

Mongo db

1. Write a query to insert batch details in batch collection.

```
db.batch.insertMany([
...   {
...     "batch": "b1",
...     "students": [
...       {
...         "name": "Adams",
...         "attendance": [
...           {
...             "date": "2020-01-20",
...             "isPresent": true
...           },
...           {
...             "date": "2020-01-21",
...             "isPresent": false
...           },
...           {
...             "date": "2020-01-22",
...             "isPresent": true
...           }
...         ]
...       },
...       {
...         "name": "Baker",
...         "attendance": [
...           {
...             "date": "2020-01-20",
...             "isPresent": true
...           },
...           {
...             "date": "2020-01-21",
...             "isPresent": true
...           },
...           {
...             "date": "2020-01-22",
...             "isPresent": false
...           }
...         ]
...       },
...       {
...         "name": "Davis",
...         "attendance": [
...           {
...             "date": "2020-01-20",
```

```

...         "isPresent": false
...     },
...     {
...         "date": "2020-01-21",
...         "isPresent": true
...     },
...     {
...         "date": "2020-01-22",
...         "isPresent": true
...     }
... ]
... }
... ]
... },
... {
...     "batch": "b2",
...     "students": [
...         {
...             "name": "Clark",
...             "attendance": [
...                 {
...                     "date": "2020-01-20",
...                     "isPresent": true
...                 },
...                 {
...                     "date": "2020-01-21",
...                     "isPresent": true
...                 },
...                 {
...                     "date": "2020-01-22",
...                     "isPresent": true
...                 }
...             ]
...         },
...         {
...             "name": "Evans",
...             "attendance": [
...                 {
...                     "date": "2020-01-20",
...                     "isPresent": true
...                 },
...                 {
...                     "date": "2020-01-21",
...                     "isPresent": true
...                 },
...                 {
...                     "date": "2020-01-22",
...                     "isPresent": false
...                 }
...             ]
...         }
...     ]
... }

```

```

...     }
...   ]
... }
... ]
... }
... ])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("606977c09a85c18b642af42b"),
    ObjectId("606977c09a85c18b642af42c")
  ]
}

```

2. Write a query to insert student details in student collection

```

db.student.insertMany([
...  {
...    "name": "Adams",
...    "batch": "b1",
...    "email": "adams@gmail.com"
...  },
...  {
...    "name": "Baker",
...    "batch": "b1",
...    "email": "baker@gmail.com"
...  },
...  {
...    "name": "Clark",
...    "batch": "b2",
...    "email": "clark@gmail.com"
...  },
...  {
...    "name": "Davis",
...    "batch": "b1",
...    "email": "davis@gmail.com"
...  },
...  {
...    "name": "Evans",
...    "batch": "b2",
...    "email": "evans@gmail.com"
...  }
... ])

```

```
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("6069792e9a85c18b642af42d"),
    ObjectId("6069792e9a85c18b642af42e"),
    ObjectId("6069792e9a85c18b642af42f"),
    ObjectId("6069792e9a85c18b642af430"),
    ObjectId("6069792e9a85c18b642af431")
  ]
}
```

3. Write a query to Mark attendance for the students at respective batches on Jan 23 2020.

```
> db.batch.update({batch:"b1"}, {$push : {"students.0.attendance":{"date":"2020-01-23",
"isPresent":true}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.batch.update({batch:"b1"}, {$push : {"students.1.attendance":{"date":"2020-01-23",
"isPresent":false}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.batch.update({batch:"b1"}, {$push : {"students.2.attendance":{"date":"2020-01-23",
"isPresent":true}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.batch.update({batch:"b2"}, {$push : {"students.0.attendance":{"date":"2020-01-23",
"isPresent":true}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.batch.update({batch:"b2"}, {$push : {"students.1.attendance":{"date":"2020-01-23",
"isPresent":true}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.batch.find().pretty();
{
  "_id" : ObjectId("606977c09a85c18b642af42b"),
  "batch" : "b1",
  "students" : [
    {
      "name" : "Adams",
      "attendance" : [
        {
          "date" : "2020-01-20",
          "isPresent" : true
        },
        {
          "date" : "2020-01-21",
          "isPresent" : false
        }
      ]
    }
  ]
}
```

```

        "date" : "2020-01-22",
        "isPresent" : true
    },
    {
        "date" : "2020-01-23",
        "isPresent" : true
    }
]
},
{
    "name" : "Baker",
    "attendance" : [
        {
            "date" : "2020-01-20",
            "isPresent" : true
        },
        {
            "date" : "2020-01-21",
            "isPresent" : true
        },
        {
            "date" : "2020-01-22",
            "isPresent" : false
        },
        {
            "date" : "2020-01-23",
            "isPresent" : false
        }
    ]
},
{
    "name" : "Davis",
    "attendance" : [
        {
            "date" : "2020-01-20",
            "isPresent" : false
        },
        {
            "date" : "2020-01-21",
            "isPresent" : true
        },
        {
            "date" : "2020-01-22",
            "isPresent" : true
        },
        {
            "date" : "2020-01-23",
            "isPresent" : true
        }
    ]
}

