

```

import pandas as pd

# ===== Create Sample Student Dataset =====
students = pd.DataFrame({
    "student_id": [1, 2, 3, 4, 5],
    "name": ["Ravi", "Sneha", "Arjun", "Meena", "Kiran"],
    "class": ["A", "B", "A", "B", "A"]
})

scores = pd.DataFrame({
    "student_id": [1, 2, 3, 5, 6],
    "math": [85, 90, 78, 88, 92],
    "science": [80, 85, 75, 89, 95]
})

print("Students:\n", students)
print("\nScores:\n", scores)

# ===== Merge Operations =====
inner_merge = pd.merge(students, scores, on="student_id", how="inner")
left_merge = pd.merge(students, scores, on="student_id", how="left")
right_merge = pd.merge(students, scores, on="student_id", how="right")

print("\n=== Inner Join ===")
print(inner_merge)

print("\n=== Left Join ===")
print(left_merge)

print("\n=== Right Join ===")
print(right_merge)

# ===== Aggregate Statistics =====
print("\n=== Aggregate Statistics ===")
print(inner_merge[["math", "science"]].agg(["mean", "max", "min", "count"]))

# Group by class and compute mean scores
group_stats = inner_merge.groupby("class")[["math", "science"]].mean()
print("\n=== Grouped Statistics by Class ===")
print(group_stats)

```

Students:

	student_id	name	class
0	1	Ravi	A
1	2	Sneha	B
2	3	Arjun	A
3	4	Meena	B
4	5	Kiran	A

Scores:

	student_id	math	science
0	1	85	80
1	2	90	85
2	3	78	75
3	5	88	89
4	6	92	95

=== Inner Join ===

	student_id	name	class	math	science
0	1	Ravi	A	85	80
1	2	Sneha	B	90	85
2	3	Arjun	A	78	75
3	5	Kiran	A	88	89

=== Left Join ===

	student_id	name	class	math	science
0	1	Ravi	A	85.0	80.0
1	2	Sneha	B	90.0	85.0
2	3	Arjun	A	78.0	75.0
3	4	Meena	B	NaN	NaN
4	5	Kiran	A	88.0	89.0

=== Right Join ===

	student_id	name	class	math	science
0	1	Ravi	A	85	80
1	2	Sneha	B	90	85
2	3	Arjun	A	78	75
3	5	Kiran	A	88	89

```

4          6   NaN   NaN   92      95

=== Aggregate Statistics ===
      math  science
mean  85.25   82.25
max   90.00   89.00
min   78.00   75.00
count  4.00    4.00

=== Grouped Statistics by Class ===
      math  science
class
A      83.666667  81.333333
B      90.000000  85.000000

# ===== Create Sample Customers & Sales Dataset =====
customers = pd.DataFrame({
    "customer_id": [101, 102, 103, 104],
    "name": ["Alice", "Bob", "Charlie", "David"],
    "region": ["North", "South", "North", "East"]
})

sales = pd.DataFrame({
    "sale_id": [1, 2, 3, 4, 5],
    "customer_id": [101, 102, 103, 101, 105],
    "amount": [500, 700, 300, 400, 900]
})

print("Customers:\n", customers)
print("\nSales:\n", sales)

# ===== Merge Operations =====
inner_sales = pd.merge(customers, sales, on="customer_id", how="inner")
left_sales = pd.merge(customers, sales, on="customer_id", how="left")

print("\n=== Inner Join (Customers with Sales) ===")
print(inner_sales)

print("\n=== Left Join (All Customers, Sales if available) ===")
print(left_sales)

# ===== Aggregate Sales Statistics =====
print("\n=== Overall Sales Statistics ===")
print(inner_sales["amount"].agg(["sum", "mean", "max", "min", "count"]))

# Sales per customer
sales_per_customer = inner_sales.groupby("name")["amount"].sum()
print("\n=== Sales per Customer ===")
print(sales_per_customer)

# Sales by region
sales_by_region = inner_sales.groupby("region")["amount"].agg(["sum", "mean", "count"])
print("\n=== Sales by Region ===")
print(sales_by_region)

```

↔ Customers:

	customer_id	name	region
0	101	Alice	North
1	102	Bob	South
2	103	Charlie	North
3	104	David	East

Sales:

	sale_id	customer_id	amount
0	1	101	500
1	2	102	700
2	3	103	300
3	4	101	400
4	5	105	900

```

=== Inner Join (Customers with Sales) ===
   customer_id  name region  sale_id  amount
0          101  Alice  North         1     500
1          101  Alice  North         4     400
2          102   Bob   South         2     700
3          103  Charlie North         3     300

```

```

=== Left Join (All Customers, Sales if available) ===
   customer_id  name region  sale_id  amount
0          101  Alice  North         1     500
1          101  Alice  North         4     400
2          102   Bob   South         2     700
3          103  Charlie North         3     300

```

0	101	Alice	North	1.0	500.0
1	101	Alice	North	4.0	400.0
2	102	Bob	South	2.0	700.0
3	103	Charlie	North	3.0	300.0
4	104	David	East	NaN	NaN

=== Overall Sales Statistics ===

sum	1900.0
mean	475.0
max	700.0
min	300.0
count	4.0

Name: amount, dtype: float64

=== Sales per Customer ===

name	
Alice	900
Bob	700
Charlie	300

Name: amount, dtype: int64

=== Sales by Region ===

	sum	mean	count
region			
North	1200	400.0	3
South	700	700.0	1