Министерство образования и науки РФ

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра Информационные технологии и автоматизированные системы

Базы данных

Лабораторная работа № 6

Тема: «Триггеры и хранимые процедуры»

Выполнил: студент группы Миннахметов Э.Ю.

Проверил: доцент кафедры ИТАС

Петренко А.А

г. Пермь – 2020

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ 3

1 АНАЛИЗ ПОСТАВЛЕННОЙ ЗАДАЧИ 4

2 ТЕХНОЛОГИЯ РЕАЛИЗАЦИИ 5

2.1 Архитектура приложения 5

2.2 Новые SQL-запросы 5

ЗАКЛЮЧЕНИЕ 9

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 10

# ВВЕДЕНИЕ

Цель: научится создавать и использовать представления в СУБД MySQL.

Задачи:

1. Изучить понятие VIEW;
2. Изучить и знать наизусть синтаксис;
3. Создать представление в базе данных лабораторной работы № 5;
4. Реализовать представления на базе запросов 2 лабораторной работы, либо 5 ЛБ в количестве 3 штук;
5. Проверить работу представлений, в коде реализуемой программы.

# 1 АНАЛИЗ ПОСТАВЛЕННОЙ ЗАДАЧИ

OpenServer – набор, который содержит в себе многие популярные инструменты веб-разработки, среди которых локальный сервер Apache, язык серверного программирования PHP (Hypertext Preprocessor – препроцессор гипертекста), СУБД MySQL, а также веб-клиент данной СУБД phpMyAdmin.

Для данной лабораторной работы HTML и CSS для фронт-энда будет достаточно, а роль серверного языка возьмет на себя PHP.

Ранее уже был реализован веб-проект, который можно взять за основу данной работы — это была лабораторная работа №4, которой был разработан веб-интерфейс просмотра содержимого таблицы, добавление новых записей, их обновление и удаление. Теперь же требуется доработать её триггерами, новой таблицей логирования и возможностью её просмотра.

**Вывод**, бэк-энд веб-приложения будет написан на связке языков PHP + MySQL, а фронт-энд на HTML + CSS.

**2 ТЕХНОЛОГИЯ РЕАЛИЗАЦИИ**

Разработка любого приложения начинается с построения его архитектуры, веб-приложение данной лабораторной работы не будет являться исключением, после чего будут рассмотрены новые SQL-запросы, используемые в веб-приложении.

**2.1 Архитектура приложения**

Особых изысков данного приложение требовать не будет и можно вообще обойтись без ООП, довольствуясь лишь парадигмой структурного программирования. Однако и такой код следует строить по архитектурному паттерну ***MVC*** (Model-View-Controller, Модель-Представление-Контроллер).

В качестве ***Модели*** будет взята ранее построенная таблица базы данных, используемая во второй и третьей лабораторной работе. За работу с моделью будут отвечать функции из файла ***database.php***.

За ***Представление*** будет отвечать файл ***layout.php***, содержащий в себе основу для вывода ***html***-кода, ***output.php*** – шаблон для вывода таблицы, и файл ***style.css***, отвечающий за стили элементов ***html***-кода.

***Контроллером*** в данном приложении будет выступать файл ***index.php*** – он будет обрабатывать запросы и запускать вывод ***html***-кода из функций подключенных ***php***-файлов.

**Подводя итог**, следует сказать, что архитектура веб-приложения была построена и теперь необходимо перейти к детальному рассмотрению SQL-запросов.

**2.2 Новые SQL-запросы**

Показательным примером использования триггеров будет ведением логов, т.е. использование журнала для документирования событий в базе данных. Ниже представлен запрос на создание таблицы логов для документирования событий таблицы лабораторной работы №4.

Листинг 2.1 – Создание таблицы Log

CREATE TABLE Log

(IdLog INT AUTO\_INCREMENT,

IdEmployee INT,

TypeOperation VARCHAR(6),

Description TEXT,

OperationDate DATETIME,

PRIMARY KEY (IdLog));

Полями такой таблицы являются идентификатор лога, идентификатор сотрудника, над учетной записью которого выполняется операция, тип операции (добавление, обновление или удаление), описание лога – конкретные изменения в таблице, дата выполнения операции.

