EX NO : 3

DATE :

Implementation of Data Cleaning, Transformation and Indexing Techniques

AIM :

To implement data cleaning, transformation and Indexing techniques in python using jupyter notebook

DATASET :

```
data = {
    'Title': ['The Great Gatsby', '1984', 'To Kill a Mockingbird', 'Pride and Prejudice', 'The Catcher in the Rye'],
    'Author': ['F. Scott Fitzgerald', 'George Orwell', 'Harper Lee', 'Jane Austen', 'J.D. Salinger'],
    'Publication Year': [1925, 1949, 1960, 1813, 1951],
    'Genre': ['Fiction', 'Dystopian', 'Fiction', 'Romance', 'Fiction'],
    'Price': [10.99, 8.99, 7.99, 6.99, 9.99],
    'Pages': [218, 328, 281, 279, 277],
    'Publisher': ['Scribner', 'Secker & Warburg', 'J.B. Lippincott & Co.', 'T. Egerton', 'Little, Brown and Company']
}
```

QUESTIONS :

1.How would you handle missing values in the Price column?

```
import pandas as pd

df['Price'].fillna(value=0)
print(df)
```

```
                  Title               Author  Publication Year      Genre  \
0       The Great Gatsby  F. Scott Fitzgerald              1925    Fiction
1                   1984        George Orwell              1949  Dystopian
2   To Kill a Mockingbird           Harper Lee              1960    Fiction
3     Pride and Prejudice          Jane Austen              1813    Romance
4  The Catcher in the Rye        J.D. Salinger              1951    Fiction

   Price  Pages                 Publisher
0  10.99    218                  Scribner
1   8.99    328          Secker & Warburg
2   7.99    281      J.B. Lippincott & Co.
3   6.99    279                T. Egerton
4   9.99    277  Little, Brown and Company
```

2. Replace missing values in the Price column with the median value of this column.

```
median_price = df['Price'].median()
df['Price'].fillna(value=median_price)
```

```
0    10.99
1     8.99
2     7.99
3     6.99
4     9.99
Name: Price, dtype: float64
```

3. Check for duplicate entries based on the Title column in the dataset.

```
duplicates = df[df.duplicated(subset='Title')]
print(duplicates)
```

```
Empty DataFrame
Columns: [Title, Author, Publication Year, Genre, Price, Pages, Publisher]
Index: []
```

4. Write a pandas function to convert the Publication Year column to integer format.

```
df['Publication Year'] = df['Publication Year'].astype(int)
print(df)
```

```
                   Title               Author  Publication Year      Genre  \
0       The Great Gatsby  F. Scott Fitzgerald              1925    Fiction
1                   1984        George Orwell              1949  Dystopian
2   To Kill a Mockingbird           Harper Lee              1960    Fiction
3     Pride and Prejudice          Jane Austen              1813    Romance
4  The Catcher in the Rye        J.D. Salinger              1951    Fiction
```

5. Drop rows from the dataset where the Publisher column has missing values.

```
df.dropna(subset=['Publisher'], inplace=True)
print(df)
```

```
                   Title               Author  Publication Year      Genre  \
0       The Great Gatsby  F. Scott Fitzgerald              1925    Fiction
1                   1984        George Orwell              1949  Dystopian
2   To Kill a Mockingbird           Harper Lee              1960    Fiction
3     Pride and Prejudice          Jane Austen              1813    Romance
4  The Catcher in the Rye        J.D. Salinger              1951    Fiction

   Price  Pages                  Publisher
0  10.99    218                   Scribner
1   8.99    328           Secker & Warburg
2   7.99    281      J.B. Lippincott & Co.
3   6.99    279                T. Egerton
4   9.99    277  Little, Brown and Company
```

6. Add a new column named PriceInDollars to the dataset, which multiplies the Price by 1.

```
df['PriceInDollars'] = df['Price'] * 1
print(df)
```

```
                   Title               Author  Publication Year      Genre  \
0       The Great Gatsby  F. Scott Fitzgerald              1925    Fiction
1                   1984        George Orwell              1949  Dystopian
2   To Kill a Mockingbird           Harper Lee              1960    Fiction
3     Pride and Prejudice          Jane Austen              1813    Romance
4  The Catcher in the Rye        J.D. Salinger              1951    Fiction

   Price  Pages                 Publisher  PriceInDollars
0  10.99    218                   Scribner           10.99
1   8.99    328           Secker & Warburg            8.99
2   7.99    281      J.B. Lippincott & Co.            7.99
3   6.99    279                 T. Egerton            6.99
4   9.99    277  Little, Brown and Company            9.99
```

