CUSTOMER FEEDBACK ANALYSIS

SQL Data Cleaning Project

Presentation by Pankaj Yadav

OBJECTIVE:

The task is to clean and prepare this data for analysis to identify common issues and areas for improvement.

DATA DICTIONARY

- **review_id:** Unique identifier for each review
- **customer_id:** Unique identifier for each customer
- **feedback_source:** Source of the feedback (e.g., "online_review", "survey", "support_call")
- **feedback_text:** The text of the customer feedback
- **rating:** Customer rating (1 to 5 stars)
- **timestamp:** The date and time when the feedback was submitted

TASKS

1. Remove duplicate rows based on the review_id column.
2. Fill missing feedback_text values with "No feedback provided".
3. Fill missing rating values with the median rating.
4. Remove HTML tags from the feedback_text column.
5. Convert timestamp values to a consistent datetime format.
6. Remove rows with invalid dates.

1. REMOVE DUPLICATE ROWS BASED ON THE REVIEW_ID COLUMN.

```
WITH delete_duplicates AS(
SELECT *, ROW_NUMBER() OVER(PARTITION BY review_id
ORDER BY review_id) AS rwn
FROM customer_feedback
)
DELETE FROM customer_feedback
WHERE review_id IN (
   SELECT review_id FROM delete_duplicates WHERE rwn > 1
);
```

2. FILL MISSING FEEDBACK_TEXT VALUES WITH "NO FEEDBACK PROVIDED".

```sql
UPDATE customer_feedback
SET
    feedback_text = 'No feedback provided'
WHERE
    feedback_text = '';
```

3. FILL MISSING RATING VALUES WITH THE MEDIAN RATING.

```
SET @avg_rating = (SELECT AVG(rating) FROM
customer_feedback WHERE rating IS NOT NULL);

UPDATE customer_feedback
SET
    rating = @avg_rating
WHERE
    rating IS NULL;
```

4. REMOVE HTML TAGS FROM THE FEEDBACK_TEXT COLUMN.

```
UPDATE customer_feedback
SET
    feedback_text = REPLACE(feedback_text,
        '<b>Bad service</b>',
        'Bad service')
WHERE
    feedback_text = '<b>Bad service</b>';
```

```
UPDATE customer_feedback
SET
    feedback_text = REPLACE(feedback_text,
        'Terrible! <a href=\'http://example.com\'>Read more</a>',
        'Terrible!')
WHERE
    feedback_text = 'Terrible! <a
href=\'http://example.com\'>Read more</a>';
```

5. CONVERT TIMESTAMP VALUES TO A CONSISTENT DATETIME FORMAT.

```sql
UPDATE customer_feedback
SET
   timestamp = STR_TO_DATE(timestamp, '%m/%d/%Y %H:%i')
WHERE
   timestamp LIKE '%/%/% %:%';
```

Updating timestamp values to date format to convert the data type of the column

```
SELECT
    timestamps
FROM
    customer_feedback
WHERE
    timestamp LIKE '%-%-% %:%:%'
        AND (timestamp REGEXP '^[0-9]{4}-(02)-(30|31)'
        OR timestamp REGEXP '^[0-9]{4}-(04|06|09|11)-(31)');
```

Checking the values of the timestamp column is valid or not because the values in the timestamp column is of date and time.

```sql
UPDATE customer_feedback
SET
    timestamp = REPLACE(timestamp,
        '2024-02-30',
        '2024-02-28')
WHERE
    timestamp LIKE '2024-02-30%';
```

Updating the value found which is not valid for the date because it is wrongly type as ‘2024-02-30’ - In the month of February 30th is never come.

```
UPDATE customer_feedback
SET
    timestamp = STR_TO_DATE(timestamp, '%Y-%m-%d
%H:%i:%s')
WHERE
    timestamp LIKE '%-%-% %:%:%';
```

Updating timestamp values to date format to convert the data type of the column

```
ALTER TABLE customer_feedback
MODIFY COLUMN timestamp DATETIME;
```

Altering table or changing the data type of the column timestamp to DATETIME

6. REMOVE ROWS WITH INVALID DATES.

```sql
DELETE FROM customer_feedback
WHERE
    timestamps = 'invalid_date';
```

THANK YOU FOR EXPLORING!

If you have any questions, feel free to reach out through any of the following:

EMAIL ADDRESS
pankyadav2002@gmail.com

SOCIAL MEDIA
@https://www.linkedin.com/in/thepankajyadav/