Первый триггер будет отслеживать добавление записей, ниже представлено его объявление.

Листинг 2.2 – Создание триггера, реагирующего на добавление записи

CREATE TRIGGER TriggerInsert

AFTER INSERT ON Lab2

FOR EACH ROW

INSERT Log (IdEmployee, TypeOperation, Description, OperationDate)

VALUES (NEW.EmployeeId, 'INSERT', CONCAT('\"',

NEW.FirstName, '\", \"',

NEW.LastName, '\", \"',

NEW.PhoneNumber, '\", \"',

NEW.Salary, '\", \"',

NEW.Address, '\", \"',

NEW.Expirience, '\"'), NOW());

Описание данного триггера будет включать значения полей новой записи.

Второй триггер будет документировать изменения, выполняемые над записями, и он представлен ниже.

Листинг 2.3 – Создание триггера, реагирующего на обновление записи

CREATE TRIGGER TriggerUpdate

AFTER UPDATE ON Lab2

FOR EACH ROW

INSERT Log (IdEmployee, TypeOperation, Description, OperationDate)

VALUES (NEW.EmployeeId, 'UPDATE', CONCAT(

IF(OLD.FirstName != NEW.FirstName, CONCAT('\"', OLD.FirstName,

'\" => \"', NEW.FirstName, '\", '), ''),

IF(OLD.LastName != NEW.LastName, CONCAT('\"', OLD.LastName,

'\" => \"', NEW.LastName, '\", '), ''),

IF(OLD.PhoneNumber != NEW.PhoneNumber, CONCAT('\"',

OLD.PhoneNumber, '\" => \"', NEW.PhoneNumber, '\", '), ''),

IF(OLD.Salary != NEW.Salary, CONCAT('\"', OLD.Salary, '\" => \"',

NEW.Salary, '\", '), ''),

IF(OLD.Address != NEW.Address, CONCAT('\"', OLD.Address, '\" => \"',

NEW.Address, '\", '), ''),

IF(OLD.Expirience != NEW.Expirience, CONCAT('\"', OLD.Expirience,

'\" => \"', NEW.Expirience, '\"'), '')

), NOW());

Во избежание утери некоторых данных, изменяемые поля будут сохранять в описании лога свое начальное и конечное значения.

Что же касается последнего триггера, то он будет следить за удалениями записей:

Листинг 2.4 – Создание триггера, реагирующего на удаление записи

CREATE TRIGGER TriggerDelete

AFTER DELETE ON Lab2

FOR EACH ROW

INSERT Log (IdEmployee, TypeOperation, Description, OperationDate)

VALUES (OLD.EmployeeId, 'DELETE', CONCAT('\"', OLD.FirstName, '\", \"',

OLD.LastName, '\", \"',

OLD.PhoneNumber, '\", \"',

OLD.Salary, '\", \"',

OLD.Address, '\", \"',

OLD.Expirience, '\"'), NOW());

Описание таких логов будет включать последние значения полей записи.

**Вывод**, были приведены SQL-запросы, необходимые для выполнения данной лабораторной работы.

**По итогу главы**, веб-приложение написано.

# ЗАКЛЮЧЕНИЕ

# Был выполнено:

# создана таблица журнала логов;

1. построен веб-интерфейс просмотра таблицы Log;
2. созданы триггеры, реагирующие на добавление, обновление и удаление записей из таблицы Employee.

**Вывод**, цель данной лабораторной работы, а именно - реализация триггеров, была выполнена.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. К. Дж. Дейт. Введение в системы баз данных, 2005 г. — 1328 с.
2. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений, 2009 г. – 736 с.
3. <https://metanit.com/sql/mysql/>
4. <https://proselyte.net/tutorials/sql/>