# DATA SCIENCE PRACTICAL – SHORT CODES

------------------------------------

## 6. BUILD TIME HUB, LINKS, SATELLITES

------------------------------------

### Hub (Unique Keys)

-----------------

```
import pandas as pd
import hashlib

df = pd.DataFrame({
'CustomerID': [101, 102, 103],
'Name': ['A', 'B', 'C'],
'City': ['Mumbai', 'Delhi', 'Pune']
})

def hash_key(x):
return hashlib.sha256(str(x).encode()).hexdigest()

hub = pd.DataFrame({
'HubCustomerKey': df['CustomerID'].apply(hash_key),
'CustomerID': df['CustomerID']
})
```

### Link (Key Relationships)

------------------------

```
orders = pd.DataFrame({
'OrderID': [1, 2, 3],
'CustomerID': [101, 102, 101]
})

link = pd.DataFrame({
'LinkKey': (orders['OrderID'].astype(str) + orders['CustomerID'].astype(str)).apply(hash_key),
'OrderID': orders['OrderID'],
'CustomerID': orders['CustomerID']
})
```

Satellite (Attributes)
----------------------

```
sat = df[['CustomerID', 'Name', 'City']]
```

---------------------

7. TRANSFORMING DATA

---------------------

```
df = pd.DataFrame({
'Name': ['A', None, 'C'],
'Age': [20, 25, None]
})

df['Name'] = df['Name'].fillna('Unknown')
df['Age'] = df['Age'].fillna(df['Age'].mean())
df['Age_norm'] = (df['Age'] - df['Age'].min()) / (df['Age'].max() - df['Age'].min())
```

-------------------

8. ORGANIZING DATA

-------------------

```
df = pd.DataFrame({
'Dept': ['IT', 'IT', 'HR'],
'Salary': [50000, 60000, 45000]
})

df_sorted = df.sort_values('Salary')
df_group = df.groupby('Dept')['Salary'].mean()
df_pivot = df.pivot_table(values='Salary', index='Dept', aggfunc='mean')
```

------------------

## 9. GENERATING DATA

------------------

```
import numpy as np
np.random.seed(42)

data = pd.DataFrame({
'Age': np.random.randint(18, 60, 10),
'Salary': np.random.normal(50000, 8000, 10).astype(int)
})
```