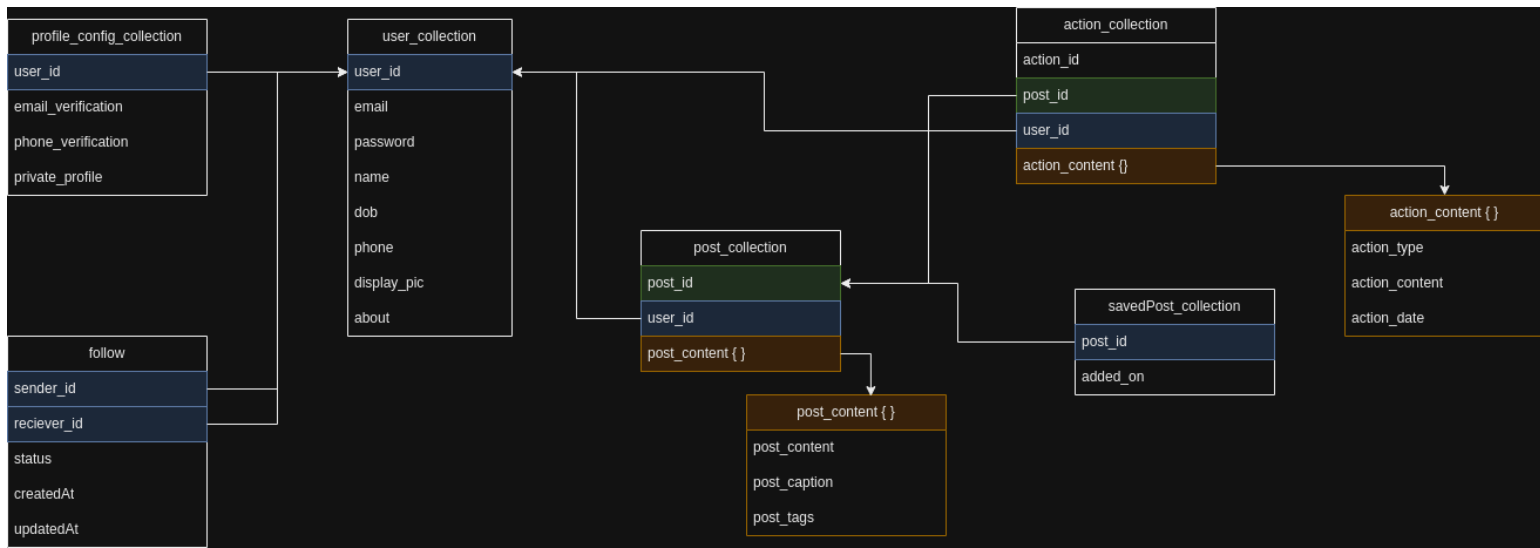


MongoDB Assigment

Schema for instagram :



Models in moongose:

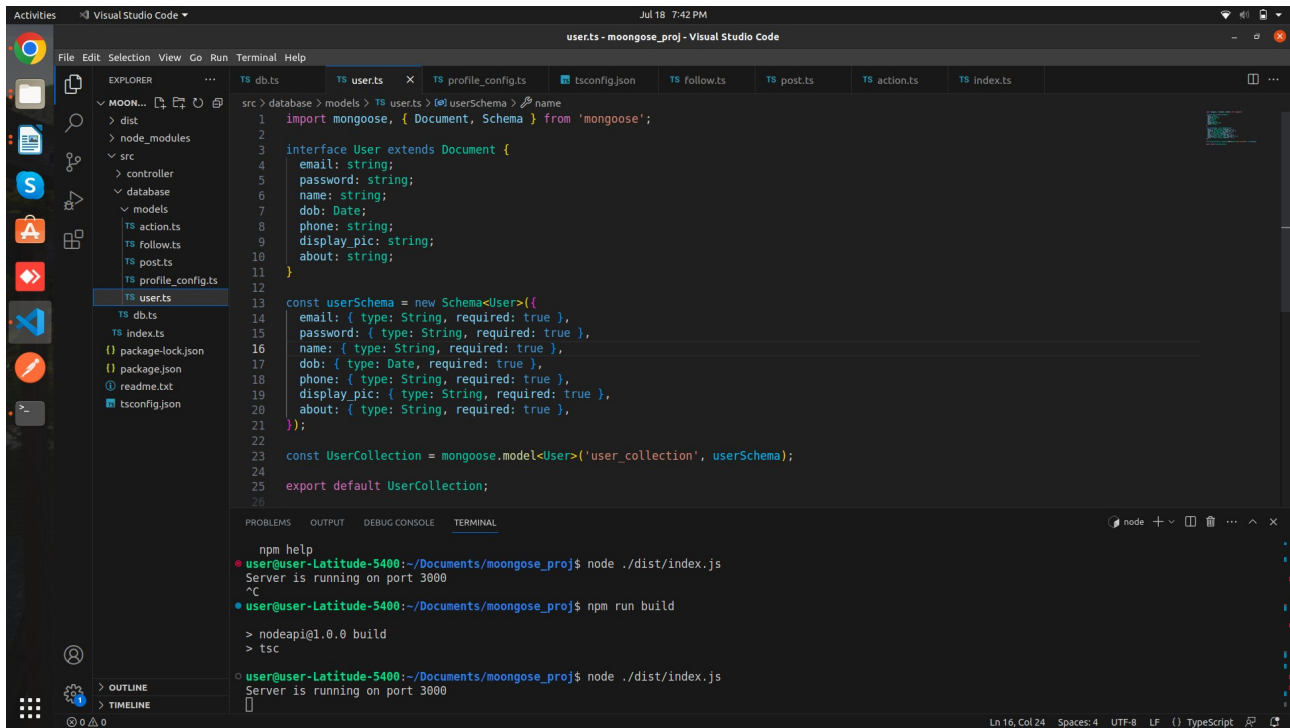
follow_model

```
1 import mongoose, { Document, Schema } from 'mongoose';
2
3 interface Follow extends Document {
4   sender_id: mongoose.Types.ObjectId;
5   receiver_id: mongoose.Types.ObjectId;
6   status: string;
7   createdAt: Date;
8   updatedAt: Date;
9 }
10
11 const followSchema = new Schema<Follow>({
12   sender_id: { type: Schema.Types.ObjectId, ref: 'user_collection' },
13   receiver_id: { type: Schema.Types.ObjectId, ref: 'user_collection' },
14   status: { type: String, enum: ['accepted', 'rejected', 'pending'], required: true },
15   createdAt: { type: Date, default: Date.now },
16   updatedAt: { type: Date, default: Date.now },
17 });
18
19 const FollowCollection = mongoose.model<Follow>('follow_collection', followSchema);
20
21 export default FollowCollection;
```

