frlily / September 19, 2019 10:35PM

20條Tips：高性能SQL查詢，優化取數速度方案

在數據這個領域摸爬滾打的人，取數往往是數據生涯的開始，而 SQL是取數的基本功。如果你已經有了資料庫知識基礎，那麼只需要做一些SQL學習和練習，很快就會提高水平。接著你想提高sql水平，就需要學習一些比較高級的用法和技巧了。普遍會遇到的一個就是訪問速度，如果你發現數據訪問的速度很慢，你可能需要考慮是不是SQL語句寫的有問題，需不需要優化。

[size=x-large]為什麼要做SQL優化[/size]

[url=http://www.finereport.com/tw/]報表[/url]的核心是數據，數據集是否合理決定報表的質量。

1.每張報表都應該有一個主數據集，為了降低維護時的工作量，盡量將所有欄位置於主數據集，除非在某些情況下，不使用多源數據集會導致主數據集異常複雜。

2.在製作報表之前，盡量考慮到所有需要展示的數據欄位，在資料庫軟體中，合理編寫sql語句，[url=http://www.finereport.com/tw/]大數據分析[/url]情況盡量對sql做優化，以及添加索引。

擁有高性能SQL查詢語句，能使查詢速度加快，報表展示速度得到較明顯的提升！

[size=large]20條Tips方案，優化你的SQL查詢[/size]

1 SELECT子句中避免使用 「*」

當你想在SELECT子句中列出所有的COLUMN時，使用動態SQL列引用『*』是一個方便的方法。不幸的是，這是一個非常低效的方法。 實際上，ORACLE在解析的過程中, 會將「*」 依次轉換成所有的列名,

這個工作是通過查詢數據字典完成的, 這意味著將耗費更多的時間。

2 刪除重複記錄

最高效的刪除重複記錄方法 (因為使用了ROWID)

DELETE FROM EMP E WHERE E.ROWID > (SELECT MIN(X.ROWID) FROM EMP X WHERE X.EMP_NO = E.EMP_NO)

3 用TRUNCATE替代DELETE

當刪除表中的記錄時，在通常情況下，回滾段(rollback segments) 用來存放可以被恢復的信息，如果你沒有COMMIT事務，ORACLE會將數據恢復到刪除之前的狀態(準確地說是恢復到執行刪除命令之前的狀況)，而當運用TRUNCATE時，回滾段不再存放任何可被恢復的信息。當命令運行後,數據不能被恢復。因此很少的資源被調用,執行時間也會很短。

4 計算記錄條數

和一般的觀點相反， count(*) 比count(1)稍快 ，當然如果可以通過索引檢索，對索引列的計數仍舊是最快的。

例如COUNT(EMPNO)

5 用EXISTS替代IN

在許多基於基礎表的查詢中，為了滿足一個條件，往往需要對另一個表進行聯接。在這種情況下，使用EXISTS(或NOT EXISTS)通常將提高查詢的效率。

–低效

SELECT * FROM EMP WHERE EMPNO > 0 AND DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE LOC =』MELB』)

–高效

SELECT * FROM EMP WHERE EMPNO > 0 AND EXISTS (SELECT 『X』 FROM DEPT WHERE DEPT.DEPTNO = EMP.DEPTNO AND LOC = 『MELB』)

6 用EXISTS替換DISTINCT

當提交一個包含一對多表資料(比如部門表和僱員表)的查詢時,避免在SELECT子句中使用DISTINCT.

一般可以考慮用EXIST替換 例如: Sql程式碼

–低效:

SELECT DISTINCT DEPT_NO,DEPT_NAME FROM DEPT D,EMP E WHERE D.DEPT_NO = E.DEPT_NO

–高效:

SELECT DEPT_NO,DEPT_NAME FROM DEPT D WHERE EXISTS (SELECT 『X』 FROM EMP E WHERE E.DEPT_NO = D.DEPT_NO)

EXISTS 使查詢更為迅速

7 用>=替代>

如果DEPTNO上有一個索引

–高效：

SELECT * FROM EMP WHERE DEPTNO >=4

–低效：

SELECT * FROM EMP WHERE DEPTNO >3

兩者的區別在於, 前者DBMS將直接跳到第一個DEPT等於4的記錄而後者將首先定位到DEPTNO=3的記錄並且向前掃

描到第一個DEPT大於3的記錄.
8 應盡量避免在 where 子句中對欄位判斷!
如：
select id from t where num is null
可以在num上設置默認值0，確保表中num列沒有null值，然後這樣查詢：
select id from t where num=0
9 應避免在 where 子句中使用!=或操作符!
將引擎放棄使用索引而進行全表掃描。優化器將無法通過索引來確定將要命中的行數,因此需要搜索該表的所有行。
10 應避免在 where 子句中使用 or 連接!
否則將導致引擎放棄使用索引而進行全表掃描，如：
select id from t where num=10 or num=20
可以這樣查詢：
select id from t where num=10 union all select id from t where num=20
11 in 和 not in 也要慎用
因為IN會使系統無法使用索引,而只能直接搜索表中的數據。如：
select id from t where num in(1,2,3)
對於連續的數值，能用 between 就不要用 in 了：
select id from t where num between 1 and 3
12 應避免在 where 子句中進行表達式操作
這將導致引擎放棄使用索引而進行全表掃描。如：
SELECT * FROM T1 WHERE F1/2=100
應改為:
SELECT * FROM T1 WHERE F1=100*2
SELECT * FROM RECORD WHERE SUBSTRING(CARD_NO,1,4)=』5378』
應改為:
SELECT * FROM RECORD WHERE CARD_NO LIKE 『5378%』
SELECT member_number, first_name, last_name FROM members WHERE DATEDIFF(yy,datofbirth,GETDATE()) > 21
應改為:
SELECT member_number, first_name, last_name FROM members WHERE dateofbirth
即：任何對列的操作都將導致表掃描，它包括資料庫函數、計算表達式等等，查詢時要儘可能將操作移至等號右邊。
13 應避免在where子句中進行函數操作
這將導致引擎放棄使用索引而進行全表掃描。如：
select id from t where substring(name,1,3)='abc'
--name以abc開頭的id
select id from t where datediff(day,createdate,'2005-11-30')=0
--『2005-11-30』生成的id
應改為:
select id from t where name like 'abc%'
select id from t where createdate>=』2005-11-30′ and createdate 60000
在這條語句中,如salary欄位是money型的,則優化器很難對其進行優化,因為60000是個整型數。我們應當在編程時將整型轉化成為錢幣型,而不要等到運行時轉化。
17 充分利用連接條件
在某種情況下，兩個表之間可能不只一個的連接條件，這時在 WHERE
子句中將連接條件完整的寫上，有可能大大提高查詢速度。
例：
SELECT SUM(A.AMOUNT) FROM ACCOUNT A,CARD B WHERE A.CARD_NO = B.CARD_NO
SELECT SUM(A.AMOUNT) FROM ACCOUNT A,CARD B WHERE A.CARD_NO = B.CARD_NO AND A.ACCOUNT_NO=B.ACCOUNT_NO
第二句將比第一句執行快得多。
18 能用DISTINCT的就不用GROUP BY
SELECT OrderID FROM Details WHERE UnitPrice > 10 GROUP BY OrderID
可改為：
SELECT DISTINCT OrderID FROM Details WHERE UnitPrice > 10
19 能用UNION ALL就不要用UNION！

UNION ALL不執行SELECT DISTINCT函數，這樣就會減少很多不必要的資源
20 盡量不要用SELECT INTO語句！
SELECT INTO 語句會導致表鎖定，阻止其他使用者訪問該表。
如果你有更好的方法，可以來finereport幫助文檔中與其他使用者分享：
https://help.finereport.com/doc-view-1813.html
初學者需要學SQL，推薦這篇：
[url=http://www.finereport.com/tw/knowledge/acquire/sql-3.html]零基礎快速自學SQL，1天足矣—附最全SQL學習資源和練習題！[/url]

喜歡這篇文章嗎？歡迎分享按讚，給予我們支持和鼓勵！