```

```

    }
  ]
}
]
{
  "_id" : ObjectId("606977c09a85c18b642af42c"),
  "batch" : "b2",
  "students" : [
    {
      "name" : "Clark",
      "attendance" : [
        {
          "date" : "2020-01-20",
          "isPresent" : true
        },
        {
          "date" : "2020-01-21",
          "isPresent" : true
        },
        {
          "date" : "2020-01-22",
          "isPresent" : true
        },
        {
          "date" : "2020-01-23",
          "isPresent" : true
        }
      ]
    },
    {
      "name" : "Evans",
      "attendance" : [
        {
          "date" : "2020-01-20",
          "isPresent" : true
        },
        {
          "date" : "2020-01-21",
          "isPresent" : true
        },
        {
          "date" : "2020-01-22",
          "isPresent" : false
        },
        {
          "date" : "2020-01-23",
          "isPresent" : true
        }
      ]
    }
  ]
}

```

```

    }
  ]
}

```

4. Write a query to display the students attending batch1 along with their attendance history.

```

> db.batch.find({batch:"b1"}).pretty();
{
  "_id" : ObjectId("606977c09a85c18b642af42b"),
  "batch" : "b1",
  "students" : [
    {
      "name" : "Adams",
      "attendance" : [
        {
          "date" : "2020-01-20",
          "isPresent" : true
        },
        {
          "date" : "2020-01-21",
          "isPresent" : false
        },
        {
          "date" : "2020-01-22",
          "isPresent" : true
        },
        {
          "date" : "2020-01-23",
          "isPresent" : true
        }
      ]
    },
    {
      "name" : "Baker",
      "attendance" : [
        {
          "date" : "2020-01-20",
          "isPresent" : true
        },
        {
          "date" : "2020-01-21",
          "isPresent" : true
        },
        {
          "date" : "2020-01-22",

```

```

        "isPresent" : false
      },
      {
        "date" : "2020-01-23",
        "isPresent" : false
      }
    ]
  },
  {
    "name" : "Davis",
    "attendance" : [
      {
        "date" : "2020-01-20",
        "isPresent" : false
      },
      {
        "date" : "2020-01-21",
        "isPresent" : true
      },
      {
        "date" : "2020-01-22",
        "isPresent" : true
      },
      {
        "date" : "2020-01-23",
        "isPresent" : true
      }
    ]
  }
]
}

```

MySQL:

-- MySQL Workbench Forward Engineering

```

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO
_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

```

-- Schema StudentDB

```
-- Schema StudentDB
```

```
CREATE SCHEMA IF NOT EXISTS `StudentDB` DEFAULT CHARACTER SET utf8 ;  
USE `StudentDB` ;
```

```
-- Table `StudentDB`.`Students`
```

```
CREATE TABLE IF NOT EXISTS `StudentDB`.`Students` (  
  `id` INT NOT NULL,  
  `name` VARCHAR(30) NOT NULL,  
  `age` INT NOT NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

```
-- Table `StudentDB`.`Batch`
```

```
CREATE TABLE IF NOT EXISTS `StudentDB`.`Batch` (  
  `batch_name` VARCHAR(30) NOT NULL,  
  `student_id` INT NOT NULL,  
  INDEX `student_id_idx` (`student_id` ASC) VISIBLE,  
  CONSTRAINT `batch_name`  
    FOREIGN KEY (`student_id`)  
      REFERENCES `StudentDB`.`Students` (`id`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE)  
ENGINE = InnoDB;
```

