

**Лабораторный практикум 12**  
**Тема: Основы оптимизации запросов. Часть 2**

**Задания**

**Этап 1 Подготовка базы данных**

1. Выполнить скрипт `hr_create.sql` с командами создания таблиц БД.
2. Выполнить скрипт `hr_popul.sql` с командами заполнения таблиц БД.

**Этап 2 Изучение структуры таблиц статистики**

2.1 Используя генератор первичного заполнения БД, внести в отношения `employees` 100000 строк с нормальным распределением следующих случайных величин:

- а) идентификатор подразделения в допустимом диапазоне идентификаторов подразделений (использовать предварительно созданную коллекцию);
- б) идентификатор должности в допустимом диапазоне идентификаторов, используя порядковый номер получаемой должности из списка должностей (использовать предварительно созданную коллекцию);
- в) зарплата из диапазона 1500 до 5000 с шагом 100;
- г) идентификатор менеджера из диапазона от 1 до 100-го сотрудника, которые уже были зарегистрированы в БД до выполнения задания 1.

2.2 Провести сбор статистики (полный сбор, частичный с процентом числа строк = 10 %) получить содержание статистики по таблицам, колонкам и содержимому гистограмм.

**Этап 3 Анализ эффективности использования индексов**

3.1 Создать запросы на получение фактора селективности всех атрибутов таблицы `employees`. Все атрибуты разделить на две группы:

- 1) фактор селективности  $\geq 10\%$ , 2) фактор селективности  $< 10\%$ .

3.2 Создать SQL-запросы к таблице `employees` по условиям создания WHERE-фразы, в которые входят атрибуты каждой из указанных групп фактора селективности:

- 1) атрибут целого типа = значение;
- 2) атрибут целого типа > значение;
- 3) атрибут целого типа < значение;
- 4) атрибут целого типа в диапазоне значений;
- 5) атрибут строкового типа = значение;
- 6) атрибут строкового типа like значение;
- 7) атрибут типа «дата» в диапазоне значений.

3.3 Определить план выполнения созданных в 3.2 запросов с помощью команды `EXPLAIN` или `SET AUTOTRACE ON`. Для каждого плана определить методы доступа.

3.4 С учетом рекомендаций стратегий оптимизации по стоимости создать индексы для запросов, которые спроектированы в пункте 3.2.

3.5 Повторить пункт 3.3, но при наличии индексов.

3.6 Сравнить значения *COST* для однотипных запросов при отсутствии и наличии индексов.

**Этап 4. Анализ и управление сложными запросами**

4.1 В решениях лабораторной работы № 6:

- выбрать запросы, включающие более двух таблиц во *FROM*-фразе;
- используя команды управления методами доступа к данным, установить порядок связи типа *ORDERED* и *STAR*;
- для полученных пар запросов определить план их выполнения и сравнить значения *COST*.

4.2 С учетом рекомендаций по использованию индексов создать индексы для запросов, которые получены в пункте 4.1.

4.3 Для полученных пар запросов отдельно с учетом связи типа *ORDERED* и *STAR* определить план их выполнения и сравнить значения *COST*.

4.4 В решениях лабораторной работы № 6:

- выбрать запросы, включающие операторы *IN*, *NOT IN*, *EXISTS*, *NOT EXISTS*;

- выполнить эквивалентную замену операторов (*IN* на *EXISTS*, *EXISTS* на *IN*), обеспечивающий тот же ответ на запрос;
- для полученных пар запросов определить план их выполнения и сравнить значения *COST*.

4.5 С учетом рекомендаций по использованию индексов создать индексы для запросов, которые получены в пункте 4.4.

4.6 Для полученных пар запросов отдельно с учетом индекса и без индекса определить план их выполнения и сравнить значения *COST*.

### **Требования к оформлению отчета решений по лабораторной работе**

Создайте файл `Фамилия_12.sql`, в который включите команды:

- выполняемые во всех пунктах заданий
- физ.планы выполнения запросов.