

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Кафедра системного програмування та спеціалізованих комп’ютерних систем**

**Лабораторна робота №3**

З дисципліни

**«Бази даних і засоби управління»**

Виконав:

Студент 3 курсу

ФПМ групи КВ-71

Бусь Б.Б.

Перевірив:

Київ – 2019

**Засоби оптимізації роботи СУБД PostgreSQL**

Метою роботи є здобуття практичних навичок використання засобів оптимізації СУБД PostgreSQL.

Завдання роботи полягає у наступному:

1. Перетворити модуль “Модель” з шаблону MVC лабораторної роботи №2 у вигляд об’єктно-реляційної проекції (ORM).

2. Створити та проаналізувати різні типи індексів у PostgreSQL.

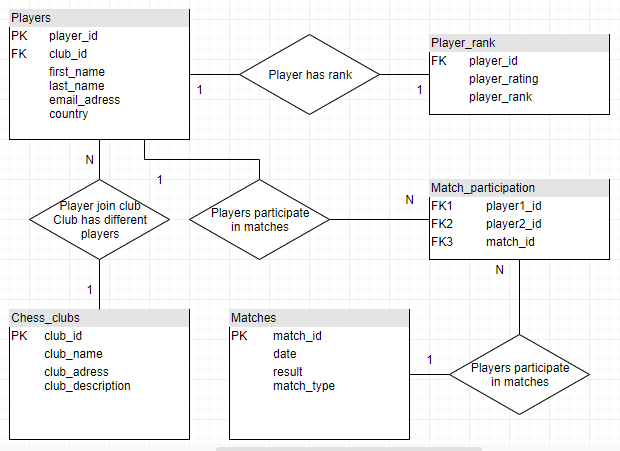
3. Розробити тригер бази даних PostgreSQL.

4. Навести приклади та проаналізувати рівні ізоляції транзакцій у PostgreSQL.

**Варіант 1**

У другому завданні проаналізувати індекси B-Tree, Hash.

Умова для тригера – before delete, insert



**player.php**

<?php

include 'query.php';

class players

{

static $table = 'players';

public $playerid;

public $clubid ;

public $firstname ;

public $lastname ;

public $emailadress ;

public $country ;

function \_\_construct($playerId, $pdo){

$atributes = $pdo->SelectData(self::$table, 'playerid' ,$playerId );

$this->playerid = $playerId;

$this->clubid = $atributes['clubid'];

$this->firstname = $atributes['firstname'];

$this->lastname = $atributes['lastname'];

$this->emailadress = $atributes['emailadress'];

$this->country = $atributes['country'];

}

static function addPlayer($data, $pdo){

$result = $pdo->insertInTable(self::$table, $data);

return $result;

}

static function deletePlayer($parameter, $valueParameter, $pdo){

$result = $pdo->deleteFromTable(self::$table, $parameter, $valueParameter);

return $result;

}

static function updatePlayer($parameter, $data, $valueParameter, $pdo){

$result = $pdo->UpdateTable(self::$table, $data, $parameter, $valueParameter);

return $result;

}

}

**query.php**

<?php

class Query

{

private $host = 'localhost';

private $port = '5432';

private $dbname = 'chess\_db';

private $user = 'postgres';

private $pdo;

function \_\_construct()

{

$pdo = new PDO("pgsql:host=".$this->host.";dbname=".$this->dbname."", $this->user);

$this->pdo = $pdo;

}

public function insertInTable($table, $data){

$insertsql = '';

$insertsqlval = '';

foreach ($data as $atribute => $atributeValue){

$insertsql .= $atribute. ' ,';

$insertsqlval .=':'.$atribute. ' ,';

}

$insertsql = mb\_substr($insertsql, 0, -1);

$insertsqlval = mb\_substr($insertsqlval, 0, -1);

$sql = "INSERT INTO $table ($insertsql)

VALUES ($insertsqlval)";

$result = $this->pdo->prepare($sql);

foreach ($data as $atribute => $atributeValue){

$result->bindParam(':'.$atribute, $data[$atribute]);

}

$result->execute();

return $result;

}

public function deleteFromTable($table, $parameter, $valueParameter){

$sql = "DELETE FROM $table WHERE $parameter = :$parameter";

$result = $this->pdo->prepare($sql);

$result->bindParam(':'.$parameter , $valueParameter);

$result->execute();

return $result;

}

public function UpdateTable ($table, $data, $parameter, $valueParameter ){

$updatesql ='';

foreach ($data as $atribute => $atributeValue){

$updatesql .= $atribute .' =:'.$atribute. ' ,';

}

$updatesql = mb\_substr($updatesql, 0, -1);

$sql = "UPDATE $table SET $updatesql

WHERE $parameter = :$parameter";

$result = $this->pdo->prepare($sql);

foreach ($data as $atribute => $atributeValue){

$result->bindParam(':'.$atribute, $data[$atribute]);

}

$result->bindParam(':'.$parameter, $valueParameter);

$result->execute();

return $result;

}

public function SelectAllData($table){

$sql = 'SELECT \* FROM $table';

$result = $this->pdo->prepare($sql);

$result->execute();

$red = $result->fetchAll();

return ($red);

}

public function SelectData($table, $parameter, $valueParameter){

$sql = "SELECT \* FROM $table WHERE $parameter = :$parameter";

$result = $this->pdo->prepare($sql);

$result->bindParam(':'.$parameter, $valueParameter);

$result->execute();

$red = $result->fetchAll();

return ($red[0]);

}

}