Рекомендации по оптимизации

Создавайте вспомогательные структуры в базе с учетом специфики запросов к данным.
Пишите так, чтобы помочь оптимизатору запросов.

1. Индексы не будут использованы

СПбГУ

- WHERE COL1 > COL2
- WHERE COL1 < COL2
- WHERE COL1 >= COL2
- WHERE COL1 <= COL2
- WHERE COL1 IS NOT NULL
- WHERE COL1 NOT IN (value1, value2)
- WHERE COL1 != expression
- WHERE COL1 LIKE '%patern'

2. Не используйте выражения от индексных столбцов

```
SELECT DEPT_NAME FROM DEPARTMENT WHERE
UPPER(DEPT_NAME) like 'SALES%');
```

3. Для фильтрации записей используйте WHERE, а не HAVING

- **Запрос без индекса:**
  SELECT DEPTID, SUM(SALARY)
  FROM EMP
  GROUP BY DEPTID
  HAVING DEPTID = 100;
- **Запрос с индексом:**
  SELECT DEPTID, SUM(SALARY)
  FROM EMP
  WHERE DEPTID = 100
  GROUP BY DEPTID;

4. Для использования индекса по нескольким столбцам указывайте в разделе WHERE начальные столбцы ключа индекса

СПбГУ

- **Индекс:** Index(PART_NUM, PRODUCT_ID)
- **Запрос с использованием индекса:**
  SELECT * FROM PARTS WHERE PART_NUM = 100;
- **Запрос без использования индекса:**
  SELECT * FROM PARTS WHERE PRODUCT_ID = 5555;

5. Не стройте индексы для маленьких таблиц

СПбГУ

- Если таблица небольшого размера, то индексы не нужны.
- При выборе из таблицы более 15-25% строк полный просмотр быстрее, чем сканирование через индекс.

6. Стройте индексы для полей, по которым производится сортировка записи - тогда оптимизатор будет использовать индексное сканирование


```
SELECT SALARY FROM EMP
ORDER BY EMPID;
```

7. Минимизируйте число просмотров таблиц

СПбГУ

- **Два просмотра таблицы:**
  SELECT *
  FROM STUDENT WHERE GroupNumber = 341
  UNION
  SELECT *
  FROM STUDENT WHERE GroupNumber = 441
- **Один просмотр таблицы:**
  SELECT *
  FROM STUDENT WHERE GroupNumber = 341
  OR GroupNumber =441)

8. Соединяйте таблицы в правильном порядке

СПбГУ

При соединении таблиц вначале указывайте таблицу с меньшим количеством строк.

9. Используйте более простые запросы

СПбГУ

- Много мелких запросов в цикле лучше объединить в один большой.
- Чтобы уменьшить количество вложенных запросов, можно делать промежуточные выборки.

10. IN, EXISTS или JOIN?

СПбГУ

- Если в основной выборке много строк,
  а в подзапросе мало, то лучше использовать **IN**.
- Если в подзапросе сложный запрос, а в основной выборке относительно мало строк, то лучше использовать **EXISTS**.
- Если оба запрос сложные, надо использовать **JOIN**.

11. Группировка

Если после группировки надо отсортировать результат, то желательно, чтобы поля сортировки и поля группировки перечислялись в одном порядке.

12. Используйте результаты работы оптимизатора повторно

СПбГУ

- В оперативной памяти хранятся все результаты ранее выполненных запросов до тех пор, пока эта память не потребуется для записи результатов последующих запросов.
- По возможности составляйте запросы таким образом, чтобы они совпадали с написанными ранее.