

Bitwise operator:

- In our example its a 32 bit each bit representing different things
- Bitwise value 7 means all access 7 -> 111

Bit 3	Bit 2	Bit 1
cafe	campus	lobby

Bitwise type:

Name	Description
<code>\$bitsAllClear</code>	Matches numeric or binary values in which a set of bit positions <i>all</i> have a value of 0.
<code>\$bitsAllSet</code>	Matches numeric or binary values in which a set of bit positions <i>all</i> have a value of 1.
<code>\$bitsAnyClear</code>	Matches numeric or binary values in which <i>any</i> bit from a set of bit positions has a value of 0.
<code>\$bitsAnySet</code>	Matches numeric or binary values in which <i>any</i> bit from a set of bit positions has a value of 1.

Query:

In MongoDB, a query is a way to search for and retrieve documents from a collection that match specified criteria. Queries are typically performed using the find() method, which allows you to define filters and conditions to narrow down the results.

Here is a basic example of a MongoDB query:

```
javascript
db.collections.find({ field: value })
```

This query searches for documents in the specified collection where field equals value.

```
CONST LOBBY_PERMISSION=1;
CONST CAMPUS_PERMISSION=2;
```

Two constants are defined: LOOBY_PERMISSION with a value 1 and CAMPUS_PERMISSION WITH A VALUE 2.

To find students with both lobby and campus permission we use

```
db.students_permission.find({
  permission : {$bitsAllSets:[LOBBY_PERMISSION,CAMPUS_PERMISSION]}
});
```

```
b> db.students_permission.find({permissions:{$bitsAllSet:[LOBBY_PERMISSION,CAMPUS_PERMISSION]}});
{
  _id: ObjectId('6663ff4286ef416122dcfd5'),
  name: 'George',
  age: 21,
  permissions: 6
},
{
  _id: ObjectId('6663ff4286ef416122dcfd6'),
  name: 'Henry',
  age: 27,
  permissions: 7
},
{
  _id: ObjectId('6663ff4286ef416122dcfd7'),
  name: 'Isla',
  age: 18,
  permissions: 6
}
b>
```

To find all the students in the collections we use

```
db.students_permission.find({
  permission : {$bitsAllSets:[LOBBY_PERMISSION,CAMPUS_PERMISSION]}
}).count()
```

```
db> db.students_permission.find({permissions:{$bitsAllSet:[LOBBY_PERMISSION]}}).count();
9
```

Here we can find the total count of student permission.

\$bitsAllSets:

In MongoDB, **\$bitsAllSet** is an operator that matches documents where all of the bit positions given by the query are set (i.e. 1) in a specified field. The field value must be either numeric or a BinData instance for the operator to work.

Geospatial Query:

A geospatial query involves retrieving information from a database based on geographic locations and spatial relationships. These queries are used in Geographic Information Systems (GIS) to analyze and visualize spatial data.

```
_id: 1
name: "Coffee Shop A"
▼ location: Object
  type: "Point"
  ▶ coordinates: Array (2)
```

```
db.locations.find({
  locations:{
    @geoWithin:{
      $centerSphere:[[-74.005,40.712],0.00621376]
    }
  }
});
```

```
l> db.students_permission.find({permissions:{$bitsAllSet:[LOBBY_PERMISSION,CAMPUS_PERMISSION]}});
{
  _id: ObjectId('6663ff4286ef416122dcfcd5'),
  name: 'George',
  age: 21,
  permissions: 6
},
{
  _id: ObjectId('6663ff4286ef416122dcfcd6'),
  name: 'Henry',
  age: 27,
  permissions: 7
},
{
  _id: ObjectId('6663ff4286ef416122dcfcd7'),
  name: 'Isla',
  age: 18,
  permissions: 6
}
l>
```

Geospatial query operations:

Name	Description
<code>\$geoIntersects</code>	Selects geometries that intersect with a GeoJSON geometry. The <code>2dsphere</code> index supports <code>\$geoIntersects</code> .
<code>\$geoWithin</code>	Selects geometries within a bounding GeoJSON geometry . The <code>2dsphere</code> and <code>2d</code> indexes support <code>\$geoWithin</code> .
<code>\$near</code>	Returns geospatial objects in proximity to a point. Requires a geospatial index. The <code>2dsphere</code> and <code>2d</code> indexes support <code>\$near</code> .
<code>\$nearSphere</code>	Returns geospatial objects in proximity to a point on a sphere. Requires a geospatial index. The <code>2dsphere</code> and <code>2d</code> indexes support <code>\$nearSphere</code> .