

## Лабораторная работа №4. Представления. Индексы.

ППСУБД  
2курс. 5 группа.  
Кушнеров А.В. 2017-2018 г.

### *Цель работы:*

Изучение понятия представления как инструмента доступа к данным реляционной БД. Получение навыков проектирования и разработки представлений. Изучение и применение понятия индексации. Размышления о быстродействии индексируемых таблиц БД.

### Минимальные теоретические сведения

**Представление** – это запрос на выборку, сохранённый в базе данных.

Представление позволяет увидеть результаты сохранённого запроса таким образом, как будто это полноценная таблица БД.

Синтаксис создания представления:

```
CREATE
    [OR REPLACE]
    [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
    [DEFINER = { user | CURRENT_USER }]
    [SQL SECURITY { DEFINER | INVOKER }]
    VIEW view_name [(column_list)]
    AS select_statement
    [WITH [CASCADED | LOCAL] CHECK OPTION]
```

---

**Все примеры и задания представлены на основе БД students из ЛР3.**

### Пример 1:

Создание **вертикального** представления. Суть состоит в выделении некоторого подмножества столбцов таблицы.

```
create view stdinfo (name,surname)
AS
select std_name,std_srn
from studs
```

Мы выделили в представление столбцы таблицы studs, в которых храниться имя и фамилия.

К полученному представлению можно обращаться как к полноценной таблице БД (**но оно не является таковой!**)

```
select * from stdinfo where name like('O%')
```

Получаем всех студентов, имена которых начинаются с буквы «О».

### Пример 2:

Аналогично можем создать и *горизонтальное* представление. Не сложно догадаться, что оно выделяет некоторое подмножество строк таблицы.





```
create view stddep
as
select * from studs
where dep_id = 1
```

### Пример 3:

Попробуем создать представление сложнее. Я добавил в БД данные об оценках на 3-х экзаменах. Между нами говоря, это должны были сделать и вы. Создадим представление для отображения оценок каждого студента и среднего балла в удобном виде.

```
create view examlook (name,surname,algebra,geom,matan,average)
AS
select s.std_name,s.std_srn,ae.mark,ge.mark ,me.mark,(me.mark+ae.mark+ge.mark)/3
from studs as s,algeexam as ae,geomexam as ge,matanexam as me
where s.std_id=ae.std_id and s.std_id=ge.std_id and s.std_id=me.std_id
```

После применения оператора SELECT \* FROM examlook, видим результат.

Result Grid			 Filter Rows:	Export:		Wrap Cell Content:	
	name	surname	algebra	geom	matan	average	
▶	Alexandra	Admiralova	10	10	9	9.6667	
	Olga	Shimko	9	7	6	7.3333	
	Alex	Kushniarou	10	9	9	9.3333	
	Konstantin	Sachhinsky	3	5	6	4.6667	
	Alexandr	Kochurka	8	9	8	8.3333	
	Anton	Grigoriev	4	2	9	5.0000	
	Dzheki	Chan	7	2	6	5.0000	
	Julia	Kots	6	3	9	6.0000	
	Olga	Kundas	4	3	5	4.0000	
	Maksim	Zhur	6	6	0	4.0000	

Индекс – средство для повышения производительности БД при поиске выборки данных.

Индексация позволяет повысить скорость поиска в таблицах БД. Индексируемое поле сервер сортирует в фоновом режиме, представляя его в виде дерева, например. Это простое решение позволяет добиться значительного повышения производительности при запросах.

НО! Создание индекса значительно нагружает память, а также замедляет операции INSERT и UPDATE. **Золотое правило: «7 раз подумай – один раз проиндексируй!».** Индексировать стоит лишь те поля по которым вероятнее всего будет идти поиск.

#### Пример 4:

Рассмотрим простой запрос к таблице students. Логика поиска можно увидеть с помощью команды explain.

```
explain select * from studs
where std_name='Olga'
```

Видим, что индексов для поля имя в таблице нет, и сервер вынужден обработать все записи таблицы.

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	studs	NULL	ALL	NULL	NULL	NULL	NULL	11	10.00	Using where

Создадим простой индекс по первым 4 символам поля имя в таблице студенты.

```
create index myind ON studs (std_name(4))
```

После создания индекса видим, что СУБД обработала только 2 записи таблицы.

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	studs	NULL	ref	myind	myind	15	const	2	100.00	Using where

#### Задания для самостоятельной работы

1. Загрузите в СУБД MySQL базу данных students, доработанную вами в ЛР3.
2. Создайте следующие представления:
  1. Представление для просмотра списка всех студентов обучающихся на кафедре дифференциальных уравнений.
  2. Представление для просмотра всех студентов, которые сдавали экзамен по геометрии.
  3. Представление для просмотра среднего балла на всех экзаменах по кафедрам. Используя, данное представление, создайте запрос, выводящий все кафедры, на которых более половины экзаменов сданы со средним баллом более 7.
  4. Представление отображающую информацию о среднем балле на экзамене по физкультуре у девочек и мальчиков.
  5. Представление для отображения зависимости среднего балла студентов от этажа, на котором расположена кафедра. Используя данное представление и запрос на выборку, выведите список студентов, у которых оценки по 2 экзаменам превосходят среднюю оценку по этажу.
  6. Представление для отображения процента студентов, не сдавших сессию по каждой кафедре.
  7. Представление для отображения всех оценок на экзамене для каждого студента, а также размера стипендии.
3. Внимательно изучите вашу БД. Создайте необходимые для поиска индексы. Подумайте, для каких полей нужно выбрать уникальный индекс, а для каких таблиц выгодно использовать составной индекс. Продемонстрируйте работу запросов на выборку с индексами и без. Попробуйте создать индекс для некоторого представления. Объясните результат.
4. Продолжите разработку БД из ЛР1. Добавьте таблиц, для расширения функциональности (выделено жёлтым цветом ниже, имеете полное право добавить ещё что либо по усмотрению). Заполните таблицы. Добавьте ограничения на внешние ключи. Создайте необходимые на ваш взгляд хранимые процедуры и триггеры.

Предусмотрите все необходимые представления и индексы. Будьте готовы вывести любую информацию по средствам представлений и SQL – запросов.

Вариант	База данных	Детали
1	Школа	Учителя, ученики, успеваемость, посещаемость, кружки, олимпиады.
2	Пиццерия	Продукты, пиццы, заказы, доставщики, повара.
3	Футбольная команда	Игроки, игры, тренировочный процесс, тренеры, соревнования (минимум 2), статистика выступления игроков.
4	Банк	Вклады, клиенты, операции, кредиты, кредитная история, сотрудники.
5	ММФ	Преподаватели, кафедры, предметы, экзамены, курсы, стипендия, кружки, секции, участие в деятельности факультета, СС и ТС.
6	Магазин одежды	Ассортимент, цены, продажи, поставки, бренды, сеть магазинов.
7	Больница	Пациенты, запись на приём к врачу, врачи, список платных услуг, финансовая отчётность, родильное отделение.