

“Documentación Iteración 4”

Esteban Reyes Marcelo, Juan Camilo Sánchez
Contexto de Presentación del documento
Universidad de los Andes, Bogotá, Colombia
{e.reyesm , jc.sanchez16 }@uniandes.edu.co

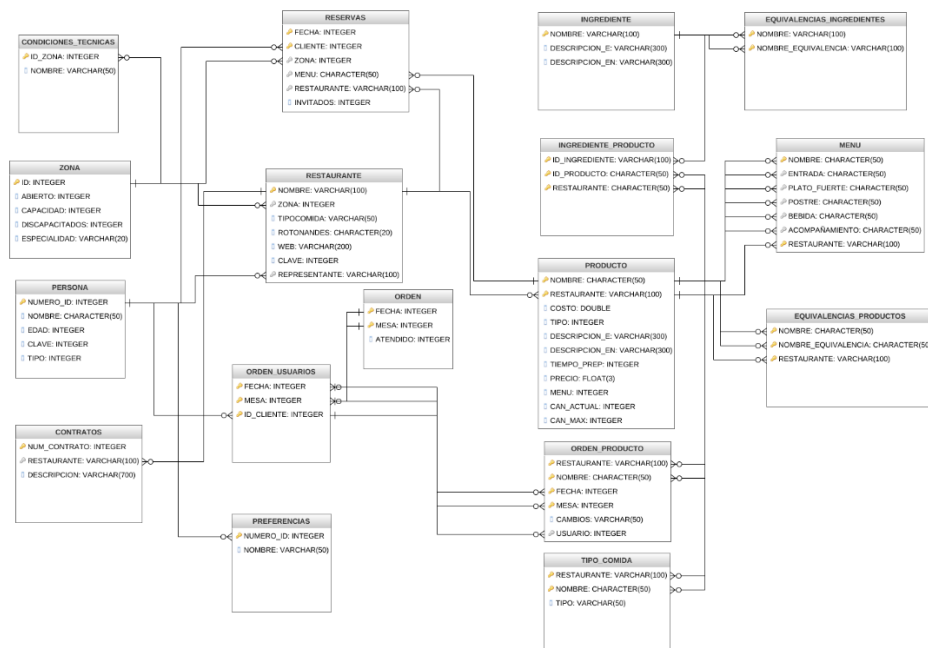
Fecha de presentación: Abril 28 de 2010

Tabla de contenido

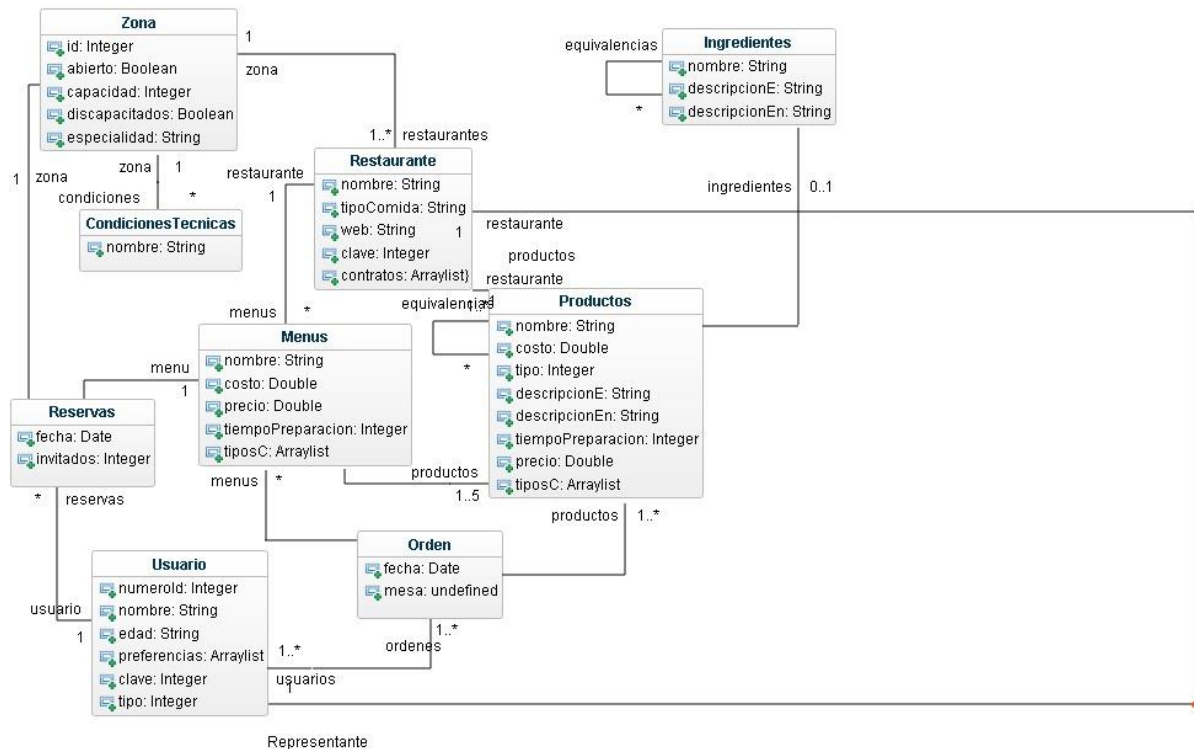
- 1 Analisis..... ¡Error! Marcador no definido.
- 2 Otros aspectos de manejo de estilos ¡Error! Marcador no definido.

