Como optimizar el uso del FOR ALL ENTRIES

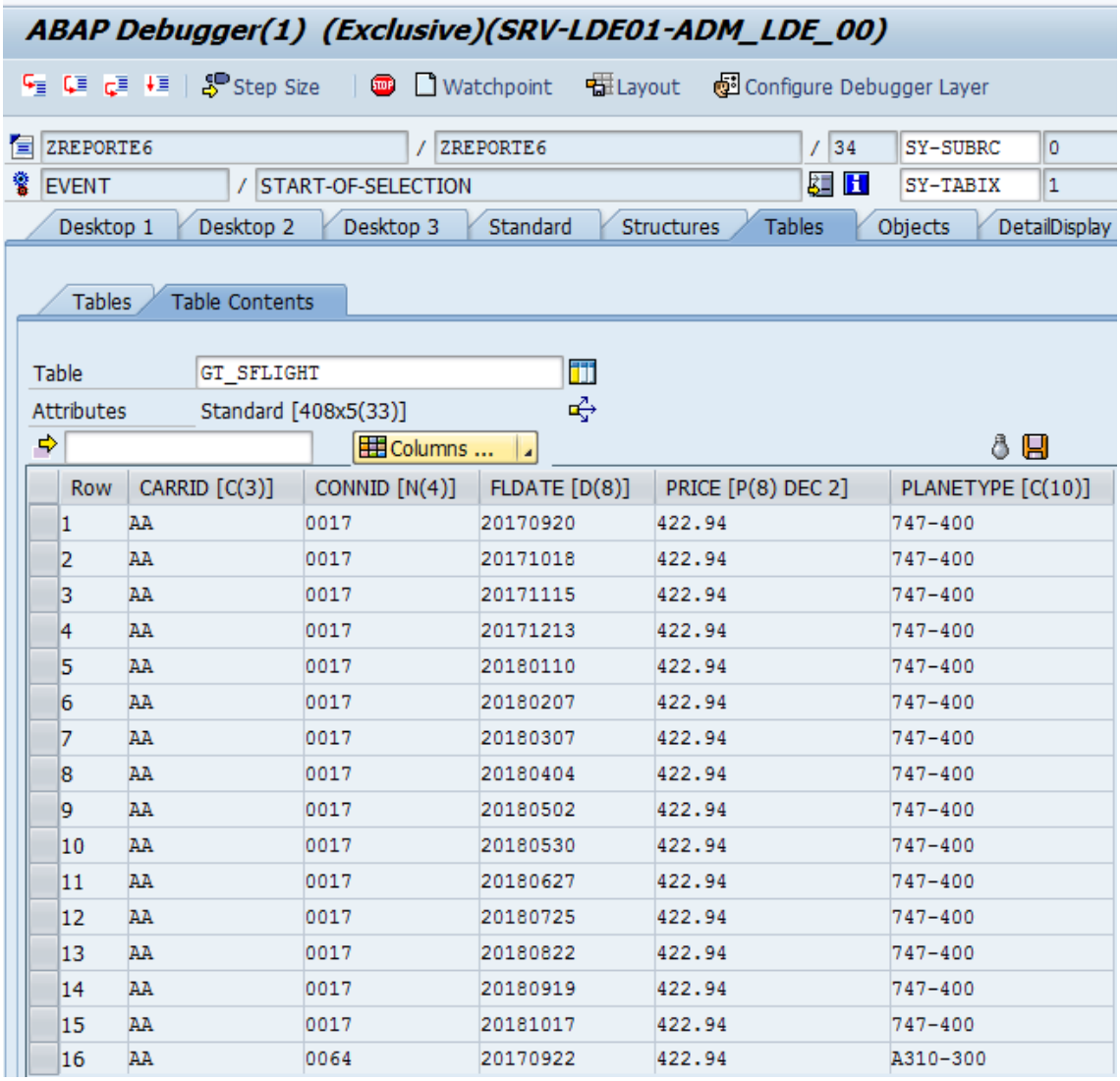
Ejemplo: ZREPORTE_EJ

```
TYPES: BEGIN OF ty_sflight,
          carrid    TYPE sflight-carrid,
          connid    TYPE sflight-connid,
          fldate    TYPE sflight-fldate,
          price     TYPE sflight-price,
          planetype TYPE sflight-planetype,
       END OF ty_sflight,

       BEGIN OF ty_scarr,
          carrid   TYPE scarr-carrid,
          carrname TYPE scarr-carrname,
       END OF ty_scarr.

DATA: gt_sflight     TYPE TABLE OF ty_sflight,
      gt_scarr       TYPE TABLE OF ty_scarr,
      gt_sflight_tmp TYPE TABLE OF ty_sflight,
      gs_sflight     TYPE ty_sflight,
      gs_scarr       TYPE ty_scarr.

REFRESH: gt_sflight, gt_scarr, gt_sflight_tmp.

SELECT carrid connid fldate price planetype FROM sflight
  INTO TABLE gt_sflight.

IF sy-subrc IS INITIAL.
```

```
  APPEND LINES OF gt_sflight TO gt_sflight_tmp.
  SORT gt_sflight_tmp BY carrid.

  DELETE ADJACENT DUPLICATES FROM gt_sflight_tmp COMPARING carrid.

  SELECT carrid carrname FROM scarr
    INTO TABLE gt_scarr
     FOR ALL ENTRIES IN gt_sflight_tmp
   WHERE carrid = gt_sflight_tmp-carrid.

Esta es la parte que se optimiza el FOR ALL ENTRIES!!!
```

```
IF sy-subrc IS INITIAL.

    LOOP AT gt_sflight INTO gs_sflight.

      READ TABLE gt_scarr INTO gs_scarr WITH KEY carrid = gs_sflight-carrid.
      IF sy-subrc IS INITIAL.

        WRITE: /    gs_sflight-carrid, gs_scarr-carrname,

                    gs_sflight-connid, gs_sflight-fldate.

       ENDIF.

    ENDLOOP.

ENDIF.
```

