Dataset: E-commerce Transactions

This dataset contains information about e-commerce transactions, including details about the products purchased, the customer who made the purchase, the price, and the transaction date.

Sample Data:

```
transaction_id,customer_id,product,category,price,quantity,discount_percentage,transacti

1,101,Laptop,Electronics,1000,1,10,2023-08-01
2,102,Smartphone,Electronics,700,2,5,2023-08-01
3,103,Shirt,Fashion,40,3,0,2023-08-02
4,104,Blender,Home Appliance,150,1,15,2023-08-03
5,101,Headphones,Electronics,100,2,10,2023-08-03
6,105,Shoes,Fashion,60,1,20,2023-08-04
7,106,Refrigerator,Home Appliance,800,1,25,2023-08-05
8,107,Book,Books,20,4,0,2023-08-05
9,108,Toaster,Home Appliance,30,1,5,2023-08-06
10,102,Tablet,Electronics,300,2,10,2023-08-06
```

Exercises:

1. **Calculate the Total Revenue per Category**
   - Group the data by `category` and calculate the total revenue generated by each category. (Hint: Multiply `price` by `quantity` and apply the discount to get the actual revenue.)
2. **Filter Transactions with a Discount Greater Than 10%**
   - Filter the dataset to show only transactions where the discount percentage is greater than 10%.
3. **Find the Most Expensive Product Sold**
   - Identify the product with the highest individual price.
4. **Calculate the Average Quantity of Products Sold per Category**
   - Group the data by `category` and calculate the average quantity of products sold in each category.
5. **Identify Customers Who Purchased More Than One Product**
   - Filter the data to show only customers who purchased more than one product in a single transaction.
6. **Find the Top 3 Highest Revenue Transactions**
   - Calculate the total revenue for each transaction and identify the top 3 highest revenue transactions.
7. **Calculate the Total Number of Transactions per Day**
   - Group the data by `transaction_date` and calculate the total number of transactions for each day.
8. **Find the Customer Who Spent the Most Money**

    - Calculate the total amount spent by each customer and identify the customer with the highest total spending.
9. **Calculate the Average Discount Given per Product Category**
    - Group the data by `category` and calculate the average discount percentage applied to products in each category.
10. **Create a New Column for Final Price After Discount**
    - Add a new column `final_price` that calculates the total price after applying the discount (`price - (price * discount_percentage / 100)`).

Dataset: Banking Transactions

This dataset contains information about customer transactions at a bank. Each row represents a transaction, including the transaction ID, customer ID, transaction type, amount, and date.

Sample Data:

```
transaction_id,customer_id,transaction_type,amount,transaction_date
1,201,Deposit,5000,2023-09-01
2,202,Withdrawal,2000,2023-09-01
3,203,Deposit,3000,2023-09-02
4,201,Withdrawal,1500,2023-09-02
5,204,Deposit,10000,2023-09-03
6,205,Withdrawal,500,2023-09-03
7,202,Deposit,2500,2023-09-04
8,206,Withdrawal,700,2023-09-04
9,203,Deposit,4000,2023-09-05
10,204,Withdrawal,3000,2023-09-05
```

Exercises:

1. **Calculate the Total Deposit and Withdrawal Amounts**
    - Group the data by `transaction_type` and calculate the total amounts for both deposits and withdrawals.
2. **Filter Transactions Greater Than $3,000**
    - Filter the dataset to show only transactions where the `amount` is greater than $3,000.
3. **Find the Largest Deposit Made**
    - Identify the transaction with the highest deposit amount.
4. **Calculate the Average Transaction Amount for Each Transaction Type**
    - Group the data by `transaction_type` and calculate the average amount for deposits and withdrawals.
5. **Find Customers Who Made Both Deposits and Withdrawals**

   - Identify customers who have made at least one deposit and one withdrawal.

6. **Calculate the Total Amount of Transactions per Day**
   - Group the data by `transaction_date` and calculate the total amount of all transactions for each day.
7. **Find the Customer with the Highest Total Withdrawal**
   - Calculate the total amount withdrawn by each customer and identify the customer with the highest total withdrawal.
8. **Calculate the Number of Transactions for Each Customer**
   - Group the data by `customer_id` and calculate the total number of transactions made by each customer.
9. **Find All Transactions That Occurred on the Same Day as a Withdrawal Greater Than $1,000**
   - Filter the data to show all transactions that occurred on the same day as a withdrawal of more than $1,000.
10. **Create a New Column to Classify Transactions as "High" or "Low" Value**
    - Add a new column `transaction_value` that classifies a transaction as "High" if the `amount` is greater than $5,000, otherwise classify it as "Low."

Dataset: Health & Fitness Tracker Data

This dataset contains information about users' daily health and fitness activities, including steps taken, calories burned, hours of sleep, and workout types.

Sample Data:

```
user_id,date,steps,calories_burned,hours_of_sleep,workout_type
1,2023-09-01,12000,500,7,Cardio
2,2023-09-01,8000,400,6.5,Strength
3,2023-09-01,15000,650,8,Yoga
1,2023-09-02,10000,450,6,Cardio
2,2023-09-02,9500,500,7,Cardio
3,2023-09-02,14000,600,7.5,Strength
1,2023-09-03,13000,550,8,Yoga
2,2023-09-03,12000,520,6.5,Yoga
3,2023-09-03,16000,700,7,Cardio
```

Exercises:

1. **Find the Total Steps Taken by Each User**
   - Group the data by `user_id` and calculate the total steps taken by each user across all days.
2. **Filter Days with More Than 10,000 Steps**

    - Filter the dataset to show only the days where the user took more than 10,000 steps.

3. **Calculate the Average Calories Burned by Workout Type**
    - Group the data by `workout_type` and calculate the average calories burned for each workout type.
4. **Identify the Day with the Most Steps for Each User**
    - For each user, find the day when they took the most steps.
5. **Find Users Who Burned More Than 600 Calories on Any Day**
    - Filter the data to show only the users who burned more than 600 calories on any day.
6. **Calculate the Average Hours of Sleep per User**
    - Group the data by `user_id` and calculate the average hours of sleep for each user.
7. **Find the Total Calories Burned per Day**
    - Group the data by `date` and calculate the total calories burned by all users combined for each day.
8. **Identify Users Who Did Different Types of Workouts**
    - Identify users who participated in more than one type of workout.
9. **Calculate the Total Number of Workouts per User**
    - Group the data by `user_id` and count the total number of workouts completed by each user.
10. **Create a New Column for "Active" Days**
    - Add a new column called `active_day` that classifies a day as "Active" if the user took more than 10,000 steps, otherwise classify it as "Inactive."

Dataset: Music Streaming Data

This dataset contains information about users' music streaming habits, including the **song title**, **artist**, **duration of the song**, **streaming time**, and **user's location**.

Sample Data:

```
user_id,song_title,artist,duration_seconds,streaming_time,location
1,Blinding Lights,The Weeknd,200,2023-09-01 08:15:00,New York
2,Shape of You,Ed Sheeran,240,2023-09-01 09:20:00,Los Angeles
3,Levitating,Dua Lipa,180,2023-09-01 10:30:00,London
1,Starboy,The Weeknd,220,2023-09-01 11:00:00,New York
2,Perfect,Ed Sheeran,250,2023-09-01 12:15:00,Los Angeles
3,Don't Start Now,Dua Lipa,200,2023-09-02 08:10:00,London
1,Save Your Tears,The Weeknd,210,2023-09-02 09:00:00,New York
2,Galway Girl,Ed Sheeran,190,2023-09-02 10:00:00,Los Angeles
3,New Rules,Dua Lipa,230,2023-09-02 11:00:00,London
```

Exercises:

1. **Calculate the Total Listening Time for Each User**
   - Group the data by `user_id` and calculate the total time spent streaming (in seconds) for each user.
2. **Filter Songs Streamed for More Than 200 Seconds**
   - Filter the dataset to show only the songs where the `duration_seconds` is greater than 200.
3. **Find the Most Popular Artist (by Total Streams)**
   - Group the data by `artist` and find the artist with the most streams (i.e., the highest number of song plays).
4. **Identify the Song with the Longest Duration**
   - Identify the song with the longest duration in the dataset.
5. **Calculate the Average Song Duration by Artist**
   - Group the data by `artist` and calculate the average song duration for each artist.
6. **Find the Top 3 Most Streamed Songs per User**
   - For each user, find the top 3 most-streamed songs (i.e., songs they played most frequently).
7. **Calculate the Total Number of Streams per Day**
   - Group the data by `streaming_time` (by extracting the date) and calculate the total number of streams for each day.
8. **Identify Users Who Streamed Songs from More Than One Artist**
   - Find users who listened to songs by more than one artist.
9. **Calculate the Total Streams for Each Location**
   - Group the data by `location` and calculate the total number of streams for each location.
10. **Create a New Column to Classify Long and Short Songs**
    - Add a new column `song_length` that classifies a song as "Long" if `duration_seconds > 200`, otherwise classify it as "Short."

Dataset: Retail Store Sales Data

This dataset contains information about sales transactions at a retail store, including the **product name**, **category**, **price**, **quantity sold**, and **sales date**.

Sample Data:

```
transaction_id,product_name,category,price,quantity,sales_date
1,Apple,Groceries,0.50,10,2023-09-01
2,T-shirt,Clothing,15.00,2,2023-09-01
```

```
3,Notebook,Stationery,2.00,5,2023-09-02
4,Banana,Groceries,0.30,12,2023-09-02
5,Laptop,Electronics,800.00,1,2023-09-03
6,Pants,Clothing,25.00,3,2023-09-03
7,Headphones,Electronics,100.00,2,2023-09-04
8,Pen,Stationery,1.00,10,2023-09-04
9,Orange,Groceries,0.60,8,2023-09-05
10,Sneakers,Clothing,50.00,1,2023-09-05
```

Exercises:

1. **Calculate the Total Revenue per Category**
   - Group the data by `category` and calculate the total revenue generated by each category. (Hint: Multiply `price` by `quantity` for each transaction.)
2. **Filter Transactions Where the Total Sales Amount is Greater Than $100**
   - Filter the dataset to show only transactions where the total sales amount (price * quantity) is greater than $100.
3. **Find the Most Sold Product**
   - Identify the product with the highest total quantity sold across all transactions.
4. **Calculate the Average Price per Product Category**
   - Group the data by `category` and calculate the average price of products in each category.
5. **Find the Top 3 Highest Grossing Products**
   - Calculate the total revenue for each product and identify the top 3 products that generated the most revenue.
6. **Calculate the Total Number of Items Sold per Day**
   - Group the data by `sales_date` and calculate the total quantity of items sold for each day.
7. **Identify the Product with the Lowest Price in Each Category**
   - For each category, identify the product with the lowest price.
8. **Calculate the Total Revenue for Each Product**
   - Group the data by `product_name` and calculate the total revenue generated by each product.
9. **Find the Total Sales per Day for Each Category**
   - Group the data by `sales_date` and `category` to calculate the total sales for each category per day.
10. **Create a New Column for Discounted Price**
    - Add a new column called `discounted_price` that applies a 10% discount to the original price for each product (`price * 0.9`).