1 Análisis.

- Diagrama relacional.



- Diagrama de clases.



2 Diseño de la aplicación.

i. Cambios relevantes en el diseño.

Para empezar, se realizó un cambio fundamental en la parte del código ya que solo se manejará una clase grande TM la cual manejará todas las transacciones en la base de datos. También se quitó los llamados entre DAOS pues con esto disminuir los punteros activos en una transacción, por parte de las tablas no se creó ninguna más ni se borró alguna, lo único fue que les aumentamos los datos tratando de dejar

ii. Documentación diseño físico.

Para los siguientes requerimientos:

- RFC9:

Para este requerimiento utilizamos la siguiente sentencia:

```

SELECT * FROM ((SELECT * FROM ORDEN_PRODUCTO PR WHERE PR.RESTAURANTE = 'bieiggyvqybybdjddzvvv' AND PR.FECHA BETWEEN 7449080400000 AND 7449166740000 ) PRR
JOIN (PRODUCTO U JOIN TIPO_COMIDA TC ON U.NOMBRE = TC.NOMBRE AND U.RESTAURANTE = TC.RESTAURANTE) ON U.RESTAURANTE = PRR.RESTAURANTE AND U.NOMBRE=PRR.NOMBRE)
JOIN PERSONA PS ON PS.NUMERO_ID = PRR.USUARIO);

```

Tiempo:

Hoja de Trabajo

Generador de Consultas

```

SELECT * from (((SELECT * FROM ORDEN_PRODUCTO PR WHERE PR.RESTaurante = 'bieigvvsqyybdjddzvww' AND PR.FECHA BETWEEN 7449080400000 AND 7449166740000 ) PRR
JOIN (PRODUCTO U JOIN TIPO_COMIDA TC ON U.NOMBRE = TC.NOMBRE AND U.RESTaurante = TC.RESTaurante)on U.RESTaurante = PRR.RESTaurante AND U.NOMBRE=PRR.NOMBRE)
JOIN PERSONA PS ON PS.NUMERO_ID = PRR.USUARIO);

```

Salida de Script x Explicación del Plan x Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0,003 segundos

| RESTAURANTE | NOMBRE | FECHA | MESA | CAMBIOS | USUARIO | NOMBRE_1 | RESTAURANTE_1 |
|----------------------|----------------------|---------------|------|---------|----------------------------|----------------|----------------|
| bieigvvsqyybdjddzvww | kqypshhcmmsusboragkd | 7449113041611 | 67 | (null) | 84707 kqypshhcmmsusboragkd | bieigvvsqyybdj | bieigvvsqyybdj |

Plan de consulta:

Hoja de Trabajo

Generador de Consultas

```

SELECT * from (((SELECT * FROM ORDEN_PRODUCTO PR WHERE PR.RESTaurante = 'bieigvvsqyybdjddzvww' AND PR.FECHA BETWEEN 7449080400000 AND 7449166740000 ) PRR
JOIN (PRODUCTO U JOIN TIPO_COMIDA TC ON U.NOMBRE = TC.NOMBRE AND U.RESTaurante = TC.RESTaurante)on U.RESTaurante = PRR.RESTaurante AND U.NOMBRE=PRR.NOMBRE)
JOIN PERSONA PS ON PS.NUMERO_ID = PRR.USUARIO);