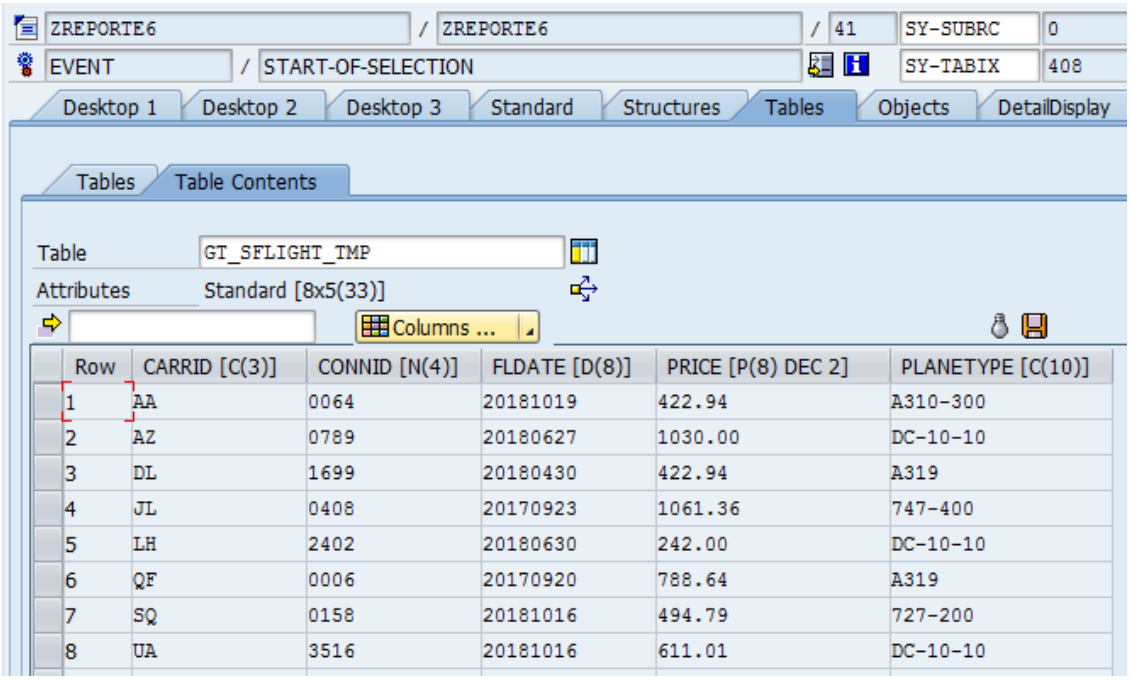
Si realizamos un DEBUG podemos observar los registros que obtenemos en la tabla interna GT_SFLIGHT:



Row	CARRID [C(3)]	CONNID [N(4)]	FLDATE [D(8)]	PRICE [P(8) DEC 2]	PLANETYPE [C(10)]
1	AA	0017	20170920	422.94	747-400
2	AA	0017	20171018	422.94	747-400
3	AA	0017	20171115	422.94	747-400
4	AA	0017	20171213	422.94	747-400
5	AA	0017	20180110	422.94	747-400
6	AA	0017	20180207	422.94	747-400
7	AA	0017	20180307	422.94	747-400
8	AA	0017	20180404	422.94	747-400
9	AA	0017	20180502	422.94	747-400
10	AA	0017	20180530	422.94	747-400
11	AA	0017	20180627	422.94	747-400
12	AA	0017	20180725	422.94	747-400
13	AA	0017	20180822	422.94	747-400
14	AA	0017	20180919	422.94	747-400
15	AA	0017	20181017	422.94	747-400
16	AA	0064	20170922	422.94	A310-300

Como se puede ver el campo CARRID se repite muchas veces entonces cuando se va a hacer el FOR AL ENTRIES con la Base de Datos SCARR por este mismo campo, pierde mucho tiempo preguntando por todas entradas repetidas si son iguales a los que contiene la tabla SCARR.

Por eso se puede copiar todos los registros de la tabla GT_SFLIGHT a otra tabla interna auxiliar como por ejemplo GT_SFLIGHT_TMP ó GT_SFLIGHT_AUX, se ordena esta nueva tabla interna por el campo CARRID y se borran los duplicados y Obtenemos lo siguiente:



Row	CARRID [C(3)]	CONNID [N(4)]	FLDATE [D(8)]	PRICE [P(8) DEC 2]	PLANETYPE [C(10)]
1	AA	0064	20181019	422.94	A310-300
2	AZ	0789	20180627	1030.00	DC-10-10
3	DL	1699	20180430	422.94	A319
4	JL	0408	20170923	1061.36	747-400
5	LH	2402	20180630	242.00	DC-10-10
6	QF	0006	20170920	788.64	A319
7	SQ	0158	20181016	494.79	727-200
8	UA	3516	20181016	611.01	DC-10-10

Entonces es con esta nueva tabla interna auxiliar con la que tenemos que hacer el FOR ALL ENTRIES a la tabla SCARR para poder optimizar la consulta.

Con la Instrucción *FOR ALL ENTRIES* hay que tener mucho cuidado en que la tabla base (IT_SFLIGHT en el ejemplo), con la se va a comparar no se encuentre vacía, porque en ese caso, estaríamos extrayendo **TODA LA DATA** existente en la tabla a consultar, perjudicando el performance del programa, es decir, siguiendo con el ejemplo, si la tabla interna IT_SFLIGTH se encuentra vacía al momento de ejecutar el *FOR ALL ENTRIES*, el sistema estaría haciendo caso omiso a la cláusula WHERE e intentaría recuperar TODOS LOS REGISTROS de la tabla SFLIGHT, la cual si la tabla tiene millones de registros, en este punto estaríamos dañando por completo el performance de nuestro programa.

Y por último, podemos tener la opción de comprobar rendimiento de la lógica del reporte, a través de la **transacción SE30**.