SQL

Tool Used :

Oracle  
SQL Developer  
17.3.1

is an [Integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for working with [SQL](https://en.wikipedia.org/wiki/SQL" \o "SQL)in [Oracle databases](https://en.wikipedia.org/wiki/Oracle_database). [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation) provides this product [free](https://en.wikipedia.org/wiki/Freeware); it uses the [Java Development Kit](https://en.wikipedia.org/wiki/Java_Development_Kit).

SQL connection

npm install mysql

var mysql = require('mysql');  
  
var con = mysql.createConnection({  
  host: "localhost",  
  user: "yourusername",  
  password: "yourpassword"  
});  
  
con.connect(function(err) {  
  if (err) throw err;  
  console.log("Connected!");  
  con.query("**CREATE DATABASE mydb**", function (err, result) {  
    if (err) throw err;  
    console.log("Database created");  
  });  
});

CREATE TABLE

var mysql = require('mysql');  
  
var con = mysql.createConnection({  
  host: "localhost",  
  user: "yourusername",  
  password: "yourpassword",  
**database: "mydb"**  
});  
  
con.connect(function(err) {  
  if (err) throw err;  
  console.log("Connected!");  
  var sql = "**CREATE TABLE customers (name VARCHAR(255), address VARCHAR(255)**)";  
  con.query(sql, function (err, result) {  
    if (err) throw err;  
    console.log("Table created");  
  });  
});

**Select**

var sql = require('mysql');

var con = sql.createConnection({

host : "loaclhost",

user : "admin",

password : "admin",

**db : "mydb"**

});

con.connect(function(err){

if(err){

console.log('err in connection '+err);

}

else{

var query = "select \* from t1";

con.query(query,function(err,data){

if(err){

console.log('err in connection '+err);

}

else{

console.log('success in sele '+err);

}

});

}

});

**Some queries**

**Insert**

**var sql = "INSERT INTO customers (name, address) VALUES ('Company Inc', 'Highway 37')";**  con.query(sql, function (err, result)

**Delete from table**

**var sql = "DELETE FROM customers WHERE address = 'Mountain 21'";**

**Drop table**

**var sql = "DROP TABLE customers";**

**Update table**

**var sql = "UPDATE customers SET address = 'Canyon 123' WHERE address = 'Valley 345'";**

**LIMIT -** Select the 5 first records in the "customers" table:

var sql = "SELECT \* FROM customers **LIMIT 5**";

## **Join Two or More Tables**

You can combine rows from two or more tables, based on a related column between them, by using a JOIN statement.

Consider you have a "users" table and a "products" table:

### **users**

[  
  { id: 1, name: 'John', favorite\_product: 154},  
  { id: 2, name: 'Peter', favorite\_product: 154},  
  { id: 3, name: 'Amy', favorite\_product: 155},  
  { id: 4, name: 'Hannah', favorite\_product:},  
  { id: 5, name: 'Michael', favorite\_product:}  
]

### **products**

[  
  { id: 154, name: 'Chocolate Heaven' },  
  { id: 155, name: 'Tasty Lemons' },  
  { id: 156, name: 'Vanilla Dreams' }  
]

These two tables can be combined by using users' favorite\_product field and products' id field.

### **Example**

Select records with a match in both tables:

  var sql = "SELECT users.name AS user, products.name AS favorite FROM users **JOIN products ON users.favorite\_product = products.id**";

**Note:** You can use INNER JOIN instead of JOIN. They will both give you the same result.

C:\Users\Your Name>node demo\_db\_join.js

Which will give you this result:

[  
  { user: 'John', favorite: 'Chocolate Heaven' },  
  { user: 'Peter', favorite: 'Chocolate Heaven' },  
  { user: 'Amy', favorite: 'Tasty Lemons' }  
]

As you can see from the result above, only the records with a match in both tables are returned.

## **Left Join**

If you want to return all users, no matter if they have a favorite product or not, use the LEFT JOIN statement:

Select all users and their favorite product:

SELECT users.name AS user,  
products.name AS favorite  
FROM users **LEFT JOIN** products ON users.favorite\_product = products.id

Which will give you this result:

[  
  { user: 'John', favorite: 'Chocolate Heaven' },  
  { user: 'Peter', favorite: 'Chocolate Heaven' },  
  { user: 'Amy', favorite: 'Tasty Lemons' },  
  { user: 'Hannah', favorite: null },  
  { user: 'Michael', favorite: null }  
]

## **Right Join**

If you want to return all products, and the users who have them as their favorite, even if no user have them as their favorite, use the RIGHT JOIN statement:

### **Example**

Select all products and the user who have them as their favorite:

SELECT users.name AS user,  
products.name AS favorite  
FROM users **RIGHT JOIN** products ON users.favorite\_product = products.id

Which will give you this result:

[  
  { user: 'John', favorite: 'Chocolate Heaven' },  
  { user: 'Peter', favorite: 'Chocolate Heaven' },  
  { user: 'Amy', favorite: 'Tasty Lemons' },  
  { user: null, favorite: 'Vanilla Dreams' }  
]

**Note:** Hannah and Michael, who have no favorite product, are not included in the result.