SalesAnalytics Data Cleaning Project

This project is designed to provide a **reusable environment** for practicing data cleaning and standardization on the `SalesAnalytics` database. It is based on the **Clean and Standardize Messy Input Data Tutorial** and will serve as the foundation for upcoming tutorials.

---

1. Project Structure

```
SalesAnalytics_DataCleaning_Project/
│
├── SQL_Scripts/
│   ├── 01_Cleanup_Customers.sql
│   ├── 02_Cleanup_Products.sql
│   ├── 03_Cleanup_Orders.sql
│   ├── 04_Cleanup_Reviews.sql
│   └── 05_Cleanup_InventoryTransactions.sql
│
├── Documentation/
│   └── Clean_Standardize_Data_Tutorial.md
│
├── Sample_Data/
│   └── Optional CSVs or inserts for testing
│
└── Project_Readme.md
```

---

2. Project Goals

1. **Practice cleaning and standardizing messy data** in SQL.
2. Maintain a **reusable set of scripts** for consistent data cleaning.
3. Ensure all **FK-safe updates** to prevent errors in relational tables.
4. Validate and document **data quality checks** after cleaning.

---

3. SQL Script Guidelines

Each script in `SQL_Scripts/` folder will correspond to a table and contain:

1. **Trimming & whitespace removal** (`LTRIM` , `RTRIM` , `TRIM`)

2. **Case standardization** (`UPPER` , `LOWER` , capitalizing names)
3. **Removing unwanted characters** (`REPLACE` for special characters in phone/email)
4. **Handling NULLs and missing data** (`ISNULL` , `NULLIF`)
5. **Correcting numeric values** (`ABS` , valid ranges)
6. **Standardizing categorical data** (`CASE` statements)
7. **Duplicate removal** using `ROW_NUMBER()` and CTEs
8. **Data validation queries** for quality checks

Each script will be **idempotent**, allowing it to run multiple times without introducing errors.

---

4. Example Script: 01_Cleanup_Customers.sql

```
-- Remove leading/trailing spaces and standardize case
UPDATE Customers
SET
    FirstName = UPPER(LEFT(LTRIM(RTRIM(FirstName)),1)) +
LOWER(SUBSTRING(LTRIM(RTRIM(FirstName)),2,LEN(FirstName))),
    LastName = UPPER(LEFT(LTRIM(RTRIM(LastName)),1)) +
LOWER(SUBSTRING(LTRIM(RTRIM(LastName)),2,LEN(LastName))),
    Email = LOWER(LTRIM(RTRIM(Email))),
    Phone = REPLACE(REPLACE(REPLACE(LTRIM(RTRIM(Phone)),'-',''),'(',''),')',''),
    City = ISNULL(NULLIF(LTRIM(RTRIM(City)),'') , 'Unknown');

-- Remove duplicate customers by email
WITH CTE AS (
    SELECT *, ROW_NUMBER() OVER (PARTITION BY Email ORDER BY CustomerID) AS rn
    FROM Customers
)
DELETE FROM CTE
WHERE rn > 1;

-- Validate
SELECT COUNT(*) AS NullEmails FROM Customers WHERE Email IS NULL;
SELECT DISTINCT City FROM Customers;
```

---

5. Data Validation Queries

Keep reusable validation queries in a separate script (optional `DataValidation.sql`) to check:

- NULL values

- Duplicates
- Numeric ranges
- Categorical values

Example:

```sql
-- Check for duplicates
SELECT Email, COUNT(*) AS cnt
FROM Customers
GROUP BY Email
HAVING COUNT(*) > 1;

-- Check numeric ranges
SELECT MIN(Price), MAX(Price) FROM Products;
```

---

6. Project Workflow

1. Backup your database before running scripts.
2. Run scripts in order:
    1. `01_Cleanup_Customers.sql`
    2. `02_Cleanup_Products.sql`
    3. `03_Cleanup_Orders.sql`
    4. `04_Cleanup_Reviews.sql`
    5. `05_Cleanup_InventoryTransactions.sql`
3. Execute **DataValidation.sql** to confirm data quality.
4. Document any anomalies in `Documentation/` folder.
5. Use this cleaned data for upcoming ETL and transformation tutorials.

---

7. Project Notes

- All scripts are written **platform-neutral**, relying only on standard T-SQL.
- Modular design ensures scripts can be updated or extended as new cleaning requirements arise.
- This project will serve as the **base for all future tutorials**, providing a consistent, clean dataset.

Ready for next steps: Once this project is set up, we can start building tutorials for **Data Transformation, Aggregation, and ETL pipelines** using the cleaned `SalesAnalytics` dataset.