```

Salida de Script x Resultado de la Consulta x Explicación del Plan x

SQL | 1,807 segundos

| OPERATION | OBJECT_NAME | OPTIONS | CARDINALITY | COST |
|---------------------------------------|-------------------|------------------------|-------------|------|
| SELECT STATEMENT | | | | 1 |
| NESTED LOOPS | | | | 1 |
| NESTED LOOPS | | | | 1 |
| NESTED LOOPS | | | | 1 |
| NESTED LOOPS | | | | 1 |
| TABLE ACCESS | ORDEN_PRODUCTO | BY INDEX ROWID BATCHED | 2 | 0 |
| INDEX | ORDEN_PRODUCTO_PK | RANGE SCAN | 1 | 0 |
| Access Predicates | | | | |
| AND | | | | |
| PR.RESTaurante='bieigvvsqyybdjddzvww' | | | | |
| PR.FECHA >= 7449080400000 | | | | |
| PR.FECHA <= 7449166740000 | | | | |
| Filter Predicates | | | | |
| AND | | | | |
| PR.FECHA >= 7449080400000 | | | | |
| PR.FECHA <= 7449166740000 | | | | |
| TABLE ACCESS | PERSONA | BY INDEX ROWID | 1 | 0 |
| INDEX | PERSONA_1_PK | UNIQUE SCAN | 1 | 0 |
| Access Predicates | | | | |
| PS.NUMERO_ID=PR.USUARIO | | | | |
| TABLE ACCESS | PRODUCTO | BY INDEX ROWID | 1 | 0 |
| INDEX | PRODUCTO_PK | UNIQUE SCAN | 1 | 0 |
| Access Predicates | | | | |
| AND | | | | |
| U.NOMBRE=PR.NOMBRE | | | | |
| U.RESTaurante='bieigvvsqyybdjddzvww' | | | | |
| INDEX | TIPO_COMIDA_PK | UNIQUE SCAN | 1 | 0 |
| Access Predicates | | | | |
| AND | | | | |
| TC.RESTaurante='bieigvvsqyybdjddzvww' | | | | |
| U.NOMBRE=TC.NOMBRE | | | | |
| TABLE ACCESS | TIPO_COMIDA | BY INDEX ROWID | 1 | 0 |

Distribución de los datos:

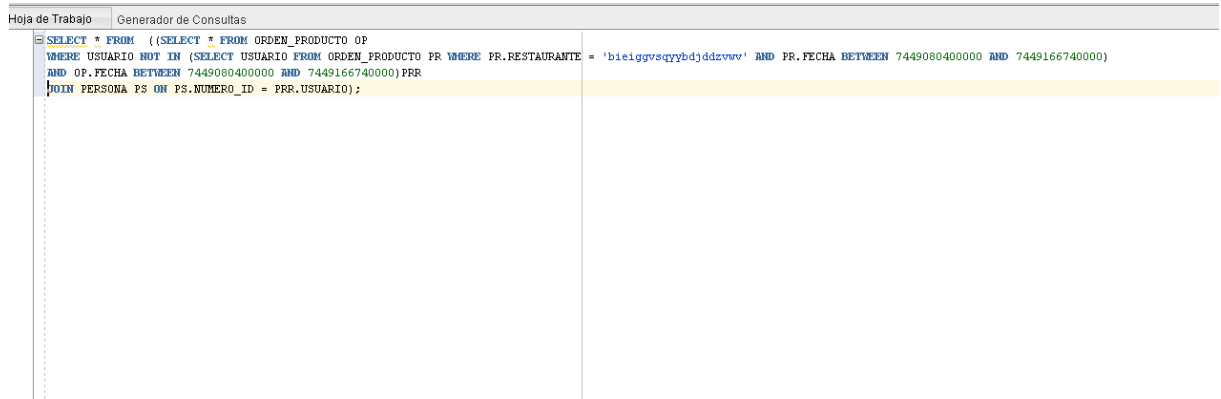
No se genera mucha variación, debido a que a la hora de insertar fuimos muy ordenados, ejemplo creamos para que a cada restaurante le correspondiera 1000 productos, que cada usuario tuviera 3 ordenes, por cosas como la anterior podemos notar que no se genera mucha variación.

