【実行計画の最適化方針】

テーブル結合の最適順序

ヒント句指定の Select 文の書き方

・表結合時の結合方法とレコード絞り込みの検索条件合わせた索引の指定の方法

・実行計画の Operation 処理順序

求めたいレコードの select 条件（ヒント句なし）

```
Select  location_name ,  from  employees  e ,  departments  d  location  l
        where  e. salary  >  10000  and  d. department_name  =  'Finance'
          and  e. department_id  =  d. department_id
          and  d. location_id  =  l. location_id
```

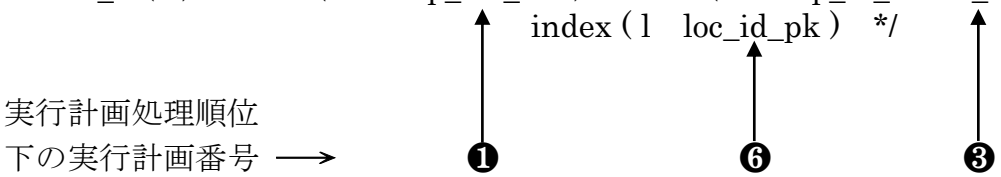
―――― ヒント句の指定方法 ――――

```
/*+  ヒント内容  */
```

――――――――――――――――――――――――――――――

A ①指定するヒント句

```
/*+  use_nl(l)  index(e  emp_sal_idx)  index(d  dep_id_name_idx)
                                        index(l  loc_id_pk)  */
```



※ use_nl(l)： テーブル間の結合には、強制的に Nested Loop 結合を行う
emp_sal_idx： e（emp）表の saraly 列に対するインデックス
dep_id_name_idx：d（department）表に対する department_id 列と department_name 列がキー構成の複合インデックス
loc_id_pk： l（location）表の id 列を構成にした主キー

A ② 実行計画の確認

explain plan for 文を使って、SQL 文の実行計画を検証する

ID	OPERATION	Name
0	SELECT STATEMENT	
1	❽ NESTED LOOPS	
2	❺ NESTED LOOPS	
3	❷ TABLE ACCESS BY INDEX ROWID	EMPLOYEES
＊4	❶ INDEX RANGE SCAN	EMP_SAL_IDX
5	❹ TABLE ACCESS BY INDEX ROWID	DEPARTMENTS
＊6	❸ INDEX RANGE SCAN	DEP_ID_NAME_IDX
7	❼ TABLE ACCESS BY INDEX ROWID	LOCATIONS
＊8	❻ INDEX UNIQUE SCAN	LOC_ID_PK

4 – access（ “E”.”SALARY” > 10000 ）
6 – access（ “E”.”DEPARTMENT_ID” = “D”.”DEPARTMENT_ID” AND “D”.” DEPARTMENT_NAME” = ‘France’ ）
8 – access（ “D”.”LOCATION_ID” = “L”.”LOCATION_ID” ）

A ③検索結合順序の説明

1. e. emp_sal_idx を使って e. salary > 10000 を検索する
2. 1.の結果の表の **e.** department_id に対して、d. dep_id_name_idx を使って e. department_id = d. department_id and d. department_name = 'Finance' を検索する
3. 2.の結果の表の **d.** location_id に対して、l. LOC_ID_PK を使って d. location_id = l. location_id を条件として結合する

A ④　アクセス・ブロック数の確認

autotrace を使って、Select 文の実行する

69　consistent gets　　（アクセスしたブロック数(バッファ文＋ディスク I／O 文)）

B ①指定するヒント句

/*+　use_nl(l)　index(e　emp_deptid_sal_idx)　index(d　dep_name_idx)
index(l　loc_id_pk)　*/

実行計画処理順位
下の実行計画番号 ⟶　❻　❸　❶

インデックスをヒント句に書いた場合、実行計画でのテーブルの結合順は、
オプティマイザがインデクスに基づき、最適順を解析して決定する

※ use_nl(l)：　テーブル間の結合には、強制的に Nested Loop 結合を行う
emp_deptid_sal_idx：e(employees)表に対する department_id と salary
がキー構成の複合インデックス
dep_name_idx：　d（department）表の name 列に対するインデックス
loc_id_pk：　l（location）表の id 列を構成にした主キー