The terminal output shows the following commands and results:

```
npm help
user@user-Latitude-5400:~/Documents/moongose_proj$ node ./dist/index.js
Server is running on port 3000
^C
user@user-Latitude-5400:~/Documents/moongose_proj$ npm run build
> nodeapi@1.0.0 build
> tsc
user@user-Latitude-5400:~/Documents/moongose_proj$ node ./dist/index.js
Server is running on port 3000
```

user_model



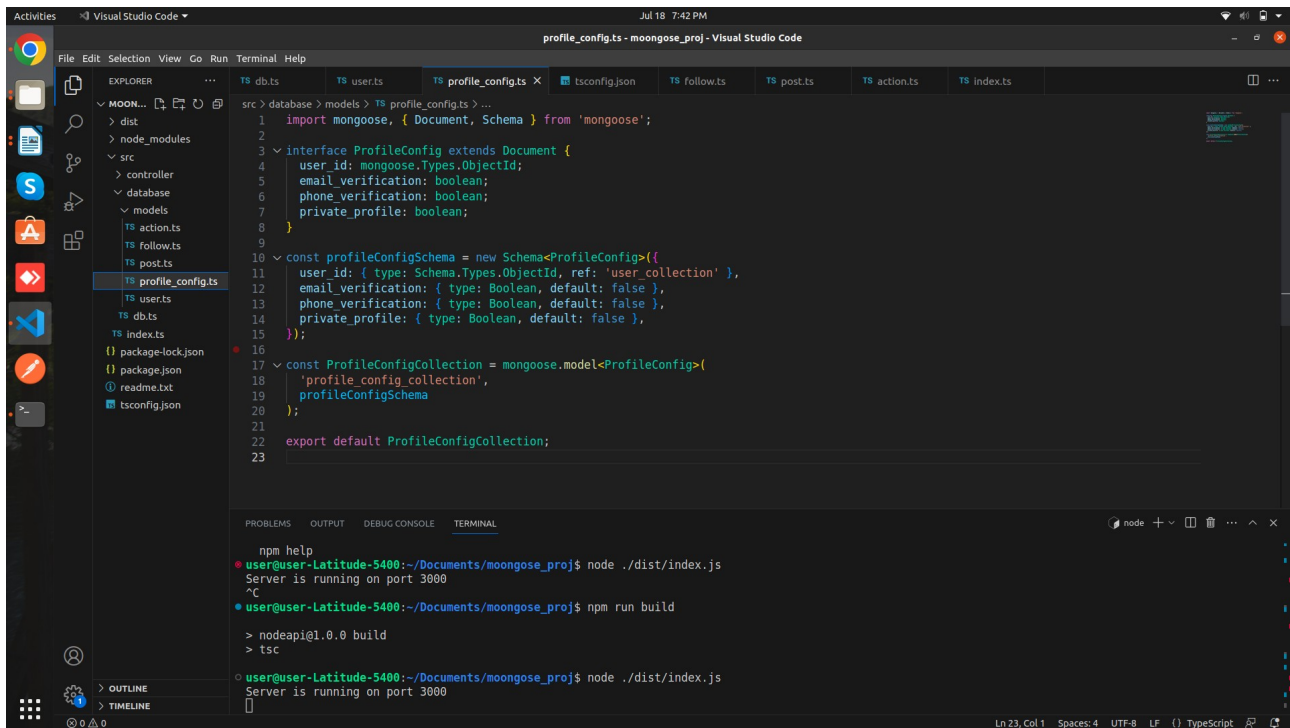
The screenshot shows the Visual Studio Code editor with the file `src > database > models > TS users > TS userschema > user_model.ts` open. The code defines a Mongoose schema for a user and a corresponding collection.

```
1 import mongoose, { Document, Schema } from 'mongoose';
2
3 interface User extends Document {
4   email: string;
5   password: string;
6   name: string;
7   dob: Date;
8   phone: string;
9   display_pic: string;
10  about: string;
11 }
12
13 const userSchema = new Schema<User>({
14   email: { type: String, required: true },
15   password: { type: String, required: true },
16   name: { type: String, required: true },
17   dob: { type: Date, required: true },
18   phone: { type: String, required: true },
19   display_pic: { type: String, required: true },
20   about: { type: String, required: true },
21 });
22
23 const UserCollection = mongoose.model<User>('user_collection', userSchema);
24
25 export default UserCollection;
```

The terminal at the bottom shows the following commands and output:

```
npm help
user@user-Latitude-5400:~/Documents/moongose_proj$ node ./dist/index.js
Server is running on port 3000
^C
user@user-Latitude-5400:~/Documents/moongose_proj$ npm run build
> nodeapi@1.0.0 build
> tsc
user@user-Latitude-5400:~/Documents/moongose_proj$ node ./dist/index.js
Server is running on port 3000
```

profile_config_model



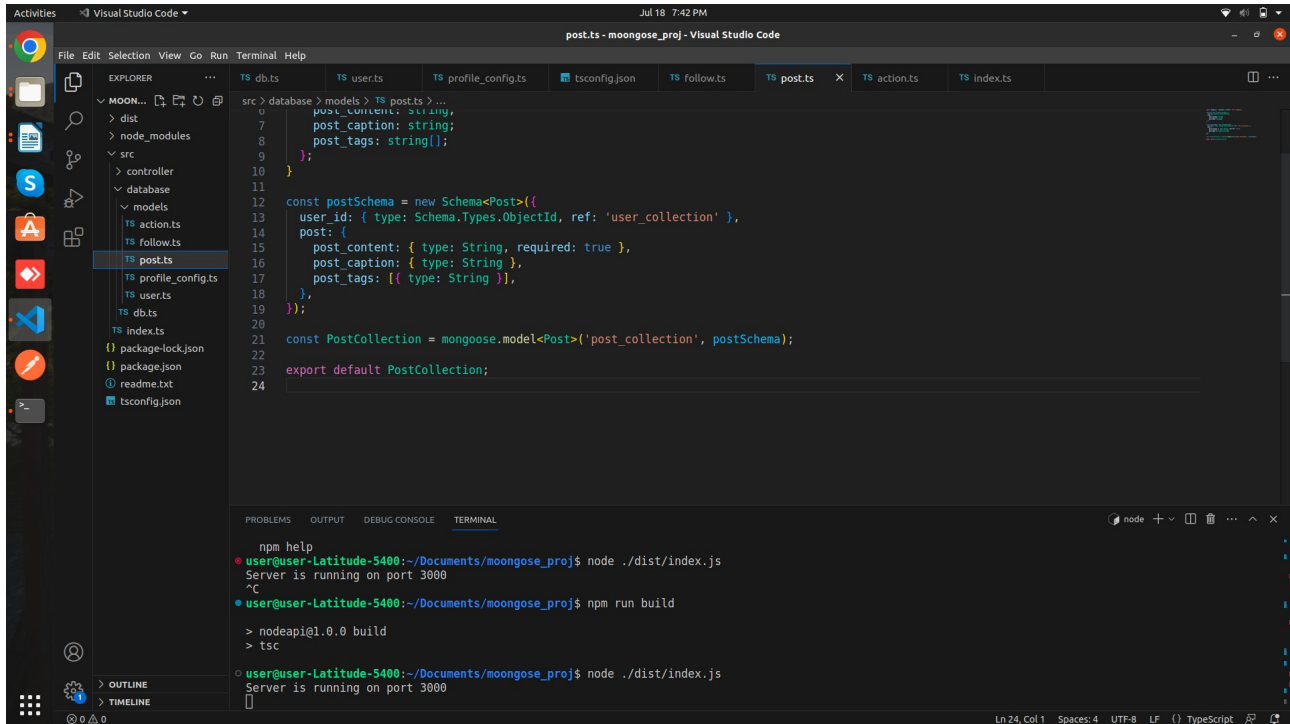
The screenshot shows the Visual Studio Code editor with the file `src > database > models > TS profile_configs > profile_config_model.ts` open. The code defines a Mongoose schema for a profile configuration and a corresponding collection.

```
1 import mongoose, { Document, Schema } from 'mongoose';
2
3 interface ProfileConfig extends Document {
4   user_id: mongoose.Types.ObjectId;
5   email_verification: boolean;
6   phone_verification: boolean;
7   private_profile: boolean;
8 }
9
10 const profileConfigSchema = new Schema<ProfileConfig>({
11   user_id: { type: Schema.Types.ObjectId, ref: 'user_collection' },
12   email_verification: { type: Boolean, default: false },
13   phone_verification: { type: Boolean, default: false },
14   private_profile: { type: Boolean, default: false },
15 });
16
17 const ProfileConfigCollection = mongoose.model<ProfileConfig>(
18   'profile_config_collection',
19   profileConfigSchema
20 );
21
22 export default ProfileConfigCollection;
```