7. Apply a function to increase the Price column values by 10%.

```
df['Price'] = df['Price'] * 1.10
print(df)
```

```
                   Title               Author  Publication Year      Genre  \
0       The Great Gatsby  F. Scott Fitzgerald              1925    Fiction
1                   1984        George Orwell              1949  Dystopian
2   To Kill a Mockingbird           Harper Lee              1960    Fiction
3     Pride and Prejudice          Jane Austen              1813    Romance
4  The Catcher in the Rye        J.D. Salinger              1951    Fiction

    Price  Pages                 Publisher  PriceInDollars
0  12.089    218                   Scribner           10.99
1   9.889    328           Secker & Warburg            8.99
2   8.789    281      J.B. Lippincott & Co.            7.99
3   7.689    279                 T. Egerton            6.99
4  10.989    277  Little, Brown and Company            9.99
```

8. Create a new column called PriceCategory that categorizes Price as 'Expensive' if greater than 9.99 and 'Affordable' otherwise.

```
df['PriceCategory'] = df['Price'].apply(lambda x: 'Expensive' if x > 9.99 else 'Affordable')
print(df)
```

```
                   Title               Author  Publication Year      Genre  \
0       The Great Gatsby  F. Scott Fitzgerald              1925    Fiction
1                   1984        George Orwell              1949  Dystopian
2   To Kill a Mockingbird           Harper Lee              1960    Fiction
3     Pride and Prejudice          Jane Austen              1813    Romance
4  The Catcher in the Rye        J.D. Salinger              1951    Fiction

    Price  Pages                 Publisher  PriceInDollars PriceCategory
0  12.089    218                   Scribner           10.99     Expensive
1   9.889    328           Secker & Warburg            8.99    Affordable
2   8.789    281      J.B. Lippincott & Co.            7.99    Affordable
3   7.689    279                 T. Egerton            6.99    Affordable
4  10.989    277  Little, Brown and Company            9.99     Expensive
```

9. Convert the Pages column to integer type.

```
df['Pages'] = df['Pages'].astype(int)
print(df)
```

```
                    Title               Author  Publication Year      Genre  \
0        The Great Gatsby  F. Scott Fitzgerald               1925    Fiction
1                    1984        George Orwell               1949  Dystopian
2   To Kill a Mockingbird           Harper Lee               1960    Fiction
3     Pride and Prejudice          Jane Austen               1813    Romance
4  The Catcher in the Rye        J.D. Salinger               1951    Fiction

    Price  Pages                  Publisher  PriceInDollars PriceCategory
0  12.089    218                   Scribner           10.99     Expensive
1   9.889    328           Secker & Warburg            8.99    Affordable
2   8.789    281      J.B. Lippincott & Co.            7.99    Affordable
3   7.689    279                 T. Egerton            6.99    Affordable
4  10.989    277  Little, Brown and Company            9.99     Expensive
```

10. Calculate the total sum of the Price column in the dataset.

```
total_price = df['Price'].sum()
print(total_price)
```

```
49.44500000000001
```

11. Create a new column named YearsSincePublication that calculates the number of years since the Publication Year.

```
current_year = 2024
df['YearsSincePublication'] = current_year - df['Publication Year']
print(df)
```

```
                    Title               Author  Publication Year      Genre  \
0        The Great Gatsby  F. Scott Fitzgerald               1925    Fiction
1                    1984        George Orwell               1949  Dystopian
2   To Kill a Mockingbird           Harper Lee               1960    Fiction
3     Pride and Prejudice          Jane Austen               1813    Romance
4  The Catcher in the Rye        J.D. Salinger               1951    Fiction

    Price  Pages                  Publisher  PriceInDollars PriceCategory  \
0  12.089    218                   Scribner           10.99     Expensive
1   9.889    328           Secker & Warburg            8.99    Affordable
2   8.789    281      J.B. Lippincott & Co.            7.99    Affordable
3   7.689    279                 T. Egerton            6.99    Affordable
4  10.989    277  Little, Brown and Company            9.99     Expensive

   YearsSincePublication
0                     99
1                     75
2                     64
3                    211
4                     73
```

12. Add a new column called GenreGroup that categorizes Genre as 'Classic' if 'Fiction' or 'Romance' and 'Modern' otherwise.

```
df['GenreGroup'] = df['Genre'].apply(lambda x: 'Classic' if x in ['Fiction', 'Romance'] else 'Modern')
print(df['GenreGroup'])
```

```
0    Classic
1     Modern
2    Classic
3    Classic
4    Classic
Name: GenreGroup, dtype: object
```

13. Replace all occurrences of missing values in the Publisher column with 'Unknown'.

```
df['Publisher'].fillna('Unknown')
print(df['Publisher'])
```

```
0                          Scribner
1                 Secker & Warburg
2          J.B. Lippincott & Co.
3                       T. Egerton
4     Little, Brown and Company
Name: Publisher, dtype: object
```