- RFC10:

Para este requerimiento utilizamos la siguiente sentencia:

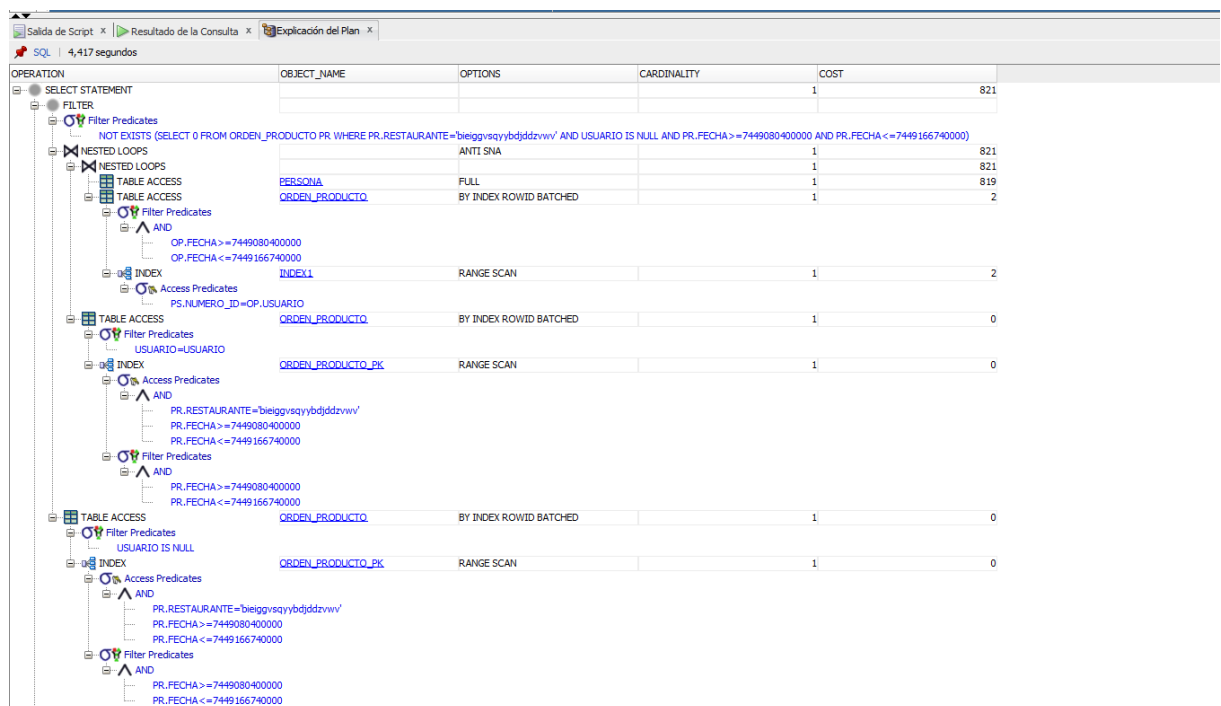


Tiempo:



| RESTAURANTE | NOMBRE | FECHA | MESA | CAMBIOS | USUARIO | NUMERO_ID | NOMBRE_1 |
|-------------|---|---------------|------|---------|---------|-----------|---|
| 1 | o j h d x o t u f n u q w e t x r x k q u t g f l j x i j h a k q c p p s | 7449088915778 | 2 | (null) | 84704 | 84704 | x i q v n g o q u i t a d y i m a g o |
| 2 | h p q j g a x e c g t g b a g k w g 1 | 7449080400001 | 1 | (null) | 1706 | 1706 | o f t x b h l y r h i s l w x q t k x |
| 3 | c x n b o c p g g i v i v c k w x d z k r j y j x r a u h k d t f n h p c s r | 7449165576768 | 168 | (null) | 84718 | 84718 | q a s m q c z t n k k q e p g a m n r |
| 4 | c s k o q f j w z z f y t a a b b t l k r p v p q e k y o x j k e p g k e | 7449165576768 | 168 | (null) | 84719 | 84719 | z o a u o q k v v d f l d a m o m q b u |
| 5 | f c m x g r l a c a m y t v q i g y v k r j v v c b t s t n j p p s k l i h | 7449165576768 | 168 | (null) | 84717 | 84717 | s k n g d v b n l z g r j n e i g o n t |