```
-- Table `StudentDB`.`attendance`
```

```
CREATE TABLE IF NOT EXISTS `StudentDB`.`attendance` (  
  `b_name` VARCHAR(30) NOT NULL,  
  `givedate` VARCHAR(30) NOT NULL,  
  `is_present` TINYINT NOT NULL,  
  `s_id` INT NOT NULL,  
  INDEX `student_id_idx` (`s_id` ASC) VISIBLE,  
  CONSTRAINT `student_id`  
    FOREIGN KEY (`s_id`)  
      REFERENCES `StudentDB`.`Students` (`id`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE)  
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

```
-- Data for table `StudentDB`.`Students`
```

```
START TRANSACTION;
USE `StudentDB`;
INSERT INTO `StudentDB`.`Students` (`id`, `name`, `age`) VALUES (1, 'Adams', 13);
INSERT INTO `StudentDB`.`Students` (`id`, `name`, `age`) VALUES (2, 'Baker', 13);
INSERT INTO `StudentDB`.`Students` (`id`, `name`, `age`) VALUES (3, 'Clark', 14);
INSERT INTO `StudentDB`.`Students` (`id`, `name`, `age`) VALUES (4, 'Davis', 15);
INSERT INTO `StudentDB`.`Students` (`id`, `name`, `age`) VALUES (5, 'Evans', 15);
INSERT INTO `StudentDB`.`Students` (`id`, `name`, `age`) VALUES (6, 'Frank', 17);
INSERT INTO `StudentDB`.`Students` (`id`, `name`, `age`) VALUES (7, 'Ghosh', 14);
INSERT INTO `StudentDB`.`Students` (`id`, `name`, `age`) VALUES (8, 'Hills', 13);
```

```
COMMIT;
```

```
-- Data for table `StudentDB`.`Batch`
```

```
START TRANSACTION;
USE `StudentDB`;
INSERT INTO `StudentDB`.`Batch` (`batch_name`, `student_id`) VALUES ('B1', 1);
INSERT INTO `StudentDB`.`Batch` (`batch_name`, `student_id`) VALUES ('B2', 2);
INSERT INTO `StudentDB`.`Batch` (`batch_name`, `student_id`) VALUES ('B2', 3);
INSERT INTO `StudentDB`.`Batch` (`batch_name`, `student_id`) VALUES ('B3', 4);
INSERT INTO `StudentDB`.`Batch` (`batch_name`, `student_id`) VALUES ('B1', 5);
INSERT INTO `StudentDB`.`Batch` (`batch_name`, `student_id`) VALUES ('B4', 6);
INSERT INTO `StudentDB`.`Batch` (`batch_name`, `student_id`) VALUES ('B3', 7);
INSERT INTO `StudentDB`.`Batch` (`batch_name`, `student_id`) VALUES ('B5', 8);
```

```
COMMIT;
```

```
-- Data for table `StudentDB`.`attendance`
```

```
START TRANSACTION;
USE `StudentDB`;
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B1', 'Jan1', true, 1);
```

```

INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B1',
'Jan2', true, 1);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B1',
'Jan3', true, 1);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B1',
'Jan1', true, 5);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B1',
'Jan2', false, 5);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B1',
'Jan3', true, 5);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B2',
'Jan1', false, 2);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B2',
'Jan2', true, 2);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B2',
'Jan3', true, 2);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B2',
'Jan1', true, 3);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B2',
'Jan2', false, 3);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B2',
'Jan3', false, 3);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B3',
'Jan1', true, 4);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B3',
'Jan2', true, 4);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B3',
'Jan3', true, 4);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B3',
'Jan1', false, 7);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B3',
'Jan2', true, 7);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B3',
'Jan3', false, 7);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B4',
'Jan1', false, 6);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B4',
'Jan2', false, 6);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B4',
'Jan3', false, 6);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B5',
'Jan1', true, 8);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B5',
'Jan2', true, 8);
INSERT INTO `StudentDB`.`attendance` (`b_name`, `givedate`, `is_present`, `s_id`) VALUES ('B5',
'Jan3', true, 8);

```



```

COMMIT;

```



1. Write a sql query to display a table containing students attending batch1 along with their attendance history

```
1 • use studentdb;
2 • SELECT s_id,name,b_name,givedate
3 FROM attendance inner join batch on attendance.s_id=batch.student_id join students on batch.student_id=students.id
4 where batch_name="b1" and is_present=1;
```

Result Grid				
Filter Rows: <input type="text"/>				
Export:  Wrap Cell Content: 				
s_id	name	b_name	givedate	
1	Adams	B1	Jan1	
1	Adams	B1	Jan2	
1	Adams	B1	Jan3	
5	Evans	B1	Jan1	
5	Evans	B1	Jan3	

2. Write a sql query to display a table containing batch details and attendance history of student 1

```
1 • use studentdb;
2 • SELECT s_id,name,b_name,givedate
3 FROM attendance inner join batch on attendance.s_id=batch.student_id join students on batch.student_id=students.id
4 where s_id=1;
```

Result Grid				
Filter Rows: <input type="text"/>				
Export:  Wrap Cell Content: 				
s_id	name	b_name	givedate	
1	Adams	B1	Jan1	
1	Adams	B1	Jan2	
1	Adams	B1	Jan3	