14. Create a new column PriceDifference which is the difference between the maximum price and each book's price.

```
max_price = df['Price'].max()
df['PriceDifference'] = max_price - df['Price']
print(df['PriceDifference'])
```

```
0    0.0
1    2.2
2    3.3
3    4.4
4    1.1
Name: PriceDifference, dtype: float64
```

15. Aggregate the dataset to find the total number of pages by Genre

```
pages_by_genre = df.groupby('Genre')['Pages'].sum()
print(pages_by_genre)
```

```
Genre
Dystopian    328
Fiction      776
Romance      279
Name: Pages, dtype: int64
```

16. Retrieve the rows for books published after 1950.

```
books_after_1950 = df[df['Publication Year'] > 1950]
print(books_after_1950)
```

```
                   Title         Author  Publication Year    Genre   Price  \
2   To Kill a Mockingbird     Harper Lee              1960  Fiction   8.789
4  The Catcher in the Rye   J.D. Salinger              1951  Fiction  10.989

   Pages                  Publisher  PriceInDollars PriceCategory  \
2    281      J.B. Lippincott & Co.            7.99    Affordable
4    277  Little, Brown and Company            9.99     Expensive

   YearsSincePublication GenreGroup  PriceDifference
2                     64    Classic              3.3
4                     73    Classic              1.1
```

17. Remove rows where the Price is less than 8.00.

```
df = df[df['Price'] >= 8.00]
print(df)
```

```
                   Title               Author  Publication Year      Genre  \
0       The Great Gatsby  F. Scott Fitzgerald              1925    Fiction
1                   1984        George Orwell              1949  Dystopian
2   To Kill a Mockingbird           Harper Lee              1960    Fiction
4  The Catcher in the Rye        J.D. Salinger              1951    Fiction

    Price  Pages                 Publisher  PriceInDollars PriceCategory  \
0  12.089    218                   Scribner           10.99     Expensive
1   9.889    328           Secker & Warburg            8.99    Affordable
2   8.789    281      J.B. Lippincott & Co.            7.99    Affordable
4  10.989    277  Little, Brown and Company            9.99     Expensive
```

18. Filter the dataset to show books that have both Price greater than 9.00 and Pages greater than 270.

```
filtered_books = df[(df['Price'] > 9.00) & (df['Pages'] > 270)]
print(filtered_books)
```

```
                   Title         Author  Publication Year      Genre   Price  \
1                   1984  George Orwell              1949  Dystopian   9.889
4  The Catcher in the Rye  J.D. Salinger              1951    Fiction  10.989

   Pages                 Publisher  PriceInDollars PriceCategory  \
1    328           Secker & Warburg            8.99    Affordable
4    277  Little, Brown and Company            9.99     Expensive

   YearsSincePublication GenreGroup  PriceDifference
1                     75     Modern              2.2
4                     73    Classic              1.1
```

19. Create a new column PublicationDecade from the Publication Year column in 'YYYYs' format (e.g., '1920s')

```
df['PublicationDecade'] = (df['Publication Year'] // 10 * 10).astype(str) + 's'
print(df)
```

```
                   Title               Author  Publication Year      Genre  \
0       The Great Gatsby  F. Scott Fitzgerald              1925    Fiction
1                   1984        George Orwell              1949  Dystopian
2   To Kill a Mockingbird           Harper Lee              1960    Fiction
4  The Catcher in the Rye        J.D. Salinger              1951    Fiction

    Price  Pages                 Publisher  PriceInDollars PriceCategory  \
0  12.089    218                   Scribner           10.99     Expensive
1   9.889    328           Secker & Warburg            8.99    Affordable
2   8.789    281      J.B. Lippincott & Co.            7.99    Affordable
4  10.989    277  Little, Brown and Company            9.99     Expensive

   YearsSincePublication GenreGroup  PriceDifference PublicationDecade
0                     99    Classic              0.0             1920s
1                     75     Modern              2.2             1940s
2                     64    Classic              3.3             1960s
4                     73    Classic              1.1             1950s
```

20. How do you check for duplicates based on the Author column?

```
author_duplicates = df[df.duplicated(subset='Author', keep=False)]
print(author_duplicates)
```

```
Empty DataFrame
Columns: [Title, Author, Publication Year, Genre, Price, Pages, Publisher, PriceInDollars, PriceCategory, YearsSincePublication, GenreGroup, PriceDifference, Pu
blicationDecade]
Index: []
```

RUBRICS

Problem Understanding (10)	Implementation (20)	Viva (10)	Time Management (10)	Total (50)

RESULT

Thus the implementation of data cleaning,transformation and Indexing techniques in python using jupyter notebook was successfully executed and the output was verified.