Plan de consulta:



Distribución de datos:

No se genera mucha variación, debido a que a la hora de insertar fuimos muy ordenados, ejemplo creamos para que a cada restaurante le correspondiera 1000 productos, que cada usuario tuviera 3 ordenes, por cosas como la anterior podemos notar que no se genera mucha variación.

- RFC11:

Para este requerimiento utilizamos la siguiente sentencia:

| Hoja de Trabajo | Generador de Consultas |
|--|------------------------|
| <pre> SELECT * FROM (SELECT RES1.* FROM (SELECT COUNT(USUARIO) AS NUMERO, RESTAURANTE FROM ORDEN_PRODUCTO WHERE ON RES1.NOMBRE=TAB1.RESTAURANTE), (SELECT RES2.* FROM (SELECT COUNT(USUARIO) AS NUMERO, RESTAURANTE FROM ORDEN_PRODUCTO WHERE ON RES2.NOMBRE=TAB2.RESTAURANTE), (SELECT RES3.* FROM (SELECT COUNT(USUARIO) AS NUMERO, NOMBRE, RESTAURANTE FROM ORDEN_PRODUC JOIN PRODUCTO RES3 ON RES3.NOMBRE= TAB3.NOMBRE AND RES3.RESTAURANTE = TAB3.RESTAURANTE), (SELECT RES4.* FROM (SELECT COUNT(USUARIO) AS NUMERO, NOMBRE, RESTAURANTE FROM ORDEN_PRODUC JOIN PRODUCTO RES4 ON RES4.NOMBRE= TAB4.NOMBRE AND RES4.RESTAURANTE = TAB4.RESTAURANTE); </pre> | |

Tiempo:

| Hoja de Trabajo | Generador de Consultas |
|---|------------------------|
| <pre> SELECT * FROM (SELECT RES1.* FROM (SELECT COUNT(USUARIO) AS NUMERO, RESTAURANTE FROM ORDEN_PRODUCTO WHERE rownum = 1 AND FECHA BETWEEN 7449080400000 AND 7449166740000 GROUP BY RESTAURANTE ORDER BY NUMERO DESC) TAB1 JO ON RES1.NOMBRE=TAB1.RESTAURANTE), (SELECT RES2.* FROM (SELECT COUNT(USUARIO) AS NUMERO, RESTAURANTE FROM ORDEN_PRODUCTO WHERE rownum = 1 AND FECHA BETWEEN 7449080400000 AND 7449166740000 GROUP BY RESTAURANTE ORDER BY NUMERO ASC)TAB2 JOIN ON RES2.NOMBRE=TAB2.RESTAURANTE), (SELECT RES3.* FROM (SELECT COUNT(USUARIO) AS NUMERO, NOMBRE, RESTAURANTE FROM ORDEN_PRODUCTO WHERE rownum = 1 AND FECHA BETWEEN 7449080400000 AND 7449166740000 GROUP BY NOMBRE, RESTAURANTE ORDER BY NUMER JOIN PRODUCTO RES3 ON RES3.NOMBRE= TAB3.NOMBRE AND RES3.RESTAURANTE = TAB3.RESTAURANTE), (SELECT RES4.* FROM (SELECT COUNT(USUARIO) AS NUMERO, NOMBRE, RESTAURANTE FROM ORDEN_PRODUCTO WHERE rownum = 1 AND FECHA BETWEEN 7449080400000 AND 7449166740000 GROUP BY NOMBRE, RESTAURANTE ORDER BY NUMER JOIN PRODUCTO RES4 ON RES4.NOMBRE= TAB4.NOMBRE AND RES4.RESTAURANTE = TAB4.RESTAURANTE); </pre> | |

