**PostgreSQL Functions:**

**create database postgrefunction**

**create table Student(sid int,sname varchar(20),scity varchar(20))**

**select \* from Student**

**==================================== INSET FUNCTION ====================================**

create or replace function insertStudents(sid int,sname varchar(20),scity varchar(20))

returns void as $$

begin

insert into Student values(sid,sname,scity);

end;

$$ language plpgsql;

select insertStudents('01','Madhan','Hanamkonda');--this Query we have use in restAPI

select insertStudents('03','Vineeth','Warangal');

select insertStudents('02','Sheshank','kazipet'),('03','Vineeth','Warangal');-- this will inserts only first record

select \* from Student

select insertStudents(); -- this doesnot works

select Students(); -- this doesnot works

**============================= GETTING ALL RECORDS FUNCTION ==============================**

create or replace function selectstd()

returns table(sid int,sname varchar,scity varchar) as $body$

begin return query

select \* from Student;

end $body$ language plpgsql stable

select selectstd() --this Query we have use in restAPI

**=========================== GETTING SINGLE RECORD FUNCTION ===============================**

create or replace function select\_std\_id(id int)

returns table(sid int,sname varchar,scity varchar) as $body$

begin return query

select \* from Student where Student.sid=id;

end $body$ language plpgsql stable

select select\_std\_id(2);--this Query we have use in restAPI

**================================ UPDATE FUNCTION =====================================**

create or replace function update\_std(sid int,sname varchar,scity varchar)

returns void as $body$

begin

update Student set sname=$2,scity=$3 where Student.sid=$1;

end;

$body$ language plpgsql volatile ;

select update\_std(3,'vineeth reddy','hyd') --this Query we have use in restAPI

**================================== DELETE FUNCTION ==================================**

create or replace function delete\_std(id int)

returns void as $$

begin

delete from Student where Student.sid=id;

end;

$$ language plpgsql volatile

select delete\_std(1) --this Query we have use in restAPI

**==========================================================================================**

**REST API FOR ABOVE FUNCTIONS**

**==========================================================================================**

var promise = require('bluebird');

var bodyParser = require('body-parser')

var options = { promiseLib: promise }

var pgp = require('pg-promise')(options);

var express = require('express');

var app = new express();

var fs = require('fs');

var path = require('path');

app.use(express.static(path.join(\_\_dirname, 'public')));

app.use(bodyParser.urlencoded({ extended: true }));

app.use(bodyParser.json());

var cs = 'postgres://postgres:root@localhost:5432/postgrefunction';

var db = pgp(cs);

/\* Getting all records from table \*/

app.get('/student', (req, res, next) => {

db.any('select selectstd()').then(function (data) {

res.send(data);

})

})

/\* Getting a single record baesd on id \*/

app.get('/student/:id', (req, res, next) => {

var i = req.params.id;

db.any('select select\_std\_id($1)', i).then(function (data) {

res.send(data);

})

})

/\* Inserting a record \*/

app.post('/student', (req, res, next) => {

var id = req.body.sid;

var sn = req.body.sname;

var c = req.body.city;

db.none('select insertStudents($1,$2,$3)', [id, sn, c])

.then(function () {

res.status(200).send({ message: "Record Inserted Success.." });

})

})

/\* Updating a record \*/

app.put('/student/:id', (req, res, next) => {

var i = req.params.id;

var u = req.body.sname;

var p = req.body.scity;

db.none('select update\_std($1,$2,$3)', [i, u, p]).then(function () {

console.log('Record Updated Success ...')

res.status(200).send({ message: "Updated Succes.." })

})

})

/\* Deleting the record \*/

app.delete('/student/:id', (req, res, next) => {

var i = parseInt(req.params.id);

db.any('select delete\_std($1)', i).then(function (data) {

res.status(200).send({ message: "Deleted Success.." })

})

})

app.listen(3600, (err) => {

if (err)

console.log('Server Cant Start ...Erorr....');

else

console.log('Server Started at : http://localhost:3600')

})