

## ЛАБОРАТОРНАЯ РАБОТА №6 СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ПРЕДСТАВЛЕНИЙ И ИНДЕКСОВ

### 6.1 Цель работы

Ознакомится с принципом работы пользовательских представлений и индексов, продемонстрировать работу с ними на примере.

### 6.2 Введение

**Представления** – это таблицы чье содержание выбирается или получается из других таблиц. Они работают в запросах и операторах DML точно также, как и основные таблицы, но не содержат никаких собственных данных.

Представления подобны окнам, через которые возможно просматривать информацию, которая фактически хранится в базовой таблице. Представление – это фактически запрос, который выполняется всякий раз, когда представление становится темой команды. Вывод запроса при этом в каждый момент становится содержанием представления.

Можно выделить следующие назначения представления:

- когда необходимо дать пользователю возможность просматривать не всю таблицу, а только ее часть;
- для объединения двух или более таблиц;
- вывод статистических данных.

**Индекс** – это упорядоченный (буквенный или числовой) список столбцов или групп столбцов в таблице. Таблицы могут иметь большое количество строк, а, так как строки не находятся в каком-нибудь определенном порядке, их поиск по указанному значению может потребовать времени. Индексный адрес – это и забота, и в то же время обеспечение способа объединения всех значений в группы из одной или больше строк, которые отличаются одна от другой.

В то время как индекс значительно улучшает эффективность запросов, использование индекса несколько замедляет операции модификации DML (такие как INSERT и DELETE), а сам индекс занимает объем памяти. Следовательно, при создании таблицы необходимо определять, индексировать ее или нет. Индексы могут состоять из многочисленных полей. Если больше чем одно поле указывается для одного индекса, второе упорядочивается внутри первого, третье внутри второго, и так далее.

SQL автоматически создает индекс, когда создается ограничение PRIMARY KEY или UNIQUE.

## 6.3 Порядок выполнения работы

### 6.3.1 Создание представления

Для заданной вариантной базы данных разработать и выполнить с помощью SQL-редактора PHPMyAdmin следующие запросы:

- 1) запрос на создание представления для ввода данных в таблицу;
- 2) запрос, который продемонстрирует успешный ввод данных через созданное представление;
- 3) запрос, который продемонстрирует отрицательный результат при вводе данных через созданное представление (объяснить причину);
- 4) запрос на удаление созданного представления;
- 5) запрос на создание представления для вывода данных из двух или более таблиц одновременно;

*Дополнительно:*

- 6) модифицировать приложение лабораторной работы №4. Добавить окно для вывода данных из двух или более таблиц, предусмотренных вариантом задания.

### 6.3.2 Создание индекса

Разработать и выполнить с помощью SQL-редактора PHPMyAdmin следующие запросы:

- 1) запрос на создание уникального индекса;
- 2) запрос на создание индекса на поле, по которому сортируются данные;
- 3) запрос на создание индекса на поле, по которому происходит группировка во время агрегации;
- 4) запрос на создание составного индекса с учетом селективности колонок;
- 5) запрос на создание составного индекса, причем обосновать очередность перечисления полей в индексе (группировка, сортировка, частота использования) и их количество.

После каждого запроса на создание индекса представить соответствующий запрос на выборку, для которого индексация имеет смысл. После чего обязательно удалять созданный индекс.

## 6.4 Содержание отчета

Отчет должен содержать следующие разделы:

- 1) Титульный лист определенного образца.
- 2) Тема, цель работы, вариант задания (полностью представленная структура данных).
- 3) Ход работы:
  - исходные данные (концептуально-логическая схема БД и операторы создания всех таблиц БД на языке SQL);

- описание представлений и тексты всех запросов на создание и удаление представлений;
- тексты запросов на выборку и результаты их выполнения (объяснение полученных результатов);
- описание клиентского приложения;
- описание назначения индексов и тексты всех запросов на создание и удаление индексов;
- тексты запросов на выборку (после индексирования) и результаты их выполнения (объяснение полученных результатов);

**Все запросы должны иметь логический смысл. Без формулировки запросов на естественном языке отчет приниматься не будет.**

4) Выводы.

