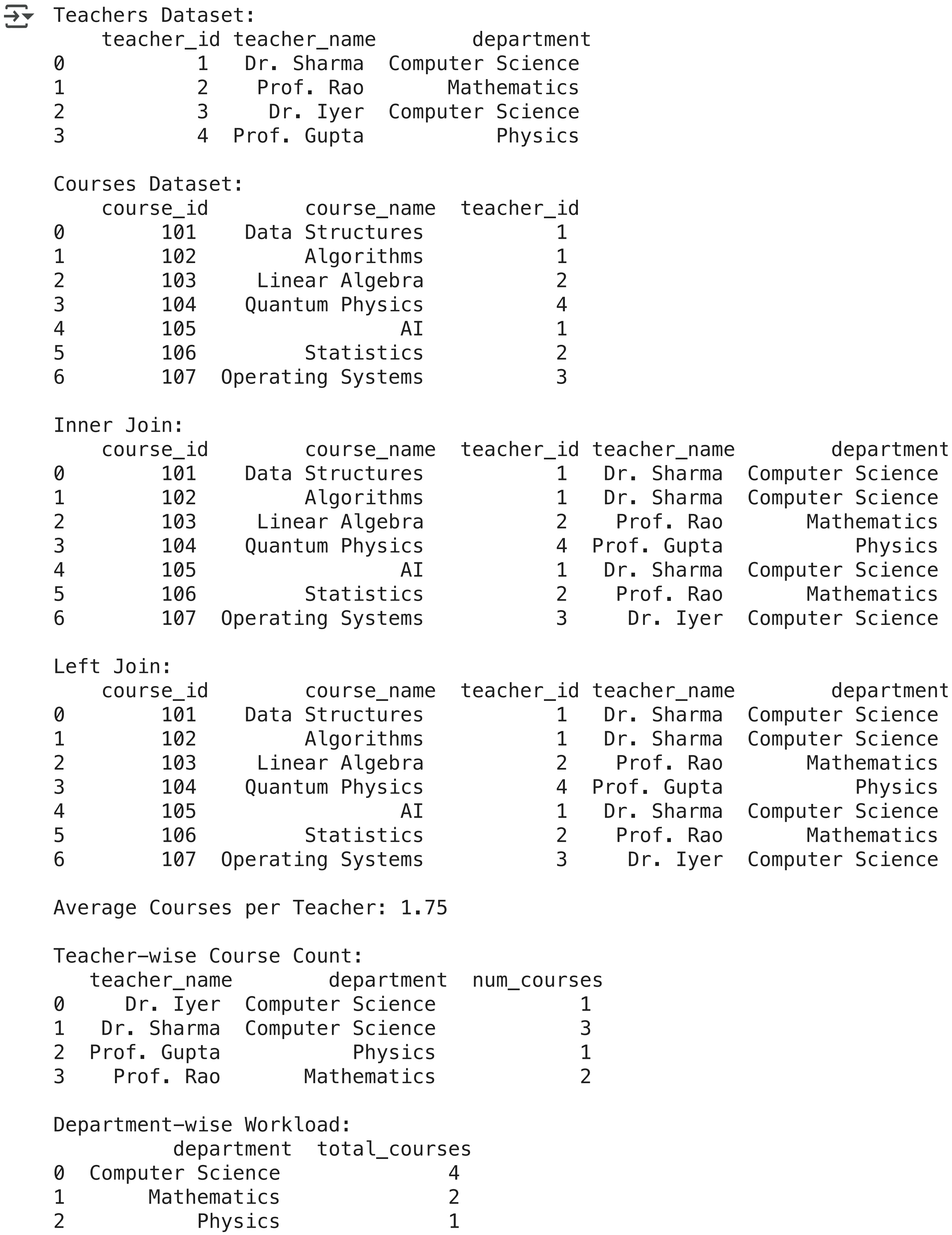
```
import pandas as pd

products_data = {
    "product_id": [101, 102, 103, 104, 105],
    "product_name": ["Laptop", "Mouse", "Keyboard", "Monitor", "Printer"],
    "category": ["Electronics", "Accessories", "Accessories", "Electronics", "El
    "price": [60000, 800, 1500, 12000, 7000]
}

products_df = pd.DataFrame(products_data)
print("Products Dataset:\n", products_df, "\n")

sales_data = {
    "sale_id": [1, 2, 3, 4, 5, 6, 7],
    "product_id": [101, 102, 103, 104, 101, 102, 105],
    "quantity": [3, 10, 5, 2, 1, 4, 6]
}

sales_df = pd.DataFrame(sales_data)
print("Sales Dataset:\n", sales_df, "\n")

inner_merged = pd.merge(sales_df, products_df, on="product_id", how="inner")
print("Inner Join:\n", inner_merged, "\n")

outer_merged = pd.merge(sales_df, products_df, on="product_id", how="outer")
print("Outer Join:\n", outer_merged, "\n")

inner_merged["revenue"] = inner_merged["quantity"] * inner_merged["price"]

total_revenue = inner_merged["revenue"].sum()
average_price = inner_merged["price"].mean()
product_sales = inner_merged.groupby("product_name")["quantity"].sum().reset_ind
product_revenue = inner_merged.groupby("product_name")["revenue"].sum().reset_in
category_sales = inner_merged.groupby("category")["revenue"].sum().reset_index()
best_category = category_sales.loc[category_sales["revenue"].idxmax()]

print("Total Revenue:", total_revenue)
print("Average Price:", average_price, "\n")
print("Product-wise Sales:\n", product_sales, "\n")
print("Revenue by Product:\n", product_revenue, "\n")
print("Revenue by Category:\n", category_sales, "\n")
print("Best-Selling Category:", best_category["category"], "with revenue", best_
```

```
Products Dataset:
     product_id product_name     category  price
0          101       Laptop  Electronics  60000
1          102        Mouse  Accessories    800
2          103     Keyboard  Accessories   1500
3          104      Monitor  Electronics  12000
```

```
4         105      Printer  Electronics   7000

Sales Dataset:
    sale_id  product_id  quantity
0        1         101         3
1        2         102        10
2        3         103         5
3        4         104         2
4        5         101         1
5        6         102         4
6        7         105         6

Inner Join:
    sale_id  product_id  quantity product_name     category  price
0        1         101         3       Laptop  Electronics  60000
1        2         102        10        Mouse  Accessories    800
2        3         103         5     Keyboard  Accessories   1500
3        4         104         2      Monitor  Electronics  12000
4        5         101         1       Laptop  Electronics  60000
5        6         102         4        Mouse  Accessories    800
6        7         105         6      Printer  Electronics   7000

Outer Join:
    sale_id  product_id  quantity product_name     category  price
0        1         101         3       Laptop  Electronics  60000
1        5         101         1       Laptop  Electronics  60000
2        2         102        10        Mouse  Accessories    800
3        6         102         4        Mouse  Accessories    800
4        3         103         5     Keyboard  Accessories   1500
5        4         104         2      Monitor  Electronics  12000
6        7         105         6      Printer  Electronics   7000

Total Revenue: 324700
Average Price: 20300.0

Product-wise Sales:
   product_name  quantity
0     Keyboard         5
1       Laptop         4
2      Monitor         2
3        Mouse        14
4      Printer         6

Revenue by Product:
   product_name  revenue
0     Keyboard     7500
1       Laptop   240000
2      Monitor    24000
3        Mouse    11200
4      Printer    42000

Revenue by Category:
       category  revenue
```

```
import pandas as pd

teachers_data = {
    "teacher_id": [1, 2, 3, 4],
    "teacher_name": ["Dr. Sharma", "Prof. Rao", "Dr. Iyer", "Prof. Gupta"],
    "department": ["Computer Science", "Mathematics", "Computer Science", "Phys
}

teachers_df = pd.DataFrame(teachers_data)
print("Teachers Dataset:\n", teachers_df, "\n")

courses_data = {
    "course_id": [101, 102, 103, 104, 105, 106, 107],
    "course_name": ["Data Structures", "Algorithms", "Linear Algebra", "Quantum
    "teacher_id": [1, 1, 2, 4, 1, 2, 3]
}

courses_df = pd.DataFrame(courses_data)
print("Courses Dataset:\n", courses_df, "\n")

inner_merged = pd.merge(courses_df, teachers_df, on="teacher_id", how="inner")
print("Inner Join:\n", inner_merged, "\n")

left_merged = pd.merge(courses_df, teachers_df, on="teacher_id", how="left")
print("Left Join:\n", left_merged, "\n")

avg_courses_per_teacher = inner_merged.groupby("teacher_name")["course_id"].cou

teacher_course_count = inner_merged.groupby(["teacher_name", "department"])["co
teacher_course_count.rename(columns={"course_id": "num_courses"}, inplace=True)

department_workload = inner_merged.groupby("department")["course_id"].count().r
department_workload.rename(columns={"course_id": "total_courses"}, inplace=True

print("Average Courses per Teacher:", avg_courses_per_teacher, "\n")
print("Teacher-wise Course Count:\n", teacher_course_count, "\n")
print("Department-wise Workload:\n", department_workload, "\n")
```



```
Teachers Dataset:
   teacher_id teacher_name        department
0           1   Dr. Sharma  Computer Science
1           2    Prof. Rao       Mathematics
2           3     Dr. Iyer  Computer Science
3           4  Prof. Gupta           Physics

Courses Dataset:
   course_id        course_name  teacher_id
0        101    Data Structures           1
1        102         Algorithms           1
2        103     Linear Algebra           2
3        104    Quantum Physics           4
4        105                 AI           1
5        106         Statistics           2
6        107  Operating Systems           3

Inner Join:
   course_id        course_name  teacher_id teacher_name        department
0        101    Data Structures           1   Dr. Sharma  Computer Science
1        102         Algorithms           1   Dr. Sharma  Computer Science
2        103     Linear Algebra           2    Prof. Rao       Mathematics
3        104    Quantum Physics           4  Prof. Gupta           Physics
4        105                 AI           1   Dr. Sharma  Computer Science
5        106         Statistics           2    Prof. Rao       Mathematics
6        107  Operating Systems           3     Dr. Iyer  Computer Science

Left Join:
   course_id        course_name  teacher_id teacher_name        department
0        101    Data Structures           1   Dr. Sharma  Computer Science
1        102         Algorithms           1   Dr. Sharma  Computer Science
2        103     Linear Algebra           2    Prof. Rao       Mathematics
3        104    Quantum Physics           4  Prof. Gupta           Physics
4        105                 AI           1   Dr. Sharma  Computer Science
5        106         Statistics           2    Prof. Rao       Mathematics
6        107  Operating Systems           3     Dr. Iyer  Computer Science

Average Courses per Teacher: 1.75

Teacher-wise Course Count:
  teacher_name        department  num_courses
0     Dr. Iyer  Computer Science            1
1   Dr. Sharma  Computer Science            3
2  Prof. Gupta           Physics            1
3    Prof. Rao       Mathematics            2

Department-wise Workload:
         department  total_courses
0  Computer Science              4
1       Mathematics              2
2           Physics              1
```