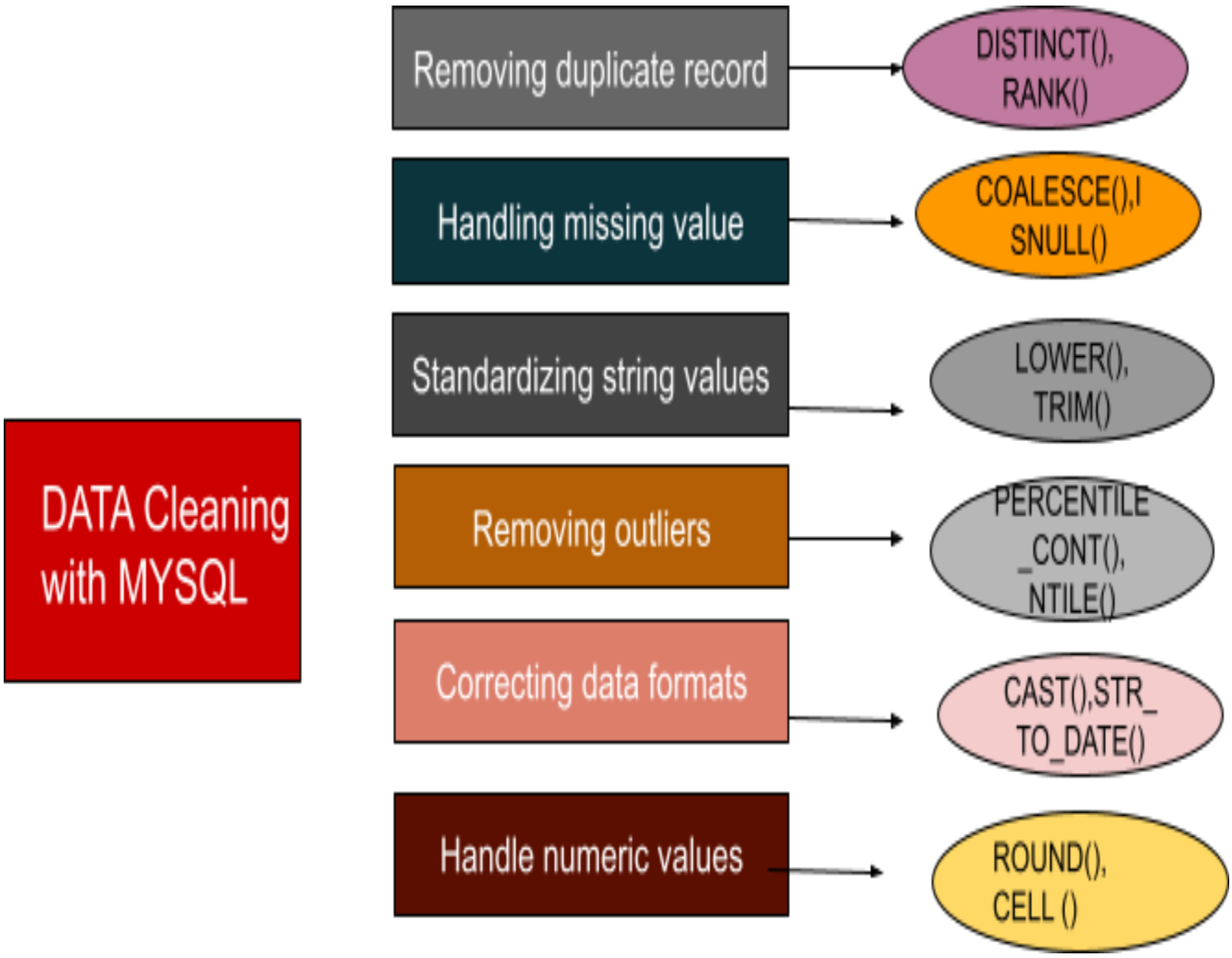
DATA CLEANING

Transform Raw DATA into Actionable Insights Using Mysql

Created by;Olanrewaju Dasola

WHAT IS DATA CLEANING?

Data cleaning is the process of identifying and correcting errors,inconsistencies,or inaccuracies in data to ensure its quality and reliability for analysis.It includes tasks like;



This process ensures that the data is clean,consistent,and ready for analysis.

01. Removing Duplicate Records

Duplicates can distort analysis, so removing them is critical.

