Задание 2

Предложить правки запросов без модификации структуры БД такие, что время выполнения запроса получения 3 000 заказов из 50 000 со средним количеством элементов в заказе равным 3 не будет привышать 1-2 сек.

Для выполнения задания используется **Gpt 4.0.**

|  |  |
| --- | --- |
| Promt | Code |
| Optimize the SQL script | SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE FUNCTION [dbo].[F\_WORKS\_LIST] ()  RETURNS TABLE  AS  RETURN  (  SELECT  w.Id\_Work,  w.CREATE\_Date,  w.MaterialNumber,  w.IS\_Complit,  w.FIO,  CONVERT(VARCHAR(10), w.CREATE\_Date, 104) AS D\_DATE,  dbo.F\_WORKITEMS\_COUNT\_BY\_ID\_WORK(w.Id\_Work, 0) AS WorkItemsNotComplit,  dbo.F\_WORKITEMS\_COUNT\_BY\_ID\_WORK(w.Id\_Work, 1) AS WorkItemsComplit,  dbo.F\_EMPLOYEE\_FULLNAME(w.Id\_Employee) AS EmployeeFullName,  w.StatusId,  ws.StatusName,  CASE  WHEN w.Print\_Date IS NOT NULL OR w.SendToClientDate IS NOT NULL OR w.SendToDoctorDate IS NOT NULL OR w.SendToOrgDate IS NOT NULL OR w.SendToFax IS NOT NULL  THEN 1  ELSE 0  END AS Is\_Print  FROM Works w  LEFT JOIN WorkStatus ws ON w.StatusId = ws.StatusID  WHERE w.IS\_DEL <> 1  ORDER BY w.Id\_Work DESC -- Or w.MaterialNumber DESC if needed  );  GO |
| Optimize the sql script if result = 3,000 & out = 50,000 | SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE FUNCTION [dbo].[F\_WORKS\_LIST] ()  RETURNS TABLE  AS  RETURN  (  SELECT TOP 3000  w.Id\_Work,  w.CREATE\_Date,  w.MaterialNumber,  w.IS\_Complit,  w.FIO,  CONVERT(VARCHAR(10), w.CREATE\_Date, 104) AS D\_DATE,  (SELECT COUNT(\*) FROM WorkItems wi WHERE wi.Id\_Work = w.Id\_Work AND wi.IsComplit = 0) AS WorkItemsNotComplit,  (SELECT COUNT(\*) FROM WorkItems wi WHERE wi.Id\_Work = w.Id\_Work AND wi.IsComplit = 1) AS WorkItemsComplit,  dbo.F\_EMPLOYEE\_FULLNAME(w.Id\_Employee) AS EmployeeFullName,  w.StatusId,  ws.StatusName,  CASE  WHEN w.Print\_Date IS NOT NULL OR w.SendToClientDate IS NOT NULL OR w.SendToDoctorDate IS NOT NULL OR w.SendToOrgDate IS NOT NULL OR w.SendToFax IS NOT NULL  THEN 1  ELSE 0  END AS Is\_Print  FROM Works w  LEFT JOIN WorkStatus ws ON w.StatusId = ws.StatusID  WHERE w.IS\_DEL <> 1  ORDER BY w.Id\_Work DESC  );  GO |

При уточнении запроса (добавления условия: if result = 3,000 & out = 50,000) качество сгенерированного скрипта повысилось. Как минимум было добавлено ограничение: TOP 3000, что должно значительно повысить скорость выполнения запроса.

Еще одно отличие: dbo.F\_WORKITEMS\_COUNT\_BY\_ID\_WORK(w.Id\_Work, 0)

заменено на подзапрос:

(SELECT COUNT(\*) FROM WorkItems wi WHERE wi.Id\_Work = w.Id\_Work AND wi.IsComplit = 0) AS WorkItemsNotComplit.

Примечание: наверное, стоит убрать сортировку, так как это точно влияет на производительность. Если пользователю понадобится сортировать эти данные, он сможет сделать это отдельным запросом. Тем самым мы избавимся от лишнего перебора значений в тех случаях, когда в этом нет необходимости.

**Объяснение изменений:**

* Предложение TOP 3000: ограничивает результирующий набор желаемым количеством строк.
* GROUP BY и SUM вместо отдельных подзапросов: это более эффективный подход, особенно когда среднее количество рабочих элементов в работе относительно невелико.
* Объединение рабочих элементов: объединяем таблицу Works с рабочими элементами, используя id\_work для вычисления количества непосредственно в запросе.

**Как работает эта оптимизация:**

**Эффективная агрегация данных**: Предложение GROUP BY эффективно группирует рабочие элементы, связанные с каждой рабочей записью, а функции суммирования с выражениями CASE предоставляют краткий способ вычисления желаемых значений.

**Сокращение количества запросов**: объединение логики подсчета в одном запросе устраняет необходимость в нескольких запросах или вызовах функций, что приводит к меньшему количеству обращений к базе данных и потенциально более высокой производительности.

**Правильная группировка**: GROUP BY гарантирует, что значения для WorkItemsNotComplit и WorkItemsComplit будут рассчитаны правильно для каждой рабочей записи.