



行政院及所屬各機關出國報告

(出國類別：研習 編號：PIO/P No.: 484-C03-30324)

參加 Oracle 公司舉辦之資料庫國際技術年會
-OracleWorld 及其它資訊管理相關議題實習
To attend 「the annual international technology
conference - OracleWorld and other Information
Management programs」

服務機關：經濟部 資訊中心

出國人職稱：約聘人員

姓名：袁國鶯

出國地區：美國

出國期間：92年9月6日至92年9月20日

報告日期：92年12月10日

I8 / 09205094

系統識別號：C09205094

公務出國報告提要

頁數：15 含附件：否

報告名稱：

參加 Oracle 公司舉辦之資料庫國際技術年會-OracleWorld 及其它資訊
管理相關議題實習

主辦機關：

經濟部

聯絡人/電話：

林純白 / 23212200#267

出國人員：

袁國薦 經濟部 資訊中心 約聘制度工程師

出國類別：實習

出國地區：美國

出國期間：民國 92 年 09 月 06 日 - 民國 92 年 09 月 20 日

報告日期：民國 92 年 12 月 10 日

分類號/目：I8 / 資訊科學 I8 / 資訊科學

關鍵詞：OracleWorld

內容摘要：本部以 Oracle 資料庫管理系統為作業平台開發完成之系統計有文案管理系統、國會業務管理系統、全國工商管理系統及經濟統計查詢系統等，然某些程式因結構化查詢語言語法撰寫不良，以至執行效率不彰，故應研究收集此方面專家學者之心得經驗，並將之撰寫成範例，以供本部相關程式開發人員及本部應用系統承包廠商之依循標準。本報告之內容主在針對結構化查詢語言語法撰寫好壞，作一初步探討及提出其對應之解決方案，並希望爾後能繼續加強此方面經驗之累積，以提供應用系統使用者最佳之回應時間。

主要內容如下：

1. 結構化查詢語言語法撰寫不良之問題及其對應之解決方案
2. 以成本為計算基礎最佳化器語法撰寫之問題及其解決方案

本文電子檔已上傳至出國報告資訊網

赴美國研習

「參加 Oracle 公司舉辦之資料庫國際技術年會 -OracleWorld 及其它資訊管理相關議題實習」 報告書

目 錄

- 一、 目的
- 二、 行程
- 三、 內容
- 四、 心得與建議

一、 目的

本次實習之主要目的，旨在使本部內部開發及委外開發之 Oracle 應用程式之執行效率皆有一定水準，唯其先決條件，在於程度設計員有一 good knowledge 有關如何撰寫優良之結構化查詢語言語法，以為依循，故本次報告亦以此為重點來說明較佳之結構化查詢語言語法撰寫之方式。

二、 行程(San Francisco)

9/7-9/11 參加 OracleWorld 資料庫管理系統技術年會

9/12-9/12 參訪 BMC Corporation : DB monitor tool :
Patrol for Oracle and Smart DBA

9/15-9/19 參加 Oracle 教育中心訓練課程:Introduction
to Oracle9i: 結構化查詢語言

三、 內容

Oracle 公司目前為全世界最大之關聯式資料庫廠商，其技術年會每年舉辦一次。本年計有 250 個參展廠商，

technical sessions 亦有 400 餘場，精彩場次甚多，本人僅收集結構化查詢語言 performance tuning 方面之實用資料，以供有須要者參考（另參加 Oracle 教育中心訓練課程:Introduction to Oracle9i: 結構化查詢語言，其主要目的亦如是）。

四、心得與建議

結構化查詢語言語法撰寫不良之問題及其對應之解決方案

結構化查詢語言語法撰寫之好壞影響使用者之 Response Time 至鉅，寫的不好，可能一個 enter 下去，數十分鐘沒有回應，以致應用系統根本無法使用。一般問題如下：

1. 結構化查詢語言之寫法讓 Oracle 最佳化器無法使用索引
2. 索引建立得不恰當或未建
3. 使用單一欄位索引作 merge
4. 誤用 nested loop, sort merge, or hash join
5. 誤用 IN, NOT IN, EXISTS, NOT EXISTS, 或表格 JOINS
6. 作不必要之排序
7. 在表格上建立太多之索引
8. 刪除許多資料
9. 其他

1. 結構化查詢語言之寫法讓 Oracle 最佳化器無法使用索引

一般當應用程式去抓取資料庫之資料，如抓取之資料量超過總資料量之 10% 時，應使用 Full 表格 Scan 才可獲得較好之執行效率（適用於 Batch 及 Report 作業）。但一般之 online 線上作業，僅會抓取少數幾筆資料，故如不使用索引去抓取資料，將會使應用系統回應時間加長，如表格之資料為好幾百萬筆，每一應用程式皆去掃整個表格，其影響之深遠，當可預見。

在未建立以功能為基礎索引之情況下，而在 where 子句，使用 Function 於索引欄位，Oracle 將無法使用該索引抓取資料。

其不良與修正結構化查詢語言撰寫方式舉例示如下：

不良：

```
select name,salary,hire_date
from employ
where substr(name,1,1) = 'D';
```

修正寫法：

```
select name,salary,hire_date
from employ
where name like = 'D%';
```

不良：

```
select name,salary,hire_date
from employ
where salary != 0;
```

修正寫法：

```
select name,salary,hire_date
from employ
where salary > 0';
```

不良：

```
select name,salary,hire_date
from employ
where
trunc(hire_date)=trunc(sysdate);
```

修正寫法：

```
select name,salary, hire_date
from employ
where hire_date between trunc(sysdate)
and trunc(sysdate) + .9999;
```

不良:

```
select
dept_code,first_name,salary,hire_date
from employ
where dept_code||first_name='0140David';
```

修正寫法:

```
select
dept_code,first_name,salary,hire_date
from employ
where dept_code='0140' and name='David';
```

不良:

```
select
dept_code,first_name,salary,hire_date
from employ
where salary+3000>100000;
```

修正寫法:

```
select
dept_code,first_name,salary,hire_date
from employ
where salary < 97000;
```

(所有數學運算-加減乘除(+*/) 皆有相同負面效果)

不良:


```
select name,salary,hire_date
from employ
where name=nvl(:emp_name ,name);
```

(因 name 出現在 where 子句兩邊,導致索引不會被使用)

修正寫法:

```
select name,salary,hire_date
from employ
where name like nvl(:emp_name ,'%');
```

另外 where 子句兩邊之比較欄位之資料型態須一致,索引才會被使用。如下述之結構化查詢語言指令,不會使用索引來抓資料,如 dept_code 欄位之資料型態之定義為 varchar2,因 0140 為一數值。

```
select dept_code
from employ
where dept_code=0140
```

Oracle 會將上述結構化查詢語言指令改寫為:

```
select dept_code
from employ
where to_number(dept_code)=0140
```

如欲使用以功能為基楚索引，其步驟如下：

a. 在 ini.ora parameter file 中設定

```
query_rewrite_enabled=true
```

b. analyze 表格及索引以使用以成本為計算基礎之最佳化器

c. 建立以功能為基楚索引

例：

```
create index employ_name_fidx on
employ(upper(first_name))
select count(*)
from employ
where upper(first_name)='DAVID'
```

2. 索引建立得不恰當或未建

撰寫結構化查詢語言程式時，應先思考如下問題：

a. 使用索引去抓取資料好，還是作 Full 表格 Scan 去抓取資料好？

a.1 抓得資料量超過總資料量 10%嗎？

a.2 是否所抓的資料由索引即可滿足？無須經由表格抓取資料，如一查詢之 response time 非常重要，可考慮將所有須查詢之欄位建立一索引。例如下述結構化查詢語言，原有一索引，其欄位為 <dept_code>，可更改其欄位為 <dept_code,area_code,salary>

```
select sum(salary)
```

```
from employ
where dept_code=0140
and area_code='03';
```

b. 使用的索引是否為最佳之索引？

如結構化查詢語言指令有 ORDER BY 子句，是否有一索引是排序依照 ORDER BY 之欄位〈如是 Oracle 無須再作排序動作，應設法使 Oracle 使用此一索引〉

c. 新建立一索引可否大大縮短 Response Time？

d. JOIN 欄位如不在索引欄位之列，應將其加入

e. 如表格資料筆數甚少，亦應建立索引

3. 使用單一欄位索引作 merge

撰寫結構化查詢語言程式時，應避免使用單一欄位索引作 merge。例如下述結構化查詢語言，原有兩索引，其欄位分別為〈close_flag〉及〈credit_code〉，可更改為單一索引，其欄位為〈credit_code, open_flag〉

```
select count(*)
from customer
where credit_code ='9'
and close_flag ='Y';
```

以上假設 close_flag ='Y' 之資料筆數有好幾百萬筆，而 credit_code ='9' 之資料筆數只有幾百筆

4. 誤用 nested loop, sort merge, or hash join

- a. 就線上作業使用 nested loops, 就批次及報表作業使用排序 merge, or hash join
- b. hash join 只能使用在 equality(=), 不能使用在 ranges(<, <=, >, >=)
- c. hash join 一般較 merge join 執行速度快
- d. 在結構化查詢語言語法中可使用 hints (USE_NL, USE_HASH or USE_MERGE), 來決定以何種方式抓取資料

5. 誤用 IN, NOT IN, EXISTS, NOT EXISTS, 或表格 JOINS

- a. 一般而言使用表格 joins 較使用 subquery 執行速度快, 如須使用 subquery, 使用 EXISTS 又較使用 IN 好(但如在 subquery 之 inner 表格資料較 outer 表格之資料少很多, 則相反)
- b. 如使用 EXISTS, 在結構化查詢語言語法中可使用 hints HASH_SJ or MERGE_SJ, 可大大加快 EXISTS 之執行效率, 因加入 hint 後, 在 subquery 之資料, 僅被讀一次

6. 作不必要之排序

- a. 排序動作非常消耗系統資源, 能避免則避免之

下列運作會使 Oracle 作排序動作：

CREATE INDEX，DISTINCT，GROUP BY，
ORDER BY，INTERSECT，MINUS，UNIONS，
UNINDEXED TABLE JOINS

b. 在建 Users 時，定義 TEMPORARY tablespace
為 TEMP，例如：

```
create user DAVID
identified by tiger
default tablespace gy0lts_data
temporary tablespace temp
quota 500m on gy0lts_data
quota 100m on gy0lts_indx
quota 100m on temp;
```

及在 INI.ORA 檔中定義大一點的
SORT_AREA_SIZE(sort in memory) 及定義
TEMPORARY tablespace 之 INITIAL 及 NEXT
extents 為 SORT_AREA_SIZE 之倍數，可加快排
序速度

c. 如情況允許使用 UNION ALL(不會去除重複之
ROWS) 取代 UNION(會去除重複之 ROWS) 以避免作
排序動作

d. 如結構化查詢語言指令有 ORDER BY 子句，可建
立一索引依照 ORDER BY 之欄位排序，以避免作
排序動作，並應使用 hint 讓 Oracle 使用此一索
引

原寫法:

```
select first_name,last_name
from employ e
order by first_name;
```

修正寫法:

```
select /*+index_asc(e emp_idx) */
first_name,last_name
from employ e
order by first_name;
```

7. 在表格上建立太多之索引

- a. 在表格上建立太多之索引，一般對 online 作業只會有一點影響，但對作大量 insert，update，delete 之 batch 作業影響較大，可考慮先 drop 索引再作此等 batch 作業
- b. rebuild 索引時可使用 NOLOGGING OR UNRECOVERABLE 選項及 PARALLEL 處理功能，以加快速度
- c. 應刪除從未使用之索引(Oracle9i 可使用 ALTER 索引 MONITORING USAGE 找出此種索引)

8. 刪除許多資料

假使表格之資料被大量 delete 或索引之欄位被大量 update 時，應將表格及索引 rebuild 以提昇結構化查詢語言執行效率，當然不要忘了作 analyze 以便 Oracle 之 cost_based 最佳化器能利用收集

到的資訊，決定最佳之抓取資料方式

9. 其他

- a. 勿使用 view 中 view
- b. 勿用 hints 於 outer view
- c. JOIN 之表格不要超過五個

以成本為計算基礎最佳化器語法撰寫之問題及其解決方案

一般問題如下：

1. 資料分配差異極大問題
2. 未適當、適時分析資料
3. 與以規則為基礎之最佳化器混淆使用
4. 選擇不佳之索引
5. init.ora 設定錯誤

1. 資料分配差異極大問題

假設有一索引欄位 Status 只有 '0' 或 'C' 兩值，而 Status='C' 之筆數，有數百萬筆，而 Status='0' 之筆數，僅有數百筆，為讓最佳化器在抓取 Status='C' 之資料用表格 Scan 之方式，而在抓取 Status='0' 之資料時，透過索引之方式，須使用下述方式 analyze 表格：

analyze table xxtable compute statistics
for all indexed columns

如是才可得到最佳之回應時間。

2. 未適當、適時分析資料

- a. 表格及索引須定期作分析
- b. 表格或索引重建後，應馬上作分析
- c. 新增索引時須馬上分析該索引

3. 與規則為計算基礎之最佳化器混淆使用

- a. 不要 JOIN 表格：當有得表格作過分析
而有得卻沒作過分析
- b. 表格及索引應同時作分析

4. 選擇不佳之索引

如遇最佳化器選擇執行不佳之索引去抓取資料，可在結構化查詢語言指令加 hint 解決之

5. init.ora 設定錯誤

- a. 注意 compatible 之設定是否正確
- b. 設定 ALWAYS_SEMI_JOIN 為 MERGE 或
HASH：當結構化查詢語言語法使用 WHERE EXIST
子句頻繁
- c. 設定 ALWAYS_ANTI_JOIN 為 MERGE 或
HASH：當結構化查詢語言語法使用 NOT IN 子句
頻繁

行政院及所屬各機關出國報告提要

出國報告名稱：參加 Oracle 公司舉辦之資料庫國際技術年會-OracleWorld 及其它資訊管理相關議題實習

To attend 「the annual international technology conference - OracleWorld and other Information Management programs」

頁數_____ 含附件：是 否

出國計畫主辦機關／聯絡人／電話：經濟部／盧美雅／(02) 23212200-614

出國人員姓名／服務機關／單位／職稱／電話：

袁國鶯／經濟部／資訊中心／約聘人員／(02)23212200-687

出國類別：1 考察 2 進修 3 研究 4 實習 5 其他

出國期間：92 年 9 月 6 日至 92 年 9 月 20 日 出國地區：美國

報告日期：92 年 12 月 10 日

分類號／目：484-C03-30324

關鍵詞：484-C03-30324

內容摘要：(二百至三百字)

本部以 Oracle 資料庫管理系統為作業平台開發完成之系統計有文案管理系統國會業務管理系統、全國工商管理系統及經濟統計查詢系統等，然某些程式因結構化查詢語言語法撰寫不良，以至執行效率不彰，故應研究收集此方面專家學者之心得經驗，並將之撰寫成範例，以供本部相關程式開發人員及本部應用系統承包廠商之依循標準。本報告之內容主在針對結構化查詢語言語法撰寫好壞，作一初步探討及提出其對應之解決方案，並希望爾後能繼續加強此方面經驗之累積，以提供應用系統使用者最佳之回應時間。

主要內容如下：

1. 結構化查詢語言語法撰寫不良之問題及其對應之解決方案
2. 以成本為計算基礎最佳化器語法撰寫之問題及其解決方案

電子檔已上傳至出國報告資訊網 (<http://report.gsn.gov.tw>)