Table Name: customer_data

Customer_id	Customer_name	Email
1	John Smith	john@example.com
2	Jane Doe	jane@example.com
3	John Smith	john@example.com

Query:

```
DELETE FROM customer_data
WHERE customer_id NOT IN (
   SELECT MIN(customer_id)
   FROM customer_data
 GROUP BY email
);
```

Explanation:

• This query keeps the first record of each unique email (MIN(customer_id)) and deletes duplicates.

Output:

Customer_id	Customer_name	Email
1	John Smith	john@example.com
2	Jane Doe	jane@example.com

O2. Handling Missing Values

Identify and handle records with NULL values.

Table Name: sales_data

Sale_id	Sale_date	Sale_amount
1	2025-01-01	100
2	NULL	200
3	2025-01-03	NULL

Query 1: Find Missing Values

```
SELECT * FROM sales_data
WHERE sale_date IS NULL OR sale_amount IS NULL;
```

Query 2: Replace Missing Values

```
UPDATE sales_data
SET sale_date = '2025-01-01'
WHERE sale_date IS NULL;

UPDATE sales_data
SET sale_amount = 0
WHERE sale_amount IS NULL;
```

Output After Updates:

Sale_id	Sale_date	Sale_amount
1	2025-01-01	100
2	2025-01-01	200
3	2025-01-03	0

03. Standardizing String Values

Standardize inconsistent string formats for better analysis.

Table Name: product_data

Product_id	Product_name
1	laptop
2	LAPTOP
3	Laptop

Query:

```
UPDATE product_data
SET product_name = LOWER(product_name);
```

Explanation:

- The LOWER function converts all product names to lowercase.

Output:

Product_id	Product_name
1	laptop
2	laptop
3	laptop

04. Removing Outliers

Outliers can skew results and need to be handled.

Table Name: employee_data

Employee_id	Salary
1	50000
2	60000
3	1000000

Query:

```
DELETE FROM employee_data
WHERE salary > (SELECT AVG(salary) + 3 * STDDEV(salary) FROM employee_data);
```

Explanation:

• The query removes records with salaries exceeding 3 standard deviations above the mean.

Output:

Employee_id	Salary
1	50000
2	60000

05. Correcting Data Formats

Fix inconsistent date formats or invalid entries.

Table Name: order_data

Order_id	Order_date
1	2025/01/21
2	21-01-2025

Query:

```
UPDATE order_data
SET order_date = STR_TO_DATE(order_date, '%Y/%m/%d')
WHERE order_date LIKE '%/%';
```

Explanation:

• The STR_TO_DATE function converts date strings into a consistent format (YYYY-MM-DD).

Output:

Order_id	Order_date
1	2025-01-21
2	2025-01-21

06. HANDLE NUMERIC VALUES

You can handle numeric values using the functions ROUND, CEIL, FLOOR, and ABS in SQL. Consider the following table:

Table Name: sales_data

Sale_id	Sale_amount
1	234.567
2	3
3	-78.423
4	456.789
5	123.001
6	-65.999

I. ROUND Function

The ROUND function is used to round a number to a specified number of decimal places.

Query:

```
SELECT
  sale_id,
  sale_amount,
  ROUND(sale_amount, 2) AS rounded_2_decimals,
  ROUND(sale_amount, 0) AS rounded_to_nearest_integer
FROM sales_data;
```

Output:

Sale_id	Sale_amount	Rounded_2_decimals	Rounded_to_nearest_integer
1	234.567	234.57	235
2	3	3.00	3
3	-78.423	-78.42	-78

II. CEIL Function

The CEIL (Ceiling) function rounds a number up to the nearest integer, regardless of its decimal part.

Query:

```
SELECT
  sale_id,
  sale_amount,
  CEIL(sale_amount) AS ceiling_value
FROM sales_data;
```

Output:

Sale_id	Sale_amount	Floor_value
1	234.567	234
2	3	3
3	-78.423	-79

III. FLOOR Function

The FLOOR function rounds a number down to the nearest integer.

Query:

```
SELECT
  sale_id,
  sale_amount,
  FLOOR(sale_amount) AS floor_value
FROM sales_data;
```

Output:

Sale_id	Sale_amount	Floor_value
1	234.567	234
2	3	3
3	-78.423	-79

IV. ABS Function

The ABS (Absolute) function returns the positive value of a number, removing any negative sign.

Query:

```
SELECT
  sale_id,
  sale_amount,
  ABS(sale_amount) AS absolute_value
FROM sales_data;
```

Output:

Sale_id	Sale_amount	Absolute_value
1	234.567	234.567
2	3	3
3	-78.423	78.423