The terminal at the bottom shows the following commands and output:

```
npm help
user@user-Latitude-5400:~/Documents/moongose_proj$ node ./dist/index.js
Server is running on port 3000
^C
user@user-Latitude-5400:~/Documents/moongose_proj$ npm run build
> nodeapi@1.0.0 build
> tsc
user@user-Latitude-5400:~/Documents/moongose_proj$ node ./dist/index.js
Server is running on port 3000
```

post_model



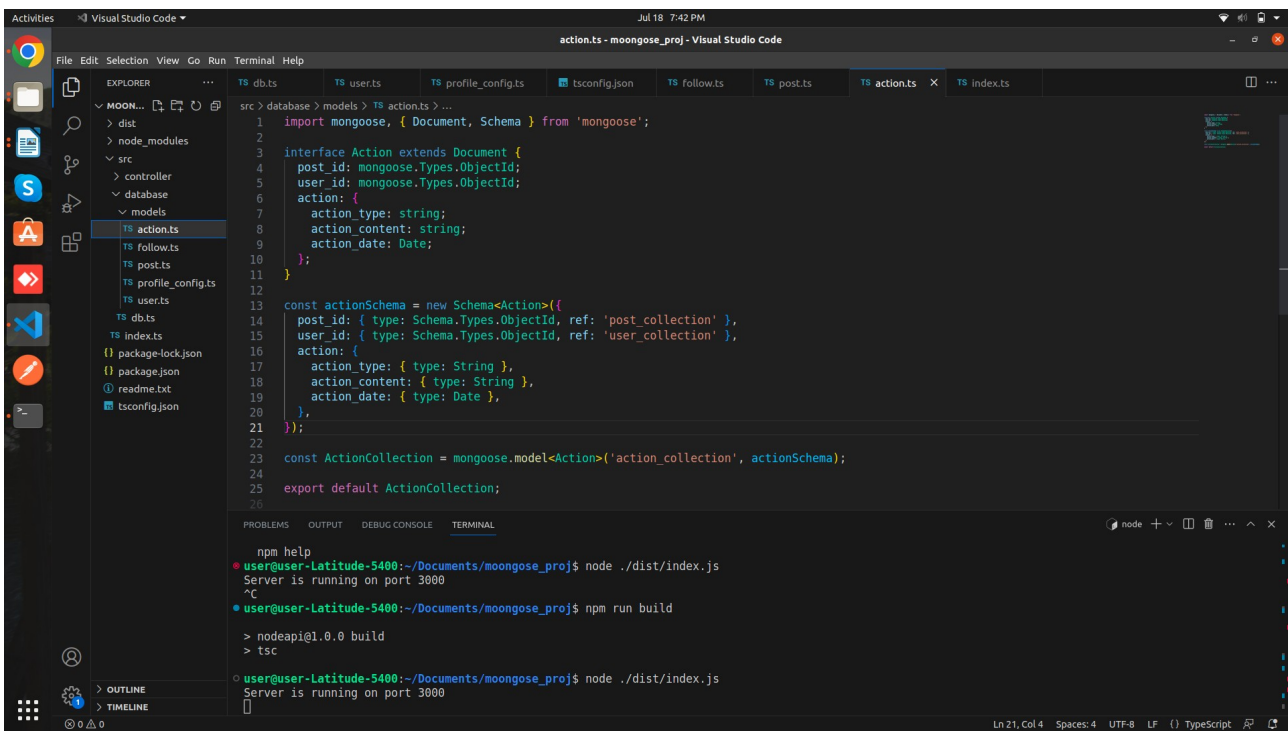
The screenshot shows the Visual Studio Code editor with the file `post_model.ts` open. The file contains the following TypeScript code:

```
src > database > models > TS posts > ...
6
7   post_content: string;
8   post_caption: string;
9   post_tags: string[];
10 };
11
12 const postSchema = new Schema<Post>({
13   user_id: { type: Schema.Types.ObjectId, ref: 'user_collection' },
14   post: {
15     post_content: { type: String, required: true },
16     post_caption: { type: String },
17     post_tags: [{ type: String }],
18   },
19 });
20
21 const PostCollection = mongoose.model<Post>('post_collection', postSchema);
22
23 export default PostCollection;
24
```

The terminal at the bottom shows the following commands and output:

```
npm help
user@user-Latitude-5400:~/Documents/moongose_proj$ node ./dist/index.js
Server is running on port 3000
^C
user@user-Latitude-5400:~/Documents/moongose_proj$ npm run build
> nodeapi@1.0.0 build
> tsc
user@user-Latitude-5400:~/Documents/moongose_proj$ node ./dist/index.js
Server is running on port 3000
```

action_model



The screenshot shows the Visual Studio Code editor with the file `action_model.ts` open. The file contains the following TypeScript code:

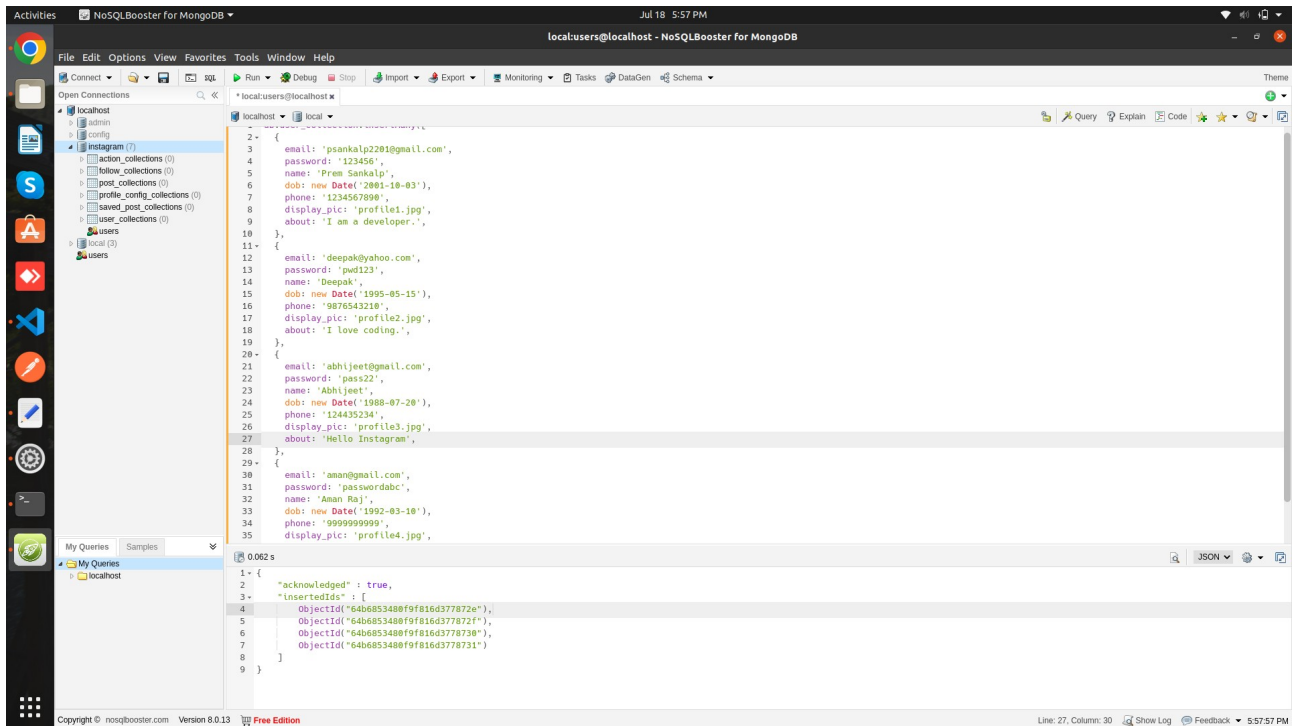
```
src > database > models > TS actions > ...
1 import mongoose, { Document, Schema } from 'mongoose';
2
3 interface Action extends Document {
4   post_id: mongoose.Types.ObjectId;
5   user_id: mongoose.Types.ObjectId;
6   action: {
7     action_type: string;
8     action_content: string;
9     action_date: Date;
10   };
11 };
12
13 const actionSchema = new Schema<Action>({
14   post_id: { type: Schema.Types.ObjectId, ref: 'post_collection' },
15   user_id: { type: Schema.Types.ObjectId, ref: 'user_collection' },
16   action: {
17     action_type: { type: String },
18     action_content: { type: String },
19     action_date: { type: Date },
20   },
21 });
22
23 const ActionCollection = mongoose.model<Action>('action_collection', actionSchema);
24
25 export default ActionCollection;
26
```

The terminal at the bottom shows the following commands and output:

```
npm help
user@user-Latitude-5400:~/Documents/moongose_proj$ node ./dist/index.js
Server is running on port 3000
^C
user@user-Latitude-5400:~/Documents/moongose_proj$ npm run build
> nodeapi@1.0.0 build
> tsc
user@user-Latitude-5400:~/Documents/moongose_proj$ node ./dist/index.js
Server is running on port 3000
```

Queries in nosqlbooster

Insertion on user



The screenshot shows the NoSQLBooster for MongoDB interface. On the left, the 'Open Connections' pane shows a connection to 'localhost' with a database named 'instagram'. The 'collections' pane shows several collections, including 'users'. The main editor displays a JSON array of user documents. The bottom pane shows the result of an insert operation, indicating that 5 documents were successfully inserted into the 'users' collection.

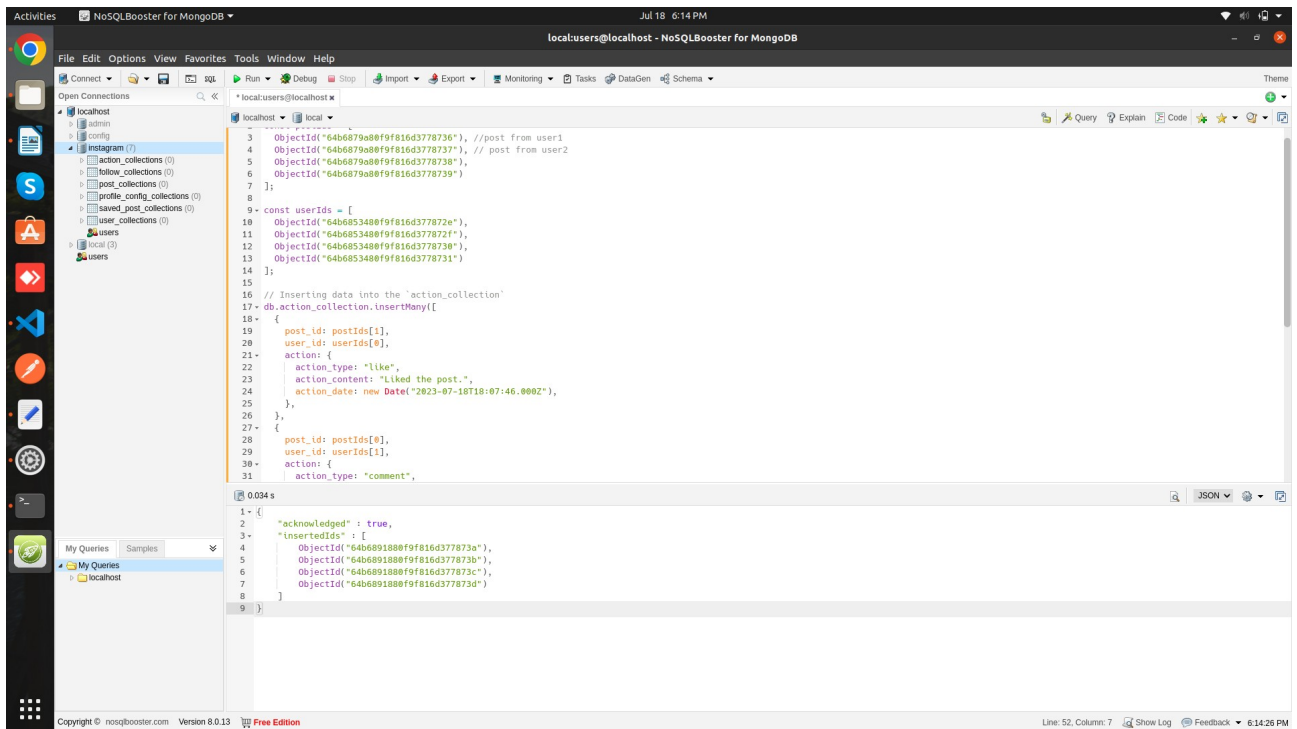
```
1 {
2   "email": "psankalp2201@gmail.com",
3   "password": "123456",
4   "name": "Prem Sankalp",
5   "dob": new Date("2001-10-03"),
6   "phone": "1234567890",
7   "display_pic": "profile1.jpg",
8   "about": "I am a developer.",
9 },
10 {
11   "email": "deepak@yahoo.com",
12   "password": "pud123",
13   "name": "Deepak",
14   "dob": new Date("1995-05-15"),
15   "phone": "9876543210",
16   "display_pic": "profile2.jpg",
17   "about": "I love coding.",
18 },
19 {
20   "email": "abhijeet@gmail.com",
21   "password": "pass22",
22   "name": "Abhijeet",
23   "dob": new Date("1988-07-20"),
24   "phone": "124435234",
25   "display_pic": "profile3.jpg",
26   "about": "Hello Instagram",
27 },
28 {
29   "email": "aman@gmail.com",
30   "password": "passwordabc",
31   "name": "Aman Raj",
32   "dob": new Date("1992-03-10"),
33   "phone": "999999999",
34   "display_pic": "profile4.jpg",
35 }

```

```
1 {
2   "acknowledged": true,
3   "insertedIds": [
4     ObjectId("64b6853480f9f816d377872e"),
5     ObjectId("64b6853480f9f816d377872f"),
6     ObjectId("64b6853480f9f816d3778730"),
7     ObjectId("64b6853480f9f816d3778731")
8   ]
9 }

```

insertion on post_action



The screenshot shows the NoSQLBooster for MongoDB interface. On the left, the 'Open Connections' pane shows a connection to 'localhost' with a database named 'instagram'. The 'collections' pane shows several collections, including 'action_collection'. The main editor displays a JavaScript script that inserts data into the 'action_collection'. The bottom pane shows the result of the insert operation, indicating that 2 documents were successfully inserted into the 'action_collection'.