B ②実行計画の確認

explain　plan　for　文を使って、SQL 文の実行計画を検証する

```
-----------------------------------------------------------------------------------------------
|  ID | OPERATION                                        |        Name                |
-----------------------------------------------------------------------------------------------
|   0  |  SELECT  STATEMENT                              |                            |
|   1  |❼ NESTED  LOOPS                                  |                            |
|   2  |❺   NESTED  LOOPS                                |                            |
|   3  |❷     TABLE  ACCESS  BY  INDEX  ROWID  | DEPARTMENTS                |
| *4  |❶       INDEX  RANGE  SCAN                       | DEP_NAME_IDX               |
|   5  |❹     TABLE  ACCESS  BY  INDEX  ROWID  | LOCATIONS                  |
| *6  |❸       INDEX  UNIQUE  SCAN                      | LOC_ID_PK                  |
| *7  |❻   INDEX  RANGE  SCAN                           | EMP_DEPID_SAL_IDX|
-----------------------------------------------------------------------------------------------
```

4 – access（ “D”.”DEPARTMENT_NAME”　=　‘France’ ）
6 – access（ “D”.”LOCATION_ID”　=　“L”.”LOCATION_ID” ）
7 – access（ “E”.”DEPARTMENT_ID”　=　“D”.”DEPARTMENT_ID”　AND
“E”.”SALARY”　>　10000 ）
filter　（ “E”.”SALARY”　>　10000 ）

B ③検索結合順序の説明

1. d. dep_name_idx を使って d. department_name = 'Finance' を検索する
2. 1.の結果の表の **d.** location_id に対して、l. LOC_ID_PK を使って、d. location_id = l. location_id を条件として結合する
3. 2.の結果の表の **d.** department_id 列に対して、e. emp_deptid_sal_idx を使って e. department_id = d. department_id and e. salary > 10000 を検索し表を結合する

B ④アクセス・ブロック数の確認

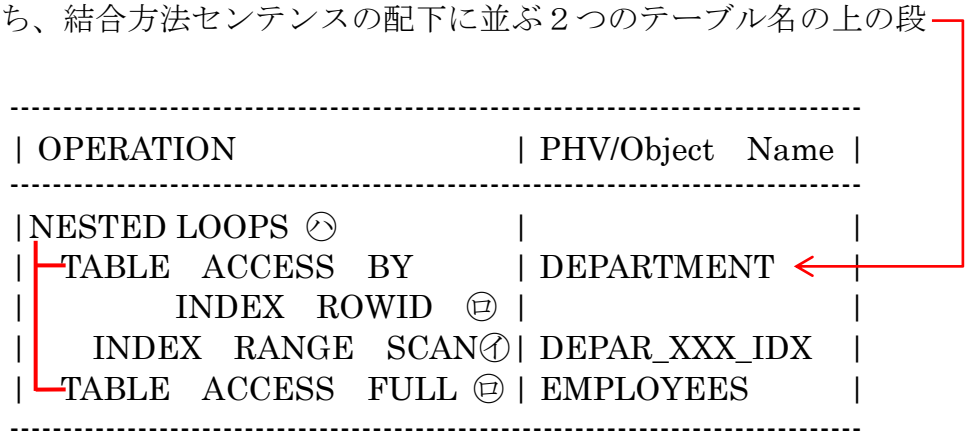
autotrace を使って、Select 文の実行

8 consistent gets （アクセスしたブロック数(バッファ文＋ディスク I／O 文)）

よって、「B の方が、効率が良い」と言える

【効率がよいテーブルの結合順序】

件数が少ない表を結合順序の前方（先処理）に持ってくる
すなわち、結合方法センテンスの配下に並ぶ２つのテーブル名の上の段



――――― ヒント句の指定方法 ―――――
/*+ ヒント内容 */

最初に、テーブル間の結合方法を指定
次に、

㋐ 使用するテーブルのインデックスを指定
（指定できるのは、どのインデックスを使うかの指示のみで、実行計画で使う優先順位は指定できない）
この場合のテーブル間の結合順序は、Oracle のオプティマイザが自動で最適順を判断する

㋑ 使用するテーブル名を指定
~~この場合のテーブル間の結合順序は、左側の記述が先に実行されるように実行計画が作成される~~）

――――――――――――――――――――――――

実際の実行計画のテーブル結合順は、実行順序の小さい番号から行われていく

【ヒント句指定の Select 文の書き方】

```
Select   /*+   use_nl(l)      index(   テーブル名１   インデックス１   )
               index(   テーブル名２   インデックス２   )   ・・・・      */
     From    ・・・
     Where   検索条件   ;
```

※ インデックス名を省略した場合には、Oracle が最も有効なインデックスを選択して処理を行う

例

```
Select   /*+   index(   flagtest   flagtest_idx   )   */
     From   flagtest
     Where   key   =   'Y'   ;
```