**EXERCISES:**

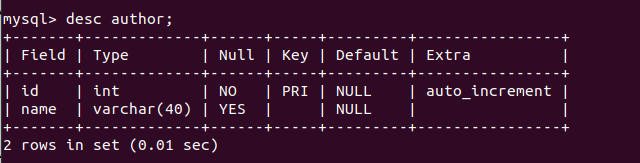
1. Create a class Address for Author with instance variables streetNumber, location, State.
2. Create instance variable of Address class inside Author class and save it as embedded object.
3. Introduce a List of subjects for author.
4. Persist 3 subjects for each author.
5. Create an Entity book with an instance variable bookName.
6. Implement One to One mapping between Author and Book.
7. Implement One to Many Mapping between Author and Book(Unidirectional, BiDirectional and without additional table ) and implement cascade save.
8. Implement Many to Many Mapping between Author and Book.
9. Which method on the session object can be used to remove an object from the cache?

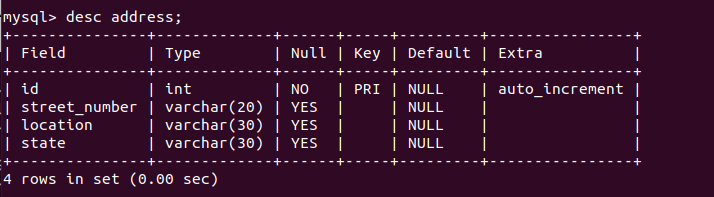
**Ans 9.** evict() method

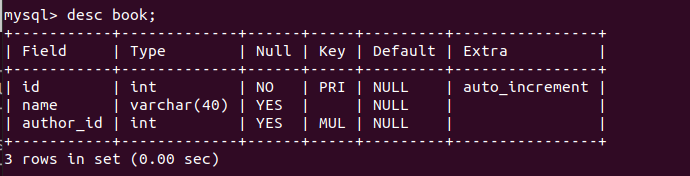
1. What does @transactional annotation do?

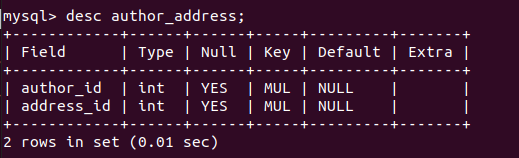
**Ans 10.** The main usage of Transactional annotation is when we are implementing fetch type as “FetchType.LAZY” (which is by default). Transactional at this point, attains a lock over the entity so that no other query related to that entity will be executed to avoid Dirty reads. Also, with the help of using Transactional annotation, if anyhow the querry/process is interrupted in midway (before the successful completion of operation), the changes are reverted back to the original.

**TABLE STRUCTURE:**









**OUTPUT AFTER IMPLEMENTATION OF THE TEST CASE:**