| Explicación del Plan x Salida de Script x Resultado de la Consulta x | | | | | | | | | | | | |
|--|-----------------------|--------------------|------|-------|---------------------|-----------------------|--------------------|---------------|-------|---------|-----------------|----------|
| Todas las Filas Recuperadas: 1 en 0,008 segundos | | | | | | | | | | | | |
| NOMBRE | ZONA_1 | TIPO_COMIDA_1 | WEB | CLAVE | REPRESENTANTE | NOMBRE_1 | ZONA_1 | TIPO_COMIDA_1 | WEB_1 | CLAVE_1 | REPRESENTANTE_1 | NOMBRE_2 |
| 1 hpqjgaaxecgtgbagkwg | 8 okiwtgubyzhustpnjza | ugijwfkjnkziszbkwi | 9429 | 1445 | hpqjgaaxecgtgbagkwg | 8 okiwtgubyzhustpnjza | ugijwfkjnkziszbkwi | 9429 | 1445 | 1 | | |

Plan de consulta:

| OPERATION | OBJECT_NAME | OPTIONS | CARDINALITY | COST | |
|------------------------|-------------|----------|-------------|------|-------|
| SELECT STATEMENT | | | | 1 | 15412 |
| NESTED LOOPS | | | | 1 | 15412 |
| NESTED LOOPS | | | | 1 | 15412 |
| NESTED LOOPS | | | | 1 | 15412 |
| NESTED LOOPS | | | | 1 | 15411 |
| NESTED LOOPS | | | | 1 | 15410 |
| NESTED LOOPS | | | | 1 | 15410 |
| NESTED LOOPS | | | | 1 | 15410 |
| NESTED LOOPS | | | | 1 | 11558 |
| NESTED LOOPS | | | | 1 | 7705 |
| VIEW | | | | 1 | 3853 |
| SORT | | ORDER BY | | 1 | 3853 |
| HASH | | GROUP BY | | 1 | 3853 |
| COL | | STOPKEY | | | |
| Filter Predicates | | | | | |
| ROWNUM=1 | | | | | |
| ORDEN_PRODUCTO | | FULL | | 156 | 3851 |
| Filter Predicates | | | | | |
| AND | | | | | |
| FECHA >= 7449080400000 | | | | | |
| FECHA <= 7449166740000 | | | | | |
| VIEW | | | | 1 | 3853 |
| SORT | | ORDER BY | | 1 | 3853 |
| SORT | | GROUP BY | | 1 | 3853 |
| COL | | STOPKEY | | | |
| Filter Predicates | | | | | |
| ROWNUM=1 | | | | | |
| ORDEN_PRODUCTO | | FULL | | 156 | 3851 |
| Filter Predicates | | | | | |
| AND | | | | | |
| FECHA >= 7449080400000 | | | | | |
| FECHA <= 7449166740000 | | | | | |
| VIEW | | | | 1 | 3853 |
| SORT | | ORDER BY | | 1 | 3853 |
| SORT | | GROUP BY | | 1 | 3853 |
| COUNT | | STOPKEY | | | |
| Filter Predicates | | | | | |
| ROWNUM=1 | | | | | |
| ORDEN_PRODUCTO | | FULL | | 156 | 3851 |
| Filter Predicates | | | | | |
| AND | | | | | |
| FECHA >= 7449080400000 | | | | | |
| FECHA <= 7449166740000 | | | | | |
| VIEW | | | | 1 | 3853 |
| SORT | | ORDER BY | | 1 | 3853 |

Distribución de datos:

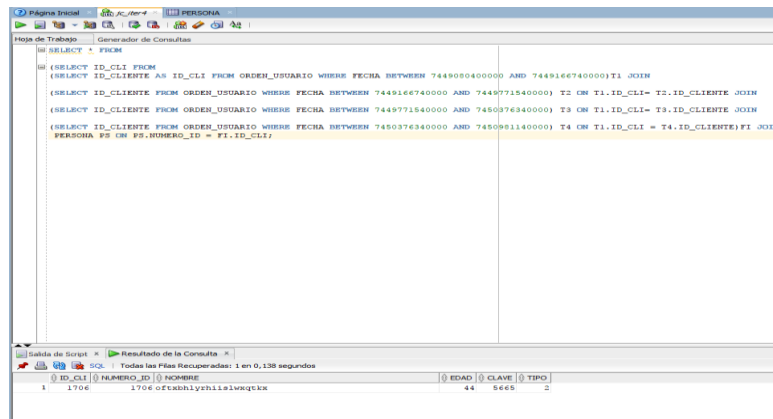
No se genera mucha variación, debido a que a la hora de insertar fuimos muy ordenados, ejemplo creamos para que a cada restaurante le correspondiera 1000 productos, que cada usuario tuviera 3 ordenes, por cosas como la anterior podemos notar que no se genera mucha variación.

- RFC12(A):

Para este requerimiento utilizamos la siguiente sentencia:

| | |
|---|--|
| COMMIT; SELECT * FROM (SELECT ID_CLI FROM (SELECT ID_CLIENTE AS ID_CLI FROM ORDEN_USUARIO WHERE FECHA BETWEEN 7449080400000 AND 7449166740000) T1 JOIN (SELECT ID_CLIENTE FROM ORDEN_USUARIO WHERE FECHA BETWEEN 7449166740000 AND 7449771540000) T2 ON T1.ID_CLI= T2.ID_CLIENTE JOIN (SELECT ID_CLIENTE FROM ORDEN_USUARIO WHERE FECHA BETWEEN 7449771540000 AND 7450376340000) T3 ON T1.ID_CLI= T3.ID_CLIENTE JOIN (SELECT ID_CLIENTE FROM ORDEN_USUARIO WHERE FECHA BETWEEN 7450376340000 AND 7450981140000) T4 ON T1.ID_CLI = T4.ID_CLIENTE) FI JOIN PERSONA PS ON PS.NUMERO_ID = FI.ID_CLI; | |
|---|--|

Tiempo:



Plan de consulta:

| OPERATION | OBJECT_NAME | OPTIONS | CARDINALITY | COST |
|--|-------------|----------------|-------------|------|
| SELECT STATEMENT | | | 1 | 12 |
| HASH JOIN | | | 1 | 12 |
| Access Predicates ID_CLIENTE=ID_CLIENTE | | | | |
| HASH JOIN | | | 1 | 9 |
| Access Predicates ID_CLIENTE=ID_CLIENTE | | | | |
| NESTED LOOPS | | | | 1 |
| NESTED LOOPS | | | 13 | 6 |
| HASH JOIN | | | 13 | 6 |
| Access Predicates ID_CLIENTE=ID_CLIENTE | | | | |
| INDEX ORDEN_USUARIOS_PK | | RANGE SCAN | 13 | 3 |
| Access Predicates AND FECHA >= 744908040000 FECHA <= 7449166740000 | | | | |
| INDEX ORDEN_USUARIOS_PK | | RANGE SCAN | 73 | 3 |
| Access Predicates AND FECHA >= 7449166740000 FECHA <= 7449771540000 | | | | |
| INDEX PERSONA_I_PK | | UNIQUE SCAN | 1 | 0 |
| Access Predicates PS.NUMERO_ID=ID_CLIENTE | | | | |
| TABLE ACCESS PERSONA | | BY INDEX ROWID | 1 | 0 |
| INDEX ORDEN_USUARIOS_PK | | RANGE SCAN | 79 | 3 |
| Access Predicates AND FECHA >= 7449771540000 FECHA <= 7450376340000 | | | | |
| INDEX ORDEN_USUARIOS_PK | | RANGE SCAN | 94 | 3 |
| Access Predicates AND FECHA >= 7450376340000 FECHA <= 7450981140000 | | | | |

Distribución de datos:

No se genera mucha variación, debido a que a la hora de insertar fuimos muy ordenados, ejemplo creamos para que a cada restaurante le correspondiera 1000 productos, que cada usuario tuviera 3 ordenes, por cosas como la anterior podemos notar que no se genera mucha variación.

- RFC12(B):

Para este requerimiento utilizamos la siguiente sentencia:

| | | | | | | | | | |
|---|-----------------------|---------------|------------|---------|--|--------|-----------------------|-----------------------|--|
| COMMIT; | | | | | | | | | |
| SELECT * FROM (SELECT * FROM (ORDEN_PRODUCTO PPR JOIN (SELECT PRECIO, RESTAURANTE, NOMBRE, TIPO FROM PRODUCTO)U ON U.RESTAURANTE = PPR.RESTAURANTE AND U.NOMBRE=PPR.NOMBRE) WHERE U.TIPO =2 AND U.PRECIO >= 33000) TOT JOIN PERSONA PS ON PS.NUMERO_ID = TOT.USUARIO; | | | | | | | | | |
| Se han recuperado 450 filas en 8,701 segundos | | | | | | | | | |
| QCSJ_C000000000500000 | QCSJ_C000000000500002 | FECHA | MESA | CAMBIOS | USUARIO | PRECIO | QCSJ_C000000000500001 | QCSJ_C000000000500003 | |
| 1 vdnqtczfdwrtreholcth hhhvbaejljcpbmgsdesh | | 1002017401108 | 141 (null) | 1058 | 40000 vdnqtczfdwrtreholcth hhhvbaejljcpbmgsdesh | | | | |
| 2 zfgukigxxtzmjbtjhrus nbirdupnukotioxfmagsz | | 1001392312197 | 87 (null) | 787 | 40000 zfgukigxxtzmjbtjhrus nbirdupnukotioxfmagsz | | | | |

Tiempo:

| | | | | | | | | | |
|--|-----------|----------------------|------|-------|------|--|--|--|--|
| SELECT * FROM (SELECT USUARIO FROM (SELECT USUARIO, MIN(PRECIO) AS PREC FROM (ORDEN_PRODUCTO PPR JOIN (SELECT TIPO, PRECIO, RESTAURANTE, NOMBRE FROM PRODUCTO WHERE TIPO =2)U ON U.RESTAURANTE = PPR.RESTAURANTE AND U.NOMBRE=PPR.NOMBRE) GROUP BY PPR.USUARIO WHERE PREC >= 34000) J1 JOIN PERSONA PS ON PS.NUMERO_ID = J1.USUARIO); | | | | | | | | | |
| Se han recuperado 50 filas en 1,749 segundos | | | | | | | | | |
| USUARIO | NUMERO_ID | NOMBRE | EDAD | CLAVE | TIPO | | | | |
| 1 266931 | 266931 | renohdyopchtkhkiwlf | 42 | 2175 | 0 | | | | |
| 2 21412 | 21412 | ymajutkicmrcwvvhhl | 35 | 2510 | 1 | | | | |
| 3 122703 | 122703 | clhrjpuqavlfhpejqtcj | 37 | 5469 | 0 | | | | |
| 4 227161 | 227161 | kfdxbxtapwqdfpdjdm | 31 | 9427 | 1 | | | | |

Plan de consulta:

| | | | | | | | | | |
|-----------------------------|--|--|--|--|----------------|--|--|--|--|
| SELECT * FROM | | | | | 0,456 segundos | | | | |
| OPERATION | | | | | OBJECT_NAME | | | | |
| SELECT STATEMENT | | | | | OPTIONS | | | | |
| FILTER | | | | | CARDINALITY | | | | |
| MIN(PRECIO)>=34000 | | | | | COST | | | | |
| HASH | | | | | 1 | | | | |
| NESTED LOOPS | | | | | 853 | | | | |
| NESTED LOOPS | | | | | GROUP BY | | | | |
| HASH JOIN | | | | | 1 | | | | |
| PS.NUMERO_ID=PPR.USUARIO | | | | | 1108 | | | | |
| NESTED LOOPS | | | | | 852 | | | | |
| STATISTICS COLLECTOR | | | | | 1011954 | | | | |
| TABLE ACCESS PERSONA | | | | | 821 | | | | |
| TABLE ACCESS ORDEN_PRODUCTO | | | | | 1 | | | | |
| INDEX | | | | | 819 | | | | |
| PS.NUMERO_ID=PPR.USUARIO | | | | | 1157510082385 | | | | |
| TABLE ACCESS ORDEN_PRODUCTO | | | | | 2 | | | | |
| INDEX PRODUCTO_PK | | | | | 7 | | | | |
| NESTED LOOPS | | | | | 1157510082385 | | | | |
| TABLE ACCESS RESTAURANTE | | | | | 1 | | | | |
| TABLE ACCESS PRODUCTO | | | | | 0 | | | | |
| TIPO=2 | | | | | 0 | | | | |

Distribución de datos:

