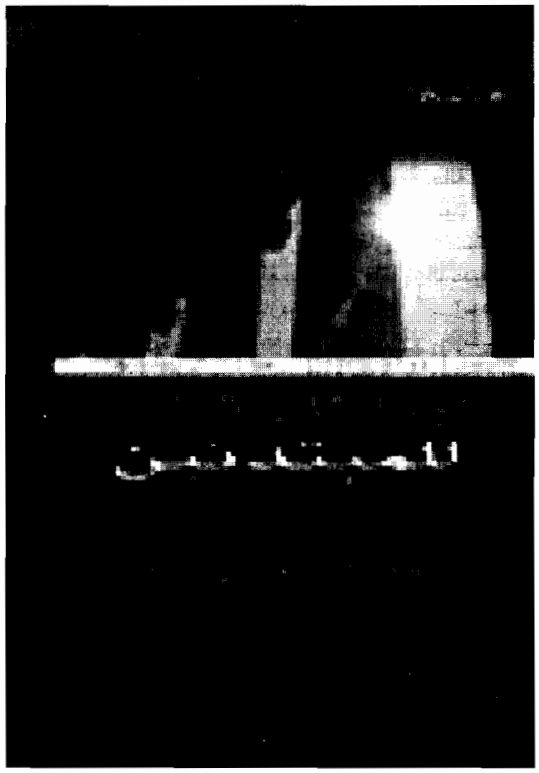
دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	أوراكل 9 آى (لمديري قواعد البيانات)
المصدر:	الاقتصاد والمحاسبة - مصر
المؤلف الرئيسي:	عبدالحميد، وليد
المجلد/العدد:	ع 614
محكمة:	نعم
التاريخ الميلادي:	2005
الشهر:	أكتوبر
الصفحات:	22 - 24
رقم MD:	92608
نوع المحتوى:	بحوث ومقالات
قواعد المعلومات:	ACI, EcoLink
مواضيع:	تخزين المعلومات، برنامج أوراكل ، لغات البرمجة، قواعد البيانات، البرمجيات، الحاسبات الإلكترونية، استرجاع المعلومات
رابط:	http://search.mandumah.com/Record/92608

© 2016 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

أوراكل ٩ آى (لمديرى قواعد البيانات)



إعداد :
وليد عبدالحميد

كما هو الحال بالنسبة لتصميم التطبيق، قد يبدو إعداد جمل SQL ليس من واجبات مديرى قواعد البيانات. ولكن الحقيقة أن مديرى قواعد البيانات يجب أن يشتركوا فى مراجعة SQL التى تمت كتابتها كجزء من التطبيق. فمن الممكن أن يعانى تطبيق مصمم جيدا من مشكلات فى الأداء إذا كان SQL التى يستخدمها تم إعدادها بشكل غير صحيح. فمن الممكن أن تتسبب مشكلات تصميم التطبيق وSQL فى أغلب مشكلات الأداء ففى قواعد البيانات المصممة جيدا.

إن أهم ما فى إعداد SQL هو تقليل مسار البحث الذى تستخدمه قاعدة البيانات فى إيجاد البيانات. وفى أغلب جداول oracle، يكون لكل صفRowld مرتبط به. ويشتمل على معلومات عن الموضع الفعلىRowld للصف ـ الملف الخاص به و block الموجود بهذا الملف والصف الذى بداخل block قاعدة البيانات.

وعندما يتم تنفيذ استعلام بدون عبارة where ، فعادة ما تقوم قاعدة البياناتباستعراض كامل للجدول، مع قراءة كل فى الجدول. وفى أثناء استعراضblock أى جدول كامل، تحدد قاعد البيانات موضع أول block فى الجدول. وبالنسبة للجداول الكبيرة، من الممكن أن تستغرق عمليات استعراض جداول بأكملها الكثير من الوقت. ويقلل تشغيل الاستعلام بشكل متواز من هذا الوقت.

وعندما يتم استعلام صفوف معينة، من الممكن أن تستخدم قاعدة البيانات فهرسا لجعل عملية إنتاج الصفوف المطلوبة أسرع. ويقوم الفهرس بتخصيص قيم اعتبارية فى جدول لـRowIDs الخاصة بها، والتى تقوم بدورها بتعيينها لموضع فعلى محدد. ومن الممكن أن تكون الفهارس مميزة ـ والتى لا تكرر فيها أية قيمة أكثر من مرة ـ أو غير مميزة، وتخزن الفهارس فقط RowIDs لقيم العمود فى الأعمدة لقيم العمود فى الأعمدة المفهرسة. ولا يتم تخزين أى شىء فى الفهرس إذا كان العمود خاليا بالنسبة لصف معين.

ومن الممكن أن تقوم بفهرسة أعمدة متعددة سويا. ويسمى هذا الفهرس بفهرس ربط وسيستخدم إذا كان أول عمود له يستخدم فى عبارةwhere الخاصة بالاستعلام. وفى oraclegi ، يمكن أن تستخدم أداة رفع الكفاءة طريقة تجاوز الاستعراضskip.scan يتم فيها استخدام فهرس حتى إذا لم يكن العمود الأول الخاص به فى عبارة where الخاصة بالاستعلام. وهذه الطريقة ليست فعالة كتلك التى تستخدم العمود الأول للفهرس.

يجب ضبط الفهارس حسب مسار الوصول المطلوب. افترض أن هناك فهرس ربط مكونا من ثلاثة أعمدة. وكما هو موضح فى الكود التالى، تم إنشاء هذا الفهرس على أعمدة zip و city و state الخاصة بجدول EMPLOYEE :

```
create index city st zlp ndx
on EMPLOYEEE (city stae
zip)
tableespace INDEXES
```

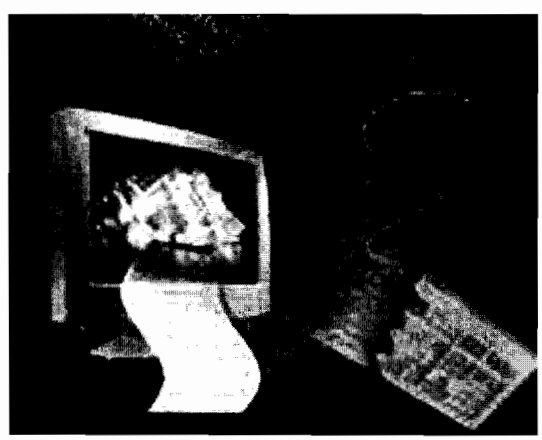
وإذا تم تنفيذ الاستعلام التالى:

```
select * from EMPLOYEE
where state NJ
```

لن يمكن استخدام الفهرس فى oraclc8i، وذلك لأن العمود الأول الخاص به (city) لم يتم استخدامه فى عبارة where. وفى ora-cle9i، تسمح إمكانية تجاوز الاستعراض الخاصة بأداة رفع الكفاءة باستخدام city st zlP NDX للوفاء باحتياجات هذا الاستعلام. وإذا كان المستخدمون يقومون بشكل متكرر بتنفيذ هذا النوع من الاستعلامات، فيجب إعادة ترتيب أعمدة الفهرس بوضع State فى البداية حتى يتم عكس نموذج الاستخدام الفعلى.

من المهم أن يتم ترتيب بيانات الجداول على قدر الإمكان. فإذا كان المستخدمون ينفذون استعلامات نطاق بشكل متكرر ـ مع تحديد القيم المكتوبة فى نطاق معين، فمن الممكن أن يحتاج ترتيب البيانات إلى التعرف على blocks بيانات أقل فى أثناء تحليل الاستعلام، مما يؤدى إلى تحسين الأداء. وستشير الادخارات المرتبة فى الفهرس إلى مجموعة من blocks المتجاورة فى الجدول وليس إلى blocks متفرقة داخل ملف (ملفات) البيانات.

فعلى سبيل المثال، سيحتاج استعلام النطاق التالى:



```
select *
from EMPLOYEE
where Empno between 1 and 100;
```

إلى التعرف على blosks بيانات أقل إذا تم ترتيب السجلات الفعلية الموجودة فى جدول EMPLOYEE حسب عمود Empno. ويجب أن يحسن هذا من أداء الاستعلام. ولكى تضمن ترتيب الصفوف بشكل صحيح فى الجدول، قم بنقل السجلات إلى ملف أساسى ثم فرزها فى الملف وحذف السجلات القديمة وإعادة تحميلها من الملف الذى تم فرزه.

وهناك بديل لنقل التعرف على البيانات إلى ملف أساسى وهو استخدام إجراءات الفرز الداخلية الخاصة بـ Oracle فى فرز البيانات الخاصة بك. فيمكنك إعادة ترتيب صفوف لجدول بإنشاء جدول آخر، باستخدام أمر create table as select. وفى الإصدارات السابقة لـOracle9i، كان أمرا insert as select create table as select لا يسمحان لك بتحديد عبارة order by.

وللتغلب على هذا النقص فى الإصدارات السابقة لـoracle8i ، قم بإنشاء عرض على الجدول الأساسى. فالعروض تدعم الآن عبارة order by، وبالتالى يمكنك إنشاء عرض والاختيار منه لملء جدول مرتب.

```
create or replace view EMPLOYEE
VIEW as
    select * from company
    order by Empno
```

ويمكنك الآن إنشاء جدول بالاختيار من EMPLOYEE VIEW; وسيكون الناتج إنشاء نسخة مكررة من EMPLOYEE ، مع فرز صفوفه بشكل صحيح.

```
create * table EMPLOYEE ORDERED
    as select from EMPLOYEE VIEW
```

فى المثال السابق، تم ترتيب البيانات حسب قيمة Empno. إنك عادة ما تحتاج لترتيب البيانات حسب عمود السمة، على سبيل المثال، عمود Name. وإذا تم ترتيب البيانات لدعم استعلامات النطاق الأكثر استخداما وتخزينها داخل كلblok يمكنك تقليل عدد blocks التى تتم قراءتها فى أثناء كل استعلام، مما يؤدى بالتالى إلى تحسين أداء استعلاماتك.

تأثير الترتيب على معدلات التحميل

تؤثر الفهارس على أداء الاستعلامات وعمليات تحميل البيانات. ففى أثناء عمليات الإدراج، يكون لترتيب الصفوف تأثير ملحوظ على أداء التحميل. وحتى فى البيئات التى تستخدم فيها الفهارس بشكل كبير، من الممكن أن يؤثر الترتيب الصحيح للصفوف قبل عملية الإدراج على أداء التحميل بنسبة ٥٠٪.

وكلما يزيد حجم أى فهرس، يقوم Oracls بتخصيص blocks جديدة. وإذا تمت إضافة إدخال فهرس جديد بعد آخر إدخال، ستتم إضافة الإدخال الجديد إلى آخر block فى الفهرس. وإذا تسبب الإدخال الجديد فى زيادة Oracle عن المساحة المتاحة فى هذا block ، سيتم نقل الإدخال إلى block جديد، مع عدم تغيير كل الإدخالات الأصلية. وهناك تأثير قليل جدا على الأداء بسبب هذا التخصيص.

وإذا لم يتم ترتيب الصفوف المدرجة، سيتم نسخ إدخالات الفهرس الجديدة لـblocks تفرع الفهرس الحالى. وإذا لم تكن هناك مساحة إضافية في التفرع الذى تمت فيه إضافة القيمة الجديدة، سيتم تقسيم التفرع إلى جزأين. سيترك ٥٠٪ من إدخالات الفهرس فى التفرع الأصلى و٥٠٪ الأخرى سيتم نقلها إلى تفرع جديد. ونتيجة لذلك، سيعانى الأداء فى أثناء عمليات التحميل (بسبب نشاط إدارة المساحة الإضافى) وفى أثناء الاستعلامات (حيث إن الفهرس يشتمل على مساحة كبيرة غير مستخدمة، مما يتطلب التعرف على blocks أكثر).

وفى اختبارات قياس الأداء، تم إنشاء فهرس لعمود Name الخاص بجدول EM-PLOYEE. وعندما تم إدراج الصفوف، مع عدم ترتيب قيم Name، يزيد عدد blocks التى يستخدمها الفهرس بنسبة ٥٠٪ ويقل معدل التحميل بأكثر من ٥٠٪ . وإذا كان للجدول الخاص بك فهارس متعددة، قم بترتيب الصفوف فى أثناء عمليات الإدراج حسب أكثر الفهارس التى يتم استخدامها بشكل متكرر فى أثناء استعلامات النطاق إذا كنت ترغب فى إعداد عمليات الاستعلام، لإعداد عمليات الإدراج والترتيب حسب أكثر الفهارس تعقيدا.

ملحوظة

سيحدث انخفاض كبير فى أداء التحميل عندما يزيد أى فهرس من عدد المستويات الداخلية الخاصة به. ويعتمد الوقت الذى تحدث فيه الزيادة على حجم block قاعدة البيانات وطول قيم الفهرس الأساسية. ولكى تعرض عدد المستويات، قم بتحليل فهرس ثم تحديد قيمة عمود Blevl الخاص به من DBA INDEXES . وللوصول لأفضل أداء للتحميل، اجعل قيم مستويات Blevl قليلة بقدر الإمكان.

بسبب الطريقة التى تستخدمها Oracle فى إدارة فهارسها داخليا، معدلات التحميل فى كل مرة تتم فيها إضافة فهرس جديد (حيث من غير المحتمل أن يتم إدراج الصفوف مع فرز بيانات بشكل صحيح لأعمدة متعددة). فمعدل التحميل يفضل التعامل مع الفهارس المكونة من أعمدة متعددة عن فهارس متعددة مكونة من عمود واحد.

الخيارات الإضافية الخاصة بالفهرسة

إذا كانت البيانات غير مميزة بشكل كبير، فمن الممكن أن تفكر فى استخدام فهارس نقطية. وتعد الفهارس النقطية فعالة جدا بالنسبة للاستعلامات التى تتم على مجموعات بيانات كبيرة وثابتة من خلال قيم قليلة مختلفة. ويمكنك إنشاء فهارس نقطية وقياسية للجدول نفسه وستقوم Or-acle بأى تحويلات ضرورية خاصة بالفهرس ديناميكيا فى أثناء معالجة الاستعلامات. ولا يمكن أن تنشىء فهرس نقطى وقياسى للعمود ذاته داخل أحد الجداول. انظر الفصل الخامس عشر لمزيد من التفاصيل عن استخدام الفهارس النقطية.

وإذا كان يتم استعلام جدولين معا بشكل متكرر، فمن الممكن أن يكون استخدام التجميعات فعالا فى تحسين الأداء. فالتجميعات تخزن صفوفا من جداول متعددة فى blocks البيانات الفعلية ذاتها، وذلك اعتمادا على قيمها المنطقية cluster key. ارجع إلى الفصل الخامس لمزيد من المعلومات عن التجميعات.

وتسمى الاستعلامات التى تتم فيها مقارنة قيمة عمود بقيمة محددة(بدلا من نطاق من القيم) باستعلامات التكافؤ. ويقوم تجميع الترميز الرقمى بتخزين صف فى موضع معين اعتمادا على قيمته فى عمود ciuster key. وفى كل مرة يتم إدراج صف، تستخدم قيمة cluster key الخاصة به فى تحديد block الذى يجب تخزينه فيها، ويمكن استخدام العملية ذاتها فى أثناء الاستعلامات للإيجاد السريع لـblocks البيانات المطلوب إنتاجها. وهذا النوع من التجميعات مصمم لتحسين أداء استعلامات التكافؤ، فهى لن تكون مفيدة بالقدر ذاته فى تحسين أداء استعلامات النطاق التى عرضنا لها قبل ذلك.

وتعد الفهارس المعكوسة التى تم فرز البايت الخاص بها بشكل عكسى حلا آخر لإعداد استعلامات التكافؤ. ففى أى فهرس تقليدى، يتم تخزين قيمتين متتاليتين بالترتيب. ولكن فى أى استعلام معكوس، لا يتم تخزين القيم المتتالية بالترتيب الصحيح. فعلى سبيل المثال، يتم تخزين القيمتين ٢٠٠٤ و ٢٠٠٥ كـ ٤٠٠٢ و ٥٠٠٢ على التوالى فى الفهرس المعكوس. وعلى الرغم من أن الفهارس المعكوسة غير مناسبة لعمليات استعراض النطاق، فمن الممكن أن تقلل التنافس blocks الفهارس إذا تم القيام باستعلامات تكافؤ عديدة.

ملحوظة

لا يمكنك جعل أى فهرس نقطى معكوس.

وفى oracle8i يمكنك إنشاء فهارس خاصة بدالة. وقبل oracle8i ، لم يكن بإمكان أى استعلام ينفذ دالة على عمود استخدام فهرس هذا العمود، والتالى، لا يمكن للاستعلام التالى استخدام فهرس على عمود Name :

```
select * from EMPLOYEE
where UPPER (Name) = JONES;
```

لكن يمكن للاستعلام التالى القيام بذلك:

```
select  from EMPLOYEE
where Nam =  JONES;
```

وذلك لأن الاستعلام الثانى لا ينفذ أية دالة على عمود Name. وفى oracle8i، يمكنك إنشاء فهارس تسمح بدعم عمليات الوصول للفهارس لعمليات الوصول المعتمدة على دالة. وبدلا من إنشاء فهرس على العمود Name، يمكنك إنشاء فهرس على تعبير العمود UPPER (Name)، وذلك كما هو موضح فى الكود التالى:

```
create index EMP UPPER NAME on
EMPLOYEE (UPPER Name);
```

وعلى الرغم من أن الفهارس المعتمدة على دالة من الممكن أن تكون مفيدة، يجب أن تتحقق من النقاط التالية عند إنشائها:

● هل يمكنك تقييد الدوال التى ستستخدم مع العمود؟ وإذا كان الأمر كذلك، هل يمكنك تقييد تنفيذ كل الدوال التى تتم على العمود؟

● هل لديك مساحة تخزين مناسبة للفهارس الإضافية؟

● عندماتقوم بحذف الجدول، فأنت تحذف فهارس أكثر (وبالتالى، امتدادات أكثر) عن ذى قبل. كيف سيؤثر هذا على الوقت المطلوب لحذف الجدول؟

تكون الفهارس المعتمدة على دوال مفيدة ولكن لا يجب أن يتم تنفيذها كثيرا. فكلما زاد عدد الفهارس التى تقوم بإنشائها لجدول، تأخذ عمليات الإدراج والتحديث والحذف وقتا أطول.

ولكى تقوم بإنشاء فهرس معتمد على دالة، يجب أن يكون لديك امتياز النظام QUERY REWRITE ويجب إعداد المعاملات التالية فى ملف معامل التهيئة، وذلك كما يلى:

```
QUERY REWRITE ENABLED =
TRUE
```

إنتاج خطط شرح

كيف يمكنك تحديد مسار الوصول الذى ستستخدمه قاعدة البيانات فى تنفيذ أى استعلام؟

والإجابة هى أنه يمكنك عرض هذه المعلومات عن طريق أمر explain plan . وسيقوم الأمر بتحديد مسار التنفيذ الخاص بالاستعلام ويضع مخرجاته فى جدول (يسمى PLAN TABLE) فى قاعدة البيانات. ويظهر مثال لهذا الأمر فى الكود التالى:

```
explain Plan
set statement Id = TEST
for
select from EMPLOYEE
where city Y8;
```

يطلب السطر الأول من هذا الأمر من قاعدة البيانات شرح خطة التنفيذ الخاصة بها للاستعلام دون تنفيذ الاستعلام. ويعنون السطر التالى سجلات هذا الاستعلام فى PLAN TABLE بـ State-ment Id تساوى TEST . وسيعرض الاستعلام الذى سيتم تحليله بعد الكلمة الأساسية for.

ويجب أن يكون لحق الدخول الذى يقوم بتشغيل هذا الأمرpLan TABLE فى مخطط قاعدة البيانات الخاص به. وتوفر Oracle أو أمر creaet table الى يحتاج إليها هذا الجدول. وعادة ما يوجد الملف utlx-plan. sql فى الدليل الفرعى ;rdbms/admin تحت الدليل الرئيسى الخاص ببرامج Ora-cle. ومن الممكن أن يشغل المستخدمون النص البرمجى هذا لإنشاء الجدول فى مخططات قاعدة البيانات الخاصة بهم.

ملحوظة

يجب حذف وإعادة إنشاء جدول الخطة بعد كل تحديث لـ Oracle ، حيث إنه من الممكن إضافة أعمدة جديدة من خلال النصوص البرمجية الخاصة بالتحديث.

قم باستعلام جدول الخطة باستخدام الاستعلام التالى. إن سجلات هذا الجدول مرتبطة ببعضها البعض، وبالتالى يمكن استخدام عبارة connect by الخاصة بجملة select فى تقييم التدرج.

سيعرض هذا الاستعلام أنواع العمليات التى يجب القيام بها على قاعدة البيانات لتنفيذ الاستعلام. وإليك أول ثلاثة أعمدة من المخرجات:

```
i p PLAN PLUS EXP
o SELECT STATEMENT optimizer
= CHOOSE
1 o TABLE ACCESS (BY INDEX
ROWID) of EMPLOYEE'
2 1 INDEX (RANGE SCAN) of
PCLTY ZIP NDX
NON . UNIQUE
```

ولكى تقوم بقراءة خطة الشرح، اقرأ ترتيب العمليات بشكل معكوس فى التدرج حتى تصل إلى مجموعة من العمليات التى توجد على المستوى نفسه من المسافة البادئة، ثم اقرأ من أعلى إلى أسفل. فى هذا المثال، لا توجد هناك عمليات على المستوى ذاته من المسافة البادئة، وبالتالى، ستقوم بقراءة ترتيب العمليات بشكل معكوس. إن أول عملية هى استعراض نطاق فهرس، متبوعة بوصول للجدول، تعرض عملية SE-LECT STATEMENT المخرجات للمستخدم. إن لكل عملية قيمة ID (العمود الأول) وقيمة ID رئيسية. وفى خطط الشرح الأكثر تعقيدا، من الممكن أن تحتاج لاستخدام قيم ID الرئيسية لتحديد ترتيب العمليات.

وتوضح هذه الخطة أن البيانات المنتجة للمستخدم تأتى من TAble Access باستخدام RowID. وتقوم عملية استعراض نطاق فهرس بتوفير RowIDs، باستخدام فهرس CITY ST ZIP NDX الذى عرضنا له قبل ذلك.

ملحوظة

كما هو الحال فى Oracle8i ، يمكنك استخدام مخططات تفصيلية مخزنة لحفظ مسار التنفيذ الخاص باستعلام. ارجع إلى الفصل الخامس لمزيد من التفاصيل عن تنفيذ المخططات التفصيلية المخزنة.