

```
table.js

js Copy Edit

const { Row } = require('./row');
const { Index } = require('./index');

class Table {
  constructor(name, cols) {
    this.name = name;
    this.cols = cols;
    this.index = {};
    this.rows = [];
  }

  addData(data) {
    if (data.length !== this.cols.length) {
      console.log("Column type not match");
      return;
    }
    const rowData = new Row(Object.fromEntries(this.cols.map((col, i) => [col, data[i]])));
    this.rows.push(rowData);

    for (const column in this.index) {
      this.index[column].addDataInIndex(rowData);
    }

    console.log(`row with ${JSON.stringify(rowData)} has been added`);
  }

  fetchData(filter = {}) {
    if (Object.keys(filter).length === 0) {
      return this.rows.map(row => row.data);
    }
  }
}
```

```
row.js

js

class Row {
  constructor(data) {
    this.data = data;
  }

  toString() {
    return JSON.stringify(this.data);
  }
}

module.exports = { Row };
```

```
const [column, value] = Object.entries(filter)[0];

if (this.index.hasOwnProperty(column)) {
  return this.index[column].getData(value);
} else {
  return this.rows.filter(row => row.data[column] === value);
}

updateData(data) {
  // To be implemented
}
```

```
addIndex(colName) {
  if (!this.cols.includes(colName)) {
    console.log("Column not present");
    return;
  } else if (this.index.hasOwnProperty(colName)) {
    console.log("Index already created over this column");
    return;
  }

  const index = new Index(colName);
  this.index[colName] = index;

  for (const row of this.rows) {
    index.addDataInIndex(row);
  }

  console.log(`Index created successfully over col ${colName}`);
}
```

```
print() {
  console.log(this.cols);
  console.log(this.rows);
  console.log(this.index);
}

module.exports = { Table };
```