Django ORM Task 2

Note - avoid N+1 query problems

1. Using the shell plus and the django project you created before, run these queries to:
   - -- fetch all the objects of authors with only id, first name and last name
   - -- fetch all the objects of books as a list that contains name
   - -- fetch all the objects of books as a tuple that contains name and count
   - -- fetch an object from books table using id and update the count.
   - -- fetch an object from the books using name and update the author of that book
   - -- fetch all books from the table and show the name of the books and full name of authors as a list of dictionaries
   - -- fetch all the authors from the table and show the name of authors with the list of names of the books of that author in a list of dictionaries format
   - -- fetch all the books from the table and show the name, price and count of the books in a list of dictionary format inside the queryset output
   - -- fetch an author from the table and increase the price with 100 for all the books of that author
   - -- fetch all the authors from the table and show the name of authors with the list of names of the books of that author that have an average rating greater than 3 in a list of dictionaries format
   - -- fetch all the authors from the table and show the name of authors with the list of names of the books of that author that have an average rating less than or equal to 3 in a list of dictionaries format
   - -- fetch all the books from the table that has an average rating as 3
   - -- fetch all the authors from the table that have books_count less than 10
   - -- fetch all the authors from the table and update the books_count value with respect to the exact number of books connected to that author
   - -- fetch all the books from the table and avoid count field while fetching the objects
   - -- featch all the authors in a queryset showing list of values that contains first name and last name of that author