## **6.5 Контрольные вопросы**

- 1) Что такое "пользовательское представление"?
- 2) Условия редактируемости представлений?
- 3) Назначение пользовательских представлений?
- 4) Привести примеры пользовательских представлений?
- 5) Отличие представлений от таблиц?
- 6) Организация ограничения доступа пользователей к набору данных?
- 7) Условие обеспечения целостности данных?
- 8) Внешняя модель и создание представлений при разработке БД?
- 9) Назначение индексов.
- 10) Типы индексов.
- 11) Порядок выполнения запросов на выборку при создании составного индекса.
- 12) Когда целесообразно использование индексов?
- 13) Когда не стоит использовать индексы?

## **6.6 Рекомендации по выполнению работы**

### **6.6.1 Представления**

Команда создания представления: CREATE VIEW. Она состоит из слов CREATE VIEW (СОЗДАТЬ ПРЕДСТАВЛЕНИЕ), имени представления которое нужно создать, слова AS (КАК), и далее запроса на выборку данных, который и будет отражать содержимое представления.

**CREATE VIEW <имя представления> [(<имя столбца> [,<имя столбца>...])] AS<подзапрос>[WITH CHECK OPTION];**

**<подзапрос>::=<оператор SELECT>;**

**Пример:**

```
CREATE VIEW PhoneView AS
SELECT pl.vcFamIl, pl.vcName, pl.vcSurName,
```

```

        dDateBirthDay, vcPhoneNumber
FROM tbPeoples pl, tbPhoneNumbers pn
WHERE pn.idPeoples=*pl.idPeoples
Чтобы использовать это представление, сформулируем запрос SELECT:
SELECT * FROM PhoneView

```

Результат выполнения запроса:

vcFamil	vcName	vcSurName	dDateBirthDay	vcPhoneNumber
ИВАНОВ	ИВАН	ИВАНОВИЧ	2004-01-31	(978) 102-51-01
ИВАНОВ	ИВАН	ИВАНОВИЧ	2004-01-31	(978) 163-31-52
ПЕТРОВ	ИВАН	ИВАНОВИЧ	1971-04-03	(978) 112-02-46
...	...	...	....	.....

Представление является обновляемым, и может распространять изменения, вносимые в представление на базовые таблицы, только при выполнении некоторых условий. Если условия не выполняются, то представление может служить только для чтения. Термины "обновляемое" и "только для чтения" относятся к содержимому представления, а не к его объявлению.

Условия обновляемости представлений:

- представление является подмножеством одной таблицы, либо другого обновляемого представления (все столбцы базовых таблиц, не включенные в представление, должны позволять состояние NULL);
- оператор SELECT не должен содержать подзапросов, слова DISTINCT, раздела HAVING, агрегатных функций, объединения таблиц, пользовательских функций, хранимых процедур.

Итак, представление – это просто запрос на языке SQL, который выбирает данные, а в базе данных оно выглядит как таблица и работа с ним происходит также. Из представления можно выбирать данные SQL запросами и также назначать права доступа.

Удалить представление можно с помощью команды DROP VIEW:

**DROP VIEW name;**

Здесь name - имя удаляемого представления. Не существует способов изменить существующее представление. Можно удалить старое представление и заново создать новое.

## 6.6.2 Индексы

Синтаксис для создания индекса:

**CREATE [UNIQUE] [ CLUSTERED | NONCLUSTERED ] INDEX <имя индекса> ON<имя таблицы>(<имя столбца>,[<упоряд.>]|,<имя столбца> [<упоряд.>]...]);**

**<упоряд.>::=ASC (возрастание)| DESC (убывание);**  
**{ по умолчанию ASC }**

Таблица, конечно, должна уже быть создана и должна содержать имя

соответствующего столбца. Однажды созданный, индекс будет невидим пользователю.