kr , tel_chatai ka dhndha stz |
| 9664130526 | ORD-2024-0003 | 2024 | 23175.88 | 2024-02-08 | BANGALORE | 2024-02-05 | 3 | 2021-09-29 | 1550 | teri aukat nhi hai order_dene ki toh apni |
| 9654049436 | ORD-2022-0004 | 2022 | 38255.82 | 2022-07-07 | MUMBAI | 2022-07-01 | 6 | 2020-07-22 | 1984 | bss itni aukat hai teri ,so you arre ambani ,) |
| 9940992571 | ORD-2022-0005 | 2022 | 91497.48 | 2022-07-04 | HYDERABAD | 2022-07-03 | 1 | 2021-11-19 | 1499 | ambani mere ____pe |
| 9811514914 | ORD-2023-0006 | 2023 | 59842.7 | 2023-11-14 | CHENNAI | 2023-11-10 | 4 | 2023-07-30 | 881 | ambani mere ____pe |
| 9466825638 | ORD-2022-0007 | 2022 | 50308.32 | 2022-11-29 | MUMBAI | 2022-11-28 | 1 | 2022-01-22 | 1435 | ambani mere ____pe |
| 9997612044 | ORD-2022-0008 | 2022 | 8336.33 | 2022-10-30 | HYDERABAD | 2022-10-26 | 4 | 2022-09-27 | 1187 | bhai tu ekk km kr , tel_chatai ka dhndha stz |
| 9215984311 | ORD-2022-0009 | 2022 | 44090.96 | 2022-03-15 | MUMBAI | 2022-03-10 | 5 | 2020-01-17 | 2171 | bss itni aukat hai teri ,so you arre ambani ,) |
| 9354794895 | ORD-2021-0010 | 2021 | 72473.58 | 2021-05-20 | CHENNAI | 2021-05-19 | 1 | 2018-12-11 | 2573 | ambani mere ____pe |
| 9701642483 | ORD-2023-0011 | 2023 | 86236.42 | 2023-08-22 | DELHI | 2023-08-15 | 7 | 2021-07-20 | 1621 | ambani mere ____pe |
| 9897677788 | ORD-2021-0012 | 2021 | 40403.51 | 2021-04-20 | PUNE | 2021-04-14 | 6 | 2019-09-29 | 2281 | bss itni aukat hai teri ,so you arre ambani ,) |