

SQL Commands for Data Cleaning in Transactions Table

1 Extract Unique Names from a Column

Title: Identify unique values in a column to understand existing data entries.

```
SELECT DISTINCT customer_id
FROM transactions;
```

Explanation: - `SELECT DISTINCT customer_id` → retrieves all unique customer IDs. - `FROM transactions` → from the `transactions` table. - This helps to see all unique identifiers without duplicates.

2 Sort Data in Ascending Order

Title: Sort numerical column data to understand distribution.

```
SELECT *
FROM transactions
ORDER BY customer_id ASC;
```

Explanation: - `ORDER BY customer_id ASC` → sorts all transaction rows by `customer_id` in ascending order. - `*` selects all columns.

3 View Rows with NULL or Empty Values

Title: Identify missing data in specific columns.

```
SELECT *
FROM transactions
WHERE Branch IS NULL OR Branch = '';
```

Explanation: - `Branch IS NULL` → selects rows where the `Branch` column is NULL. - `Branch = ''` → selects rows where `Branch` is an empty string. - `OR` ensures both conditions are included.

4 Replace NULL Values with 'Unknown'

Title: Fill missing values with a placeholder.

```
UPDATE transactions
SET Branch = 'Unknown'
WHERE Branch IS NULL OR Branch = '';
```

Explanation: - `UPDATE transactions` → target table. - `SET Branch = 'Unknown'` → assigns 'Unknown' to empty or NULL Branch. - `WHERE Branch IS NULL OR Branch = ''` → ensures only missing entries are updated.

5 Identify Foreign/Invalid Branch Names

Title: Find Branch values that do not match official Nepal district names.

```
SELECT DISTINCT Branch
FROM transactions
WHERE Branch IS NOT NULL
  AND TRIM(Branch) <> ''
  AND Branch NOT IN (
    SELECT district_name
    FROM nepal_districts
    WHERE district_name IS NOT NULL
  );
```

Explanation: - `DISTINCT Branch` → unique branch names. - `TRIM(Branch)` → removes spaces. - `Branch NOT IN (SELECT district_name ...)` → filters out valid districts. - Result: foreign countries or invalid entries.

6 Identify Used Nepali Districts in Branch Column

Title: Determine which districts are already used in transactions.

```
SELECT DISTINCT Branch
FROM transactions
WHERE Branch IS NOT NULL
  AND Branch <> ''
  AND Branch IN (SELECT district_name FROM nepal_districts);
```

Explanation: - `DISTINCT Branch` → only unique district names. - `Branch IN (SELECT district_name ...)` → ensures only valid districts are included.

7 Insert Missing Districts into Nepal Districts Table

Title: Add missing district names to the reference table.

```
INSERT INTO nepal_districts (district_name)
VALUES ('Khotang');
```

Explanation: - Adds a new row 'Khotang' into `nepal_districts`. - `VALUES` specifies the value to insert.

8 Delete Values from Nepal Districts Table

Title: Remove incorrect or unwanted entries.

```
DELETE FROM nepal_districts
WHERE district_name = 'Rukum';
```

Explanation: - Deletes the row where `district_name` is 'Rukum'. - Only affects rows matching the WHERE condition.

9 Find Specific Values in a Table

Title: Look up specific data entries.

```
SELECT *
FROM nepal_districts
WHERE district_name IN ('Eastern Rukum', 'Western Rukum');
```

Explanation: - `IN` checks multiple values. - Returns rows matching either 'Eastern Rukum' or 'Western Rukum'.

10 Replace Foreign Countries with Unused Districts

Title: Map remaining foreign countries in Branch column to unused Nepal districts.

```
WITH foreign_branches AS (
  SELECT DISTINCT Branch
  FROM transactions
  WHERE Branch IS NOT NULL
  AND Branch <> ''
  AND Branch IN (
    'Martinica', 'Nepal', 'Lituania', 'Tayikistan', 'Sierra Leona',
    'Jordania', 'Suiza', 'Turkmenistan', 'Papua Nueva Darchula',
    'Yibuti',
    'Oman', 'Sri Lanka', 'Laos', 'Macedonia', 'Zimbabue', 'Uganda',
    'Tunez', 'Republica Checa', 'Qatar', 'Moldavia', 'Unknown',
    'Siria',
```

```

        'Lesoto', 'Mali', 'Libia', 'Republica del Congo', 'Paraguay'
    )
),
foreign_rn AS (
    SELECT Branch, ROW_NUMBER() OVER (ORDER BY Branch) AS rn
    FROM foreign_branches
),
districts_rn AS (
    SELECT district_name, ROW_NUMBER() OVER () AS rn
    FROM (VALUES
        ('Morang'), ('Sankhuwasabha'), ('Taplejung'), ('Terhathum'),
        ('Udayapur'),
        ('Dhanusha'), ('Sarlahi'), ('Siraha'), ('Dolakha'), ('Nuwakot'),
        ('Rasuwa'), ('Sindhupalchok'), ('Parasi'), ('Rupandehi'), ('Dolpa'),
        ('Jumla'), ('Mugu'), ('Salyan'), ('Bajhang'), ('Kailali'),
        ('Kanchanpur')
    ) AS t(district_name)
),
mapping AS (
    SELECT f.Branch, d.district_name
    FROM foreign_rn f
    JOIN districts_rn d
        ON ((f.rn - 1) % (SELECT COUNT(*) FROM districts_rn)) + 1 = d.rn
)
UPDATE transactions t
SET Branch = m.district_name
FROM mapping m
WHERE TRIM(t.Branch) = TRIM(m.Branch);

```

Explanation:

1. `foreign_branches` → lists all foreign countries currently in Branch.
2. `foreign_rn` → assigns a unique row number to each foreign branch.
3. `districts_rn` → assigns row numbers to unused districts.
4. `mapping` → joins foreign branches with unused districts, cycling districts if needed.
5. `UPDATE` → replaces each foreign country with the assigned district.
6. `TRIM()` ensures that extra spaces don't prevent matching.

This Word document would contain all these commands **with comments and explanations**, helping you follow the data cleaning process clearly.

Do you want me to **generate the actual** `.docx` **file** so you can download it directly?