Task: Object-Oriented Programming with a Deck of Cards

Objective:

The objective of this task is to reinforce your understanding of object-oriented programming in Python by creating classes to represent playing cards and decks, and implementing various functionalities.

Requirements:

1. Create a **`Card`** class to represent individual playing cards. Each card should have attributes for its suit (e.g., hearts, diamonds) and rank (e.g., 2, 3, queen, king).

2. Create a **`Deck`** class to represent a standard deck of cards. The deck should be initialized with 52 cards, one for each rank in each of the four suits.

3. Implement the following functionalities for the **`Deck`** class:
   - **`shuffle()`**: A method to shuffle the cards in the deck.
   - **`draw()`**: A method to draw a single card from the deck. This should remove the card from the deck.
   - **`__len__()`**: Overload the len() operator to return the number of cards in the deck.
   - **`__str__()`**: Overload the str() operator to provide a human-readable representation of the deck.
   - **`__getitem__()`**: Overload the [] operator to access cards in the deck by index.
   - **`sort_by_suit()`**: A method to sort the deck by suit.
   - **`sort_by_rank()`**: A method to sort the deck by rank.

4. Add the option to include joker cards in the deck (optional).

5. Implement a **`deal_hand(num_cards)`** method to deal a specified number of cards from the deck into a hand.

6. Implement a **`count_cards()`** method to count how many cards of each rank are in the deck.

7. Implement the **__lt__** and **__gt__** magic methods in the '**Card**' class to allow comparisons between cards based on their ranks.

8. Create a demonstration script that showcases the use of the `**Deck**` class, including shuffling, sorting, dealing hands, and displaying the current state of the deck.