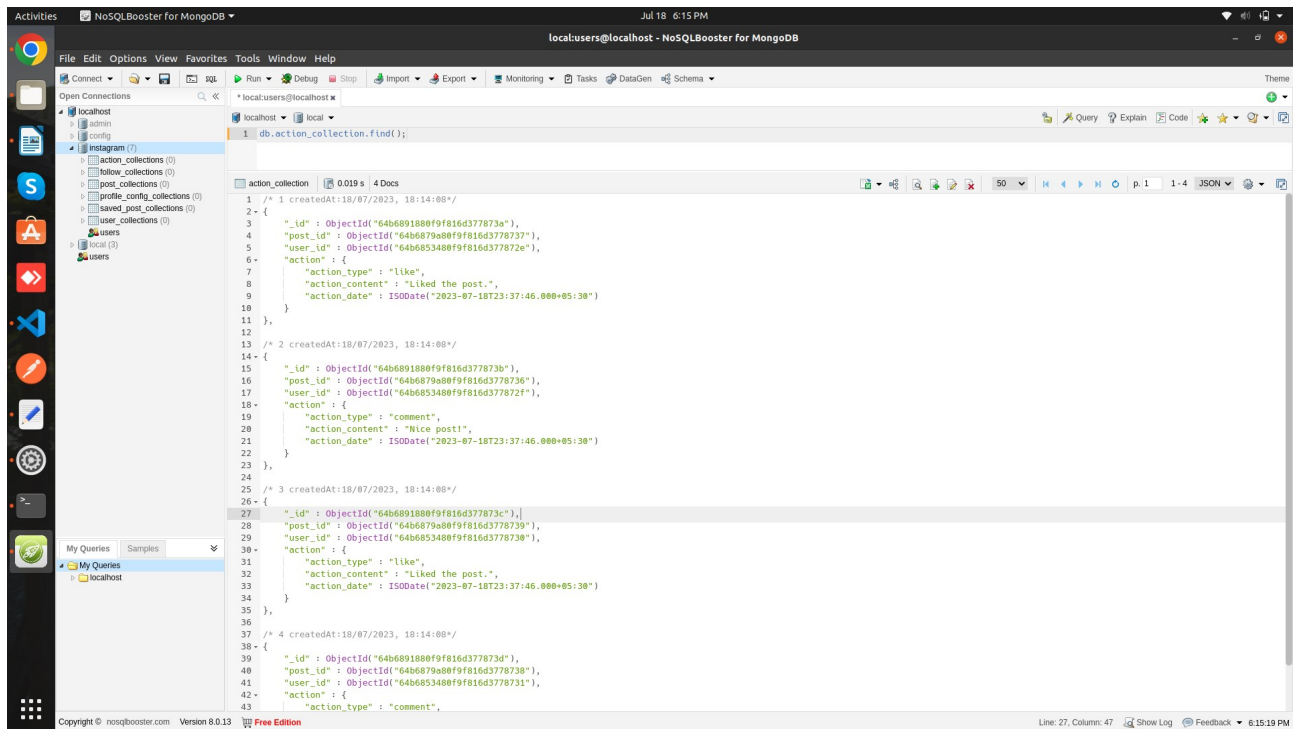
```
1 // Inserting data into the 'action_collection'
2 db.action_collection.insertMany([
3   {
4     post_id: postIds[0],
5     user_id: userIds[0],
6     action: {
7       action_type: "like",
8       action_content: "Liked the post.",
9       action_date: new Date("2023-07-18T18:07:46.000Z"),
10     },
11   },
12   {
13     post_id: postIds[1],
14     user_id: userIds[1],
15     action: {
16       action_type: "comment",
17     },
18   },
19 ])

```

```
1 {
2   "acknowledged": true,
3   "insertedIds": [
4     ObjectId("64b6891880f9f816d377873a"),
5     ObjectId("64b6891880f9f816d377873b"),
6     ObjectId("64b6891880f9f816d377873c")
7   ]
8 }

