Add MongoDB Array using insert () with Example

The "insert" command can also be used to insert multiple documents into a collection at one time. The below code example can be used to insert multiple documents at a time.

The following example shows how this can be done,

1. Create a JavaScript variable called myEmployee to hold the array of documents
2. Add the required documents with the Field Name and values to the variable
3. Use the insert command to insert the array of documents into the collection

```
var myEmployee=
        [

                {
                        "Employeeid" : 1,
                        "EmployeeName" : "Smith"
                },
                {
                        "Employeeid"   : 2,
                        "EmployeeName" : "Mohan"
                },
                {
                        "Employeeid"   : 3,
                        "EmployeeName" : "Joe"
                },

        ];

        db.Employee.insert(myEmployee);
```

Printing in JSON format

JSON is a format called **JavaScript Object Notation**, and is just a way to store information in an organized, easy-to-read manner. In our further examples, we are going to use the JSON print functionality to see the output in a better format.

Let's look at an example of printing in JSON format

```
db.Employee.find().forEach(printjson)
```

Mongodb Primary Key: Example to set _id field with ObjectId()

What is Primary Key in MongoDB?

In MongoDB, _id field as the primary key for the collection so that each document can be uniquely identified in the collection. The _id field contains a unique ObjectID value.
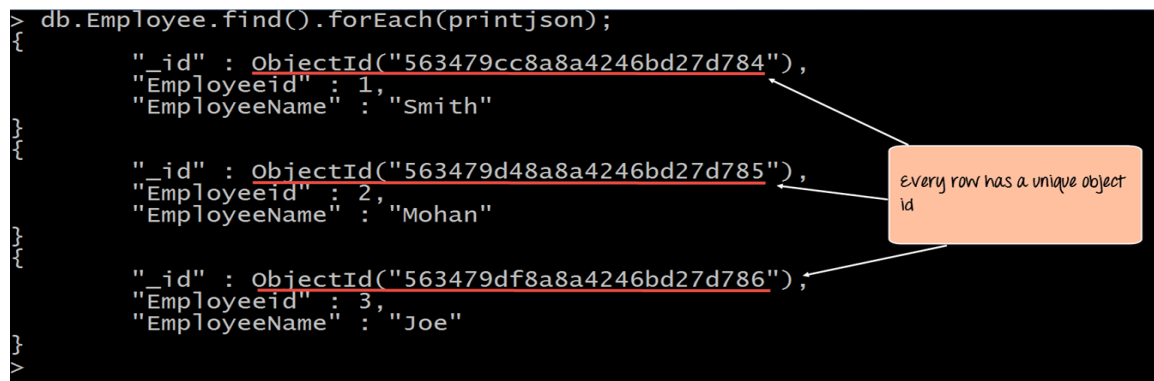
By default when inserting documents in the collection, if you don't add a field name with the _id in the field name, then MongoDB will automatically add an Object id field as shown below

When you query the documents in a collection, you can see the ObjectId for each document in the collection.

If you want to ensure that MongoDB does not create the _id Field when the collection is created and if you want to specify your own id as the _id of the collection, then you need to explicitly define this while creating the collection.

When explicitly creating an id field, it needs to be created with _id in its name.

Let's look at an example on how we can achieve this.

```
db.Employee.insert({_id:10, "EmployeeName" : "Smith"})
```