

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

ЗВІТ
з лабораторної роботи № 3
«Засоби оптимізації роботи СУБД PostgreSQL»

Виконав:

студент 3-го курсу, групи КП-80,
спеціальності 121 – Інженерія
програмного забезпечення
Ящук Сергій Олегович

Перевірив:

к. т. н, старший викладач
Радченко Костянтин Олександрович

Київ – 2020

Мета роботи

Здобуття є здобуття практичних навичок використання засобів оптимізації СУБД PostgreSQL.

Завдання (Варіант 22)

Загальне завдання роботи полягає у наступному:

1. Перетворити модуль “Модель” з шаблону MVC лабораторної роботи No2 у вигляд об’єктно-реляційної проекції (ORM).
2. Створити та проаналізувати різні типи індексів у PostgreSQL.
3. Розробити тригер бази даних PostgreSQL.

Хід роботи

1. Опис схеми бази даних..

При розробленні була використана ORM Sequelize та створені класи-сутності для об'єктів-сутностей: client, driver trip.

```
const Trip = sequelize.define("trips", {
  id: {
    type: Sequelize.INTEGER,
    autoIncrement: true,
    primaryKey: true,
    allowNull: false
  },
  price: {
    type: Sequelize.STRING,
    allowNull: false
  },
  address: {
    type: Sequelize.STRING,
    allowNull: true
  },
}, {
  freezeTableName: true,
  timestamps: false,
});

Trip.belongsTo(Client, {
  as: 'client',
  foreignKey: 'client_id'
});

Trip.belongsTo(Driver, {
  as: 'driver',
  foreignKey: 'driver_id'
});
```

```
const Client = sequelize.define("clients", {
  id: {
    type: Sequelize.INTEGER,
    autoIncrement: true,
    primaryKey: true,
    allowNull: false
  },
  name: {
    type: Sequelize.STRING,
    allowNull: false
  },
  regdate : {
    type: 'TIMESTAMP',
    defaultValue: Sequelize.literal('CURRENT_TIMESTAMP'),
    allowNull: false
  },
}, {
  freezeTableName: true,
  timestamps: false,
});
```

```
const Driver = sequelize.define("drivers", {
  id: {
    type: Sequelize.INTEGER,
    autoIncrement: true,
    primaryKey: true,
    allowNull: false
  },
  name: {
    type: Sequelize.STRING,
    allowNull: false
  },
  car: {
    type: Sequelize.STRING,
    allowNull: false
  },
}, {
  freezeTableName: true,
  timestamps: false,
});
```

Рис 1. Класи-сутності

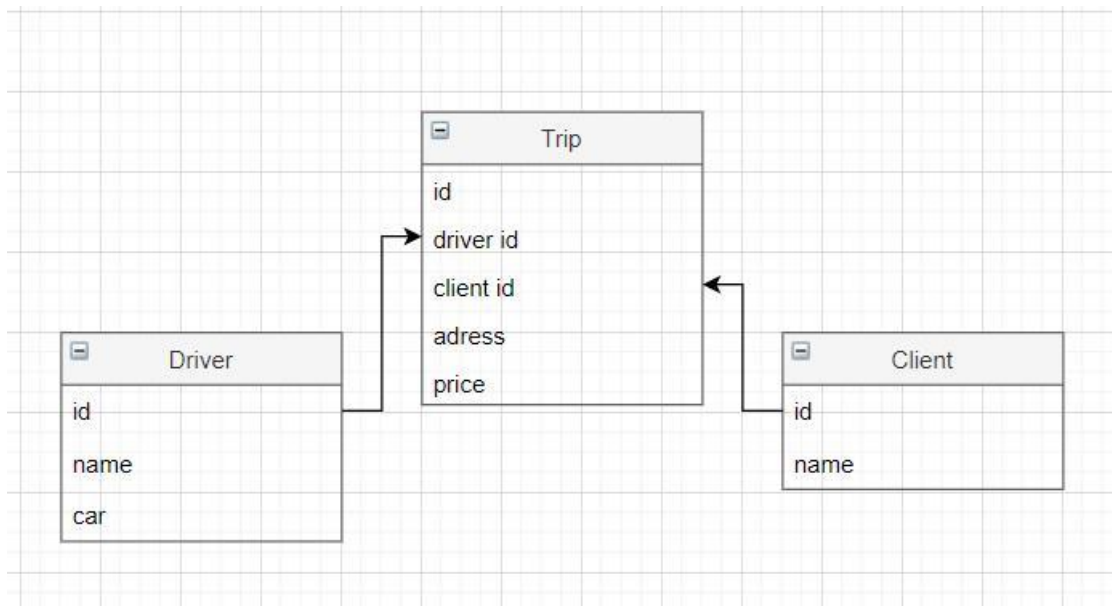


Рис 1. Схема БД.

2. Приклади запитів

```
Program started
Command type: entity commandOperation
Entities: clients, drivers, trips
Command Operations: create, readone, readmany, update, delete, search, generate
Enter the command to execute: trips readmany
Enter limit: 10
```

```
[
  trips {
    dataValues: {
      id: 1,
      price: '150',
      address: 'Davis Court',
      client_id: 7,
      driver_id: 1
    },
    _previousDataValues: {
      id: 1,
      price: '150',
      address: 'Davis Court',
      client_id: 7,
      driver_id: 1
    },
    _changed: Set(0) {},
    _options: {
      isNewRecord: false,
      _schema: null,
      _schemaDelimiter: '',
      raw: true,
      attributes: [Array]
    },
    isNewRecord: false
  },
  trips {
    dataValues: {
      id: 2,
      price: '280',
      address: 'Dennison Street',
      client_id: 4,
    },
    _previousDataValues: {
      id: 4,
      price: '220',
      address: 'Shobe Lane',
      client_id: 4,
      driver_id: 3
    },
    _changed: Set(0) {},
    _options: {
      isNewRecord: false,
      _schema: null,
      _schemaDelimiter: '',
      raw: true,
      attributes: [Array]
    },
    isNewRecord: false
  },
  trips {
    dataValues: {
      id: 5,
      price: '500',
      address: 'Coleman Avenue',
      client_id: 5,
      driver_id: 7
    },
    _previousDataValues: {
      id: 4,
      price: '220',
      address: 'Shobe Lane',
      client_id: 4,
      driver_id: 3
    },
    _changed: Set(0) {},
    _options: {
      isNewRecord: false,
      _schema: null,
      _schemaDelimiter: '',
      raw: true,
      attributes: [Array]
    },
    isNewRecord: false
  }
]
```

Рис 3. Запит всіх поїздок

```
Program started
Command type: entity commandOperation
Entities: clients, drivers, trips
Command Operations: create, readone, readmany, update, delete, search, generate
Enter the command to execute: trips create
Enter driver_id: 2
Enter client_id: 2
Enter price: 301
Enter address: testaddress

Executing (default): CREATE TABLE IF NOT EXISTS "clients" ("id" SERIAL , "name" VARCHAR(255) NOT NULL, "regd"
d));
Executing (default): INSERT INTO "trips" ("id","price","address","client_id","driver_id") VALUES (DEFAULT,$1,$
Executing (default): SELECT i.relname AS name, ix.indisprimary AS primary, ix.indisunique AS unique, ix.indkey
name) AS column_names, pg_get_indexdef(ix.indexrelid) AS definition FROM pg_class t, pg_class i, pg_index ix,
AND a.attrelid = t.oid AND t.relkind = 'r' and t.relname = 'clients' GROUP BY i.relname, ix.indexrelid, ix.in

Successfully created
```

29	32	4	7	301	31 incldon Ave
30	33	2	2	301	testaddress

Рис 4. Запит на створення поїздки

```

Program started
Command type: entity commandOperation
Entities: clients, drivers, trips
Command Operations: create, readone, readmany, update, delete, search, generate
Enter the command to execute: trips update
Enter id: 33
Enter driver_id: 3
Enter client_id: 3
Enter price: 1
Enter address: test2

Executing (default): CREATE TABLE IF NOT EXISTS "clients" ("id" SERIAL , "name" VARCHAR(255) , "address" VARCHAR(255) , "client_id" SERIAL , "driver_id" SERIAL , "price" SERIAL , "created_at" TIMESTAMP);
Executing (default): UPDATE "trips" SET "driver_id"=$1,"client_id"=$2,"price"=$3,"address"=$4 WHERE "id"=$5;
Executing (default): SELECT i.relname AS name, ix.indisprimary AS primary, ix.indexrelid AS indexrelid, t.relname AS column_names, pg_get_indexdef(ix.indexrelid) AS definition FROM pg_class t JOIN pg_index ix ON t.oid = ix.indrelid JOIN pg_attribute a ON t.oid = a.attrelid AND a.attrelid = t.oid AND t.relkind = 'r' and t.relname = 'clients' GROUP BY i.relname, ix.indexrelid, t.relname;
Successfully updated

```

32	4	7	361	5 Pineknoll Ave
33	3	3	1	test2

Рис 5. Запит на оновлення поїздки

```

Program started
Command type: entity commandOperation
Entities: clients, drivers, trips
Command Operations: create, readone, readmany, update, delete, search, generate
Enter the command to execute: trips delete
Enter id: 33

Executing (default): DELETE FROM "trips" WHERE "id" = '33'
Executing (default): CREATE TABLE IF NOT EXISTS "drivers" ("id" SERIAL , "name" VARCHAR(255) , "address" VARCHAR(255) , "client_id" SERIAL , "driver_id" SERIAL , "price" SERIAL , "created_at" TIMESTAMP);
Executing (default): SELECT i.relname AS name, ix.indisprimary AS primary, ix.indexrelid AS indexrelid, t.relname AS column_names, pg_get_indexdef(ix.indexrelid) AS definition FROM pg_class t JOIN pg_index ix ON t.oid = ix.indrelid JOIN pg_attribute a ON t.oid = a.attrelid AND a.attrelid = t.oid AND t.relkind = 'r' and t.relname = 'drivers' GROUP BY i.relname, ix.indexrelid, t.relname;
successfully removed

```

```

Program started
Command type: entity commandOperation
Entities: clients, drivers, trips
Command Operations: create, readone, readmany, update, delete, search, generate
Enter the command to execute: trips readone
Enter id: 33

Executing (default): SELECT "id", "price", "address", "client_id", "driver_id" FROM "trips" WHERE "id" = 33;
null

```

Рис 6. Запит на видалення поїздки

Висновок

Виконавши дану лабораторну роботу ми навчилися використовувати засоби оптимізації СУБД PostgreSQL.