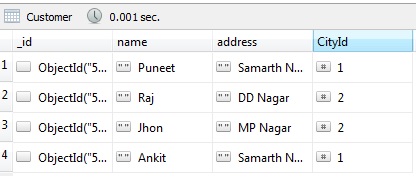
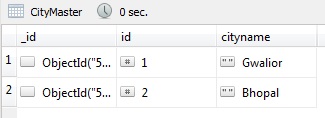
We already know how to use joins in SQL Server but MongoDB is document based and there is no syntax for joins in MongoDB. Most of the time we require a join between more than one table. However, it provides $lookup a aggregation function that works like join.  
  
**Below, I have created a demostration example.**We have two collections- Customer and Citymaster.

1. //Customer collection we insert some documents .
3. db.getCollection('Customer').find({})



1. //CityMaster  collection  insert some documents .
3. db.getCollection('CityMaster').find({})

     
  
Now, we use the $lookup aggregation function.

1. db.Customer.aggregate([
3. {$lookup: {from: "CityMaster", localField: "CityId", foreignField: "id", as: "CityDetail"}},
4. {$match: {details: {$ne: []}}}
5. ]);



Output:

1. {
2. "\_id" : ObjectId("598d4dd16abd19bbecbac78b"),
3. "name" : "Puneet",
4. "address" : "Samarth Nagar",
5. "CityId" : 1,
6. "CityDetail" : [
7. {
8. "\_id" : ObjectId("598d4d4f6abd19bbecbac779"),
9. "id" : 1,
10. "cityname" : "Gwalior"
11. }
12. ]
13. }
15. /\* 2 \*/
16. {
17. "\_id" : ObjectId("598d4dec6abd19bbecbac78d"),
18. "name" : "Raj",
19. "address" : "DD Nagar",
20. "CityId" : 2,
21. "CityDetail" : [
22. {
23. "\_id" : ObjectId("598d4d696abd19bbecbac77f"),
24. "id" : 2,
25. "cityname" : "Bhopal"
26. }
27. ]
28. }
30. /\* 3 \*/
31. {
32. "\_id" : ObjectId("598d4e0b6abd19bbecbac791"),
33. "name" : "Jhon",
34. "address" : "MP Nagar",
35. "CityId" : 2,
36. "CityDetail" : [
37. {
38. "\_id" : ObjectId("598d4d696abd19bbecbac77f"),
39. "id" : 2,
40. "cityname" : "Bhopal"
41. }
42. ]
43. }
45. /\* 4 \*/
46. {
47. "\_id" : ObjectId("598d4e216abd19bbecbac795"),
48. "name" : "Ankit",
49. "address" : "Samarth Nagar",
50. "CityId" : 1,
51. "CityDetail" : [
52. {
53. "\_id" : ObjectId("598d4d4f6abd19bbecbac779"),
54. "id" : 1,
55. "cityname" : "Gwalior"
56. }
57. ]
58. }

