Redis Commands To Be Used

Addition 1:
I need to leverage Redis Hashes for storing trip details, therefore I need to maintain a structured and efficient data storage mechanism. These are the commands I will be using to do so. This implementation will have all CRUD operations used.

1. Initialize:
    - This is deletes all the data from the databases in Redis so the application will have a clean slate
    - **FLUSHALL**
2. Create trip details (CREATE):
    - This command creates a new trip entry in Redis by setting multiple field-value pairs in the Hash associated with the specified trip ID.
    - **HMSET** trip:<trip_id> field1 value1 field2 value2 …
3. Read trip details (READ):
    - This command retrieves all the field-value pairs stored in the Redis Hash associated with the specified trip ID, allowing us to read the details of a trip.
    - **HGETALL** trip:<trip_id>
4. Update trip details (UPDATE):
    - This command updates the existing trip details in Redis by setting multiple field-value pairs in the Hash associated with the specified trip ID.
    - **HMSET** trip:<trip_id> field1 value1 field2 value2 …
5. Delete trip details (DELETE):
    - This command deletes the trip entry from Redis, including all its associated field-value pairs in the Hash, based on the specified trip ID.
    - **DEL** trip:<trip_id>

Addition 2:
To efficiently manage and display the "Most Viewed Properties" using Redis, we need to employ a robust and scalable data storage strategy. The commands outlined below will be utilized to achieve this, covering a comprehensive set of operations to handle property views effectively. This approach leverages Redis's powerful data manipulation capabilities to ensure real-time updates and quick data retrieval, crucial for maintaining an interactive and responsive user experience.

1. Initialize:
    - Clears the entire Redis database to start fresh
    - **FLUSHALL**
2. Increment the View Count for a Property

    - Increases the view count of a property each time it is viewed, reflecting its growing popularity.
    - **ZINCRBY mostViewedProperties 1 propertyId**
3. Retrieve the Top 10 Most Viewed Properties
    - Fetches the properties with the highest view counts
    - **ZRANGE mostViewedProperties 0 9 WITHSCORES**
4. Get the View Count for a Specific Property
    - Retrieves the current view count of a specific property.
    - **ZSCORE mostViewedProperties propertyId**
5. Find the Rank of a Specific Property
    - Determines the rank of a property based on its view count within the sorted list.
    - **ZRANK mostViewedProperties propertyId**
6. Update the View Count for a Property Directly
    - Sets a specific view count for a property, useful for corrections or batch updates.
    - **ZADD mostViewedProperties score propertyId**