```

result of insertion on post_action

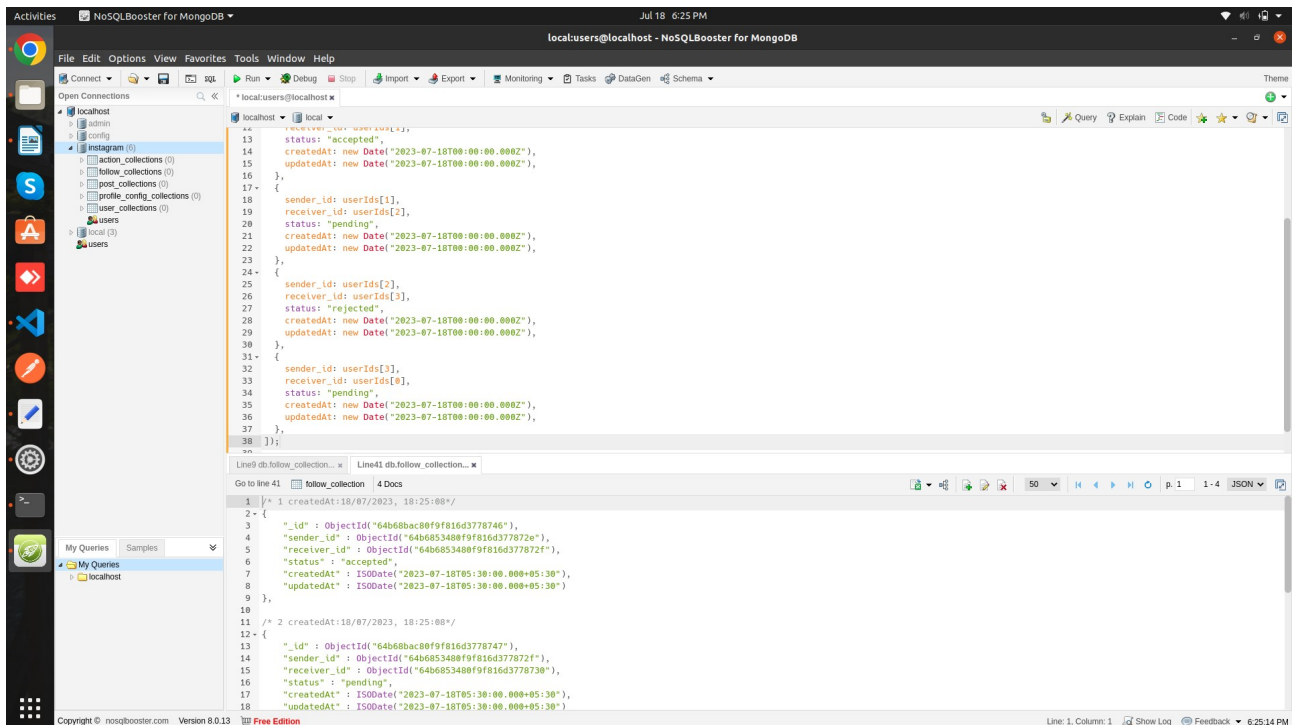


The screenshot shows the NoSQLBooster for MongoDB interface. The left sidebar displays the database structure with collections like 'action_collections', 'follow_collections', 'post_collections', 'profile_config_collections', 'saved_post_collections', and 'user_collections'. The main window shows a query result for 'ob.action_collection.find()'. The result is a JSON array of four documents, each representing an action. The first document is a 'like' action, and the others are 'comment' actions. The status of the actions is 'accepted'.

```
1 ob.action_collection.find();
```

```
{
  "id": "ObjectId('64b6891880f9f816d377873c')",
  "post_id": "ObjectId('64b6879a80f9f816d3778739')",
  "user_id": "ObjectId('64b6853480f9f816d377872e')",
  "action": {
    "action_type": "like",
    "action_content": "Liked the post.",
    "action_date": "ISODate('2023-07-18T23:37:46.000+05:30')",
    "status": "accepted"
  }
},
{
  "id": "ObjectId('64b6891880f9f816d377873d')",
  "post_id": "ObjectId('64b6879a80f9f816d3778739')",
  "user_id": "ObjectId('64b6853480f9f816d377872f')",
  "action": {
    "action_type": "comment",
    "action_content": "Nice post!",
    "action_date": "ISODate('2023-07-18T23:37:46.000+05:30')",
    "status": "accepted"
  }
},
{
  "id": "ObjectId('64b6891880f9f816d377873e')",
  "post_id": "ObjectId('64b6879a80f9f816d3778739')",
  "user_id": "ObjectId('64b6853480f9f816d377872e')",
  "action": {
    "action_type": "like",
    "action_content": "Liked the post.",
    "action_date": "ISODate('2023-07-18T23:37:46.000+05:30')",
    "status": "accepted"
  }
},
{
  "id": "ObjectId('64b6891880f9f816d377873f')",
  "post_id": "ObjectId('64b6879a80f9f816d3778739')",
  "user_id": "ObjectId('64b6853480f9f816d377872f')",
  "action": {
    "action_type": "comment",
    "action_content": "Nice post!",
    "action_date": "ISODate('2023-07-18T23:37:46.000+05:30')",
    "status": "accepted"
  }
}
```

insertion on follow



The screenshot shows the NoSQLBooster for MongoDB interface. The left sidebar displays the database structure with collections like 'action_collections', 'follow_collections', 'post_collections', 'profile_config_collections', 'saved_post_collections', and 'user_collections'. The main window shows a query result for 'db.follow_collection.find()'. The result is a JSON array of four documents, each representing a follow action. The first document is a 'pending' action, and the others are 'accepted' actions. The status of the actions is 'accepted'.

```
1 db.follow_collection.find();
```

```
{
  "sender_id": "ObjectId('64b688ac80f9f816d3778746')",
  "receiver_id": "ObjectId('64b6853480f9f816d377872e')",
  "status": "pending",
  "created_at": "ISODate('2023-07-18T08:00:00.000+05:30')",
  "updated_at": "ISODate('2023-07-18T08:00:00.000+05:30')",
  "status": "pending"
},
{
  "sender_id": "ObjectId('64b688ac80f9f816d3778747')",
  "receiver_id": "ObjectId('64b6853480f9f816d377872f')",
  "status": "accepted",
  "created_at": "ISODate('2023-07-18T08:00:00.000+05:30')",
  "updated_at": "ISODate('2023-07-18T08:00:00.000+05:30')",
  "status": "accepted"
},
{
  "sender_id": "ObjectId('64b688ac80f9f816d3778748')",
  "receiver_id": "ObjectId('64b6853480f9f816d3778730')",
  "status": "pending",
  "created_at": "ISODate('2023-07-18T08:00:00.000+05:30')",
  "updated_at": "ISODate('2023-07-18T08:00:00.000+05:30')",
  "status": "pending"
},
{
  "sender_id": "ObjectId('64b688ac80f9f816d3778749')",
  "receiver_id": "ObjectId('64b6853480f9f816d3778731')",
  "status": "accepted",
  "created_at": "ISODate('2023-07-18T08:00:00.000+05:30')",
  "updated_at": "ISODate('2023-07-18T08:00:00.000+05:30')",
  "status": "accepted"
}
```

Lookup queries :

Activities NoSQLBooster for MongoDB Jul 18 8:03 PM

local:action_collection@localhost - NoSQLBooster for MongoDB

File Edit Options View Favorites Tools Window Help

Connect Import Export Monitoring Tasks DataGen Schema

Open Connections

- localhost
 - admin
 - config
 - instagram (6)
 - action_collections (0)
 - follow_collections (0)
 - post_collections (0)
 - profile_config_collections (0)
 - user_collections (0)
 - users
- local
 - users

```
1- db.post_collection.aggregate([
2- {
3-   $lookup: {
4-     from: "user_collection",
5-     localField: "user_id",
6-     foreignField: "user_id",
7-     as: "user_details"
8-   }
9- }
10- ]);
11-
```

post_collection 0.034 s | 4 Docs

```
1- /* 1 createdAt:18/07/2023, 18:07:46 */
2- {
3-   "_id" : ObjectId("64b6879a80f9f816d3778736"),
4-   "user_id" : ObjectId("64b6853480f9f816d377872e"),
5-   "post" : {
6-     "post_content" : "image1.jpeg",
7-     "post_caption" : "First post!",
8-     "post_tags" : [ "#first_post", "#instagram" ]
9-   },
10-   "user_details" : [ ]
11- },
12-
13- /* 2 createdAt:18/07/2023, 18:07:46 */
14- {
15-   "_id" : ObjectId("64b6879a80f9f816d3778737"),
16-   "user_id" : ObjectId("64b6853480f9f816d377872f"),
17-   "post" : {
18-     "post_content" : "image2.jpeg",
19-     "post_caption" : "Hi i am on Instagram",
20-     "post_tags" : [ "#photo", "instagram" ]
21-   },
22-   "user_details" : [ ]
23- },
24-
```

My Queries Samples

My Queries

localhost

Copyright © noSQLbooster.com Version 8.0.13 Free Edition Line: 1, Column: 1 Show Log Feedback 8:03:14 PM

Activities NoSQLBooster for MongoDB Jul 18 8:02 PM

local:action_collection@localhost - NoSQLBooster for MongoDB

File Edit Options View Favorites Tools Window Help

Connect Import Export Monitoring Tasks DataGen Schema

Open Connections

- localhost
 - admin
 - config
 - instagram (6)
 - action_collections (0)
 - follow_collections (0)
 - post_collections (0)
 - profile_config_collections (0)
 - user_collections (0)
 - users
- local
 - users

```
1- db.follow_collection.aggregate([
2- {
3-   $lookup: {
4-     from: "user_collection",
5-     localField: "sender_id",
6-     foreignField: "user_id",
7-     as: "sender_details"
8-   },
9-   {
10-    $lookup: {
11-      from: "user_collection",
12-      localField: "receiver_id",
13-      foreignField: "user_id",
14-      as: "receiver_details"
15-    }
16-  }
17- }
18- ]);
19-
```

follow_collection 0.056 s | 4 Docs

Key	Value	Type
(1) 64b68bac09f9f816d3778746	{ status: "accepted" } (8 fields)	Document
_id	64b68bac09f9f816d3778746	ObjectId
sender_id	64b6853480f9f816d377872e	ObjectId
receiver_id	64b6853480f9f816d377872f	ObjectId
status	accepted	String
createdAt	18/07/2023, 05:30:00 - 15 hours ago	Date
updatedAt	18/07/2023, 05:30:00 - 15 hours ago	Date
sender_details	Array[]	Array
receiver_details	Array[]	Array
(2) 64b68bac09f9f816d3778747	{ status: "pending" } (8 fields)	Document
(3) 64b68bac09f9f816d3778748	{ status: "rejected" } (8 fields)	Document
(4) 64b68bac09f9f816d3778749	{ status: "pending" } (8 fields)	Document

My Queries Samples

My Queries

localhost

Copyright © noSQLbooster.com Version 8.0.13 Free Edition Line: 16, Column: 6 Show Log Feedback 8:02:51 PM

Activities NoSQLBooster for MongoDB Jul 18 8:11 PM

local:action_collection@localhost - NoSQLBooster for MongoDB

File Edit Options View Favorites Tools Window Help

Connect Open Connections

localhost

- admin
- config
- instagram (6)
 - action_collections (0)
 - follow_collections (0)
 - post_collections (0)
 - profile_config_collections (0)
 - user_collections (0)
- local
- users

```
1- db.post_collection.aggregate([
2- {
3-   $lookup: {
4-     from: "user_collection",
5-     let: { user_id: "$user_id" },
6-     pipeline: [
7-       { $match: { $expr: { $eq: [ "$user_id", "$$userid" ] } } },
8-       { $project: { name: 1, email: 1 } } }
9-     ],
10-    as: "user_details"
11-  }
12- }
13- ]]);
14
```

post_collection 0.033 s 4 Docs

Key	Value	Type
(1) 64b6879a809b16d3778736	{}	Document
id	64b6879a809b16d3778736	ObjectId
user_id	64b68534809b16d377872e	ObjectId
post	{ post_content: "image1.jpeg", post_caption: "First post", post_tags: ["first_post", "instagram"] }	Object
post_content	image1.jpeg	String
post_caption	First post	String
post_tags	Array[1]	Array
user_details	Array[1]	Array
(2) 64b6879a809b16d3778737	{}	Document
(3) 64b6879a809b16d3778738	{}	Document
(4) 64b6879a809b16d3778739	{}	Document

My Queries Samples

My Queries

- localhost

Copyright © nosqlbooster.com Version 8.0.13 Free Edition

Line: 14, Column: 1 Show Log Feedback 8:11:19 PM

Activities NoSQLBooster for MongoDB Jul 18 8:12 PM

local:action_collection@localhost - NoSQLBooster for MongoDB

File Edit Options View Favorites Tools Window Help

Connect Open Connections

localhost

- admin
- config
- instagram (6)
 - action_collections (0)
 - follow_collections (0)
 - post_collections (0)
 - profile_config_collections (0)
 - user_collections (0)
- local
- users

```
1- db.follow_collection.aggregate([
2- {
3-   $lookup: {
4-     from: "user_collection",
5-     localField: "sender_id",
6-     foreignField: "user_id",
7-     as: "sender_details"
8-   }
9- },
10- {
11-   $lookup: {
12-     from: "user_collection",
13-     localField: "receiver_id",
14-     foreignField: "user_id",
15-     as: "receiver_details"
16-   }
17- },
18- {
19-   $project: {
20-     sender_id: 1,
21-     receiver_id: 1,
22-     sender_name: { $arrayElemAt: ["$sender_details.name", 0] },
23-     receiver_name: { $arrayElemAt: ["$receiver_details.name", 0] }
24-   }
25- }
26- ]]);
27
```

follow_collection 0.030 s 4 Docs

Key	Value	Type
(1) 64b68bac809b16d3778746	{ sender_id: ObjectId("64b68534809b16d377872e"), receiver_id: ObjectId("64b68534809b16d377872f") }	Document
id	64b68bac809b16d3778746	ObjectId
sender_id	64b68534809b16d377872e	ObjectId
receiver_id	64b68534809b16d377872f	ObjectId
(2) 64b68bac809b16d3778747	{ sender_id: ObjectId("64b68534809b16d377872f"), receiver_id: ObjectId("64b68534809b16d3778730") }	Document
(3) 64b68bac809b16d3778748	{ sender_id: ObjectId("64b68534809b16d3778730"), receiver_id: ObjectId("64b68534809b16d3778731") }	Document
(4) 64b68bac809b16d3778749	{ sender_id: ObjectId("64b68534809b16d3778731"), receiver_id: ObjectId("64b68534809b16d377872e") }	Document

My Queries Samples

My Queries

- localhost

Copyright © nosqlbooster.com Version 8.0.13 Free Edition

Line: 19, Column: 16 Show Log Feedback 8:12:29 PM

Sort queries

The screenshot shows the NoSQLBooster for MongoDB interface. The query editor contains the following query:

```
1 db.user_collection.find().sort({ name: 1 });
```

The results pane shows 4 documents from the 'user_collection'. The documents are sorted by the 'name' field in ascending order.

Key	Value	Type
(1) 64b68534809b16d3778730	{ email: "abhiject@gmail.com", name: "Abhiject" }	Document
	64b68534809b16d3778730	ObjectId
	abhiject@gmail.com	String
	passw0rd	String
	Abhiject	String
	20/07/1988, 05:30:00 - 35 years ago	Date
	124435234	String
	profile3.jpg	String
	Hello Instagram	String
(2) 64b68534809b16d3778731	{ email: "aman@gmail.com", name: "Aman Raj" }	Document
(3) 64b68534809b16d3778722	{ email: "deepak@yahoo.com", name: "Deepak" }	Document
(4) 64b68534809b16d377872e	{ email: "psankalp2201@gmail.com", name: "Prem Sankalp" }	Document

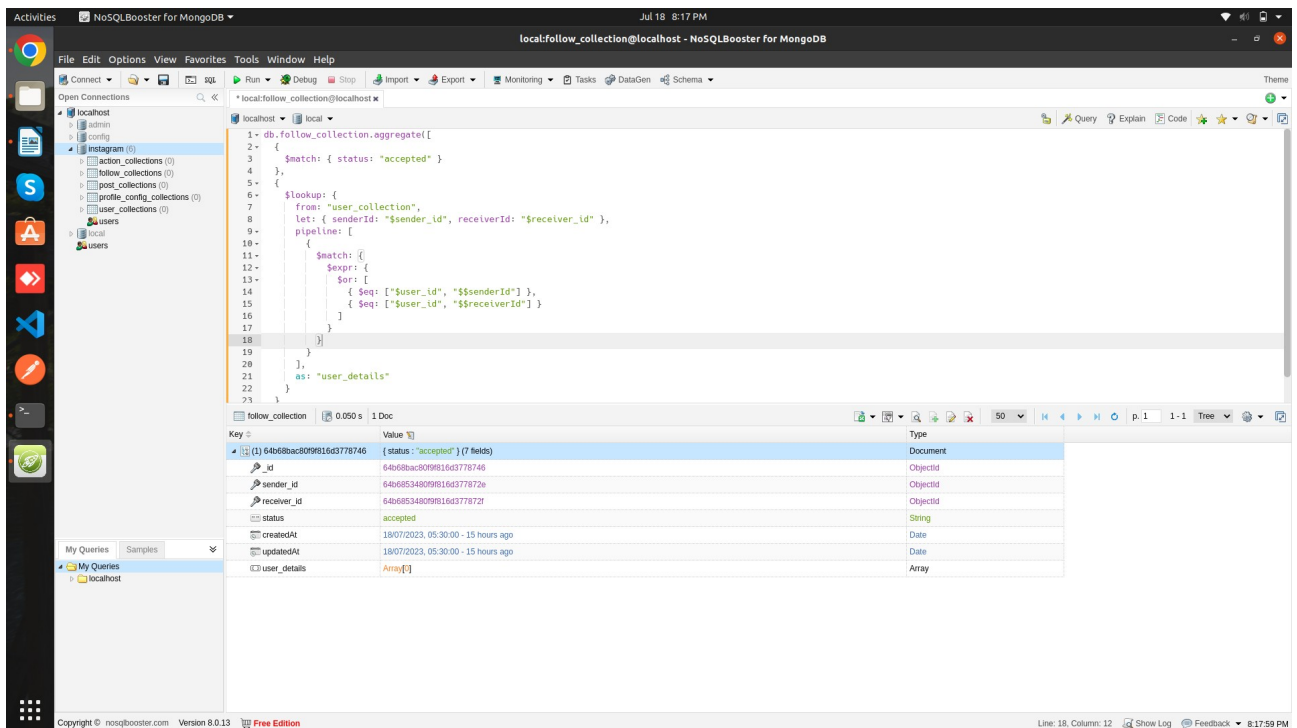
The screenshot shows the NoSQLBooster for MongoDB interface. The query editor contains the following query:

```
1 db.post_collection.find().sort({ createdAt: -1 });
```

The results pane shows 4 documents from the 'post_collection'. The documents are sorted by the 'createdAt' field in descending order.

Key	Value	Type
(1) 64b6879a809b16d3778736	{ }	Document
	64b6879a809b16d3778736	ObjectId
	64b68534809b16d377872e	ObjectId
	{ post_content: "image1.jpeg", post_caption: "First post", post_tags: ["first_post", "instagram"] }	Object
	image1.jpeg	String
	First post	String
	Array[]	Array
(2) 64b6879a809b16d3778737	{ }	Document
(3) 64b6879a809b16d3778738	{ }	Document
(4) 64b6879a809b16d3778739	{ }	Document

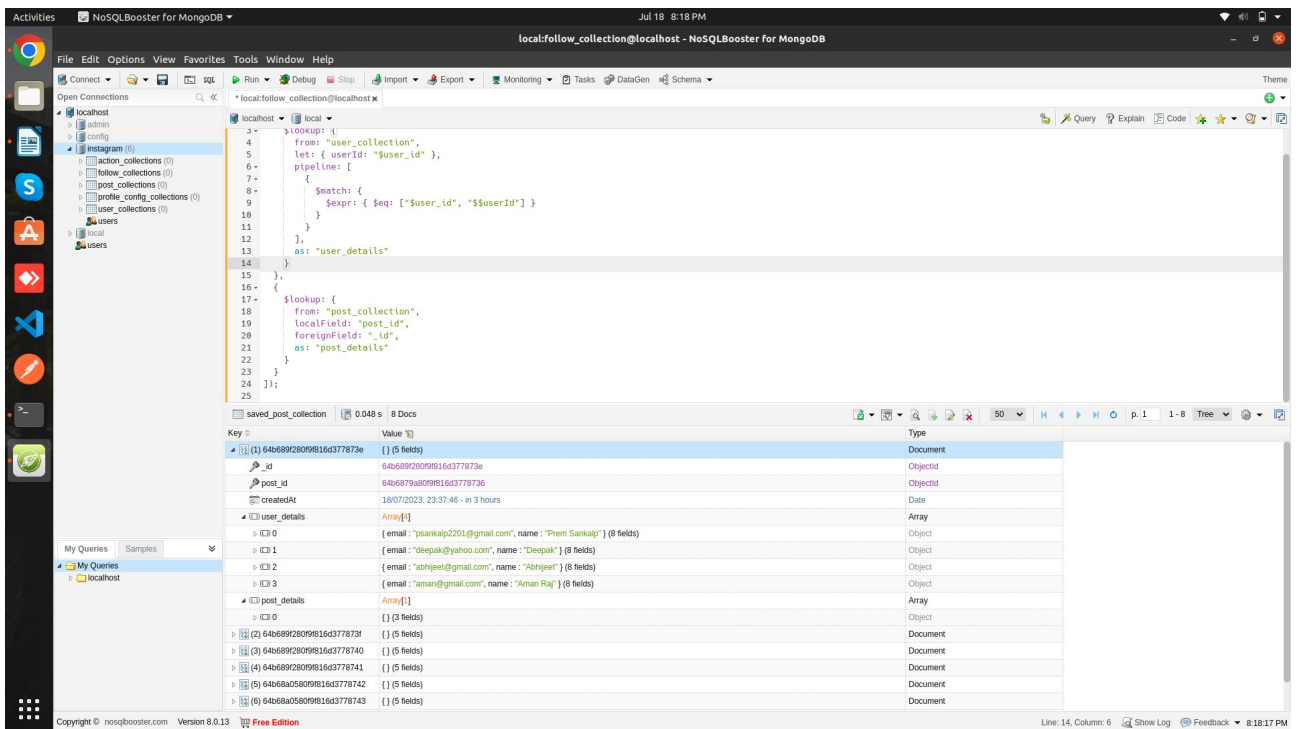
Pipeline Lookup based queries :



local:follow_collection@localhost - NoSQLBooster for MongoDB

```
1- db.follow_collection.aggregate([
2-   {
3-     $match: { status: "accepted" }
4-   },
5-   {
6-     $lookup: {
7-       from: "user_collection",
8-       let: { senderId: "$sender_id", receiverId: "$receiver_id" },
9-       pipeline: [
10-        {
11-          $match: {
12-            $expr: {
13-              $or: [
14-                { $eq: ["$user_id", "$$senderId"] },
15-                { $eq: ["$user_id", "$$receiverId"] }
16-              ]
17-            }
18-          }
19-        }
20-      ],
21-      as: "user_details"
22-    }
23-  ]
24- })
```

Key	Value	Type
status	accepted	String
createdAt	18/07/2023, 05:30:00 - 15 hours ago	Date
updatedAt	18/07/2023, 05:30:00 - 15 hours ago	Date
user_details	Array[1]	Array



local:follow_collection@localhost - NoSQLBooster for MongoDB

```
1- db.follow_collection.aggregate([
2-   {
3-     $match: { status: "accepted" }
4-   },
5-   {
6-     $lookup: {
7-       from: "user_collection",
8-       let: { userId: "$sender_id" },
9-       pipeline: [
10-        {
11-          $match: {
12-            $expr: { $eq: ["$user_id", "$$userId"] }
13-          }
14-        }
15-      ],
16-      as: "user_details"
17-    }
18-  },
19-   {
20-     $lookup: {
21-       from: "post_collection",
22-       localField: "post_id",
23-       foreignField: "_id",
24-       as: "post_details"
25-     }
26-   }
27- ])
```

Key	Value	Type
status	accepted	String
createdAt	18/07/2023, 23:37:46 - in 3 hours	Date
user_details	Array[4]	Array
post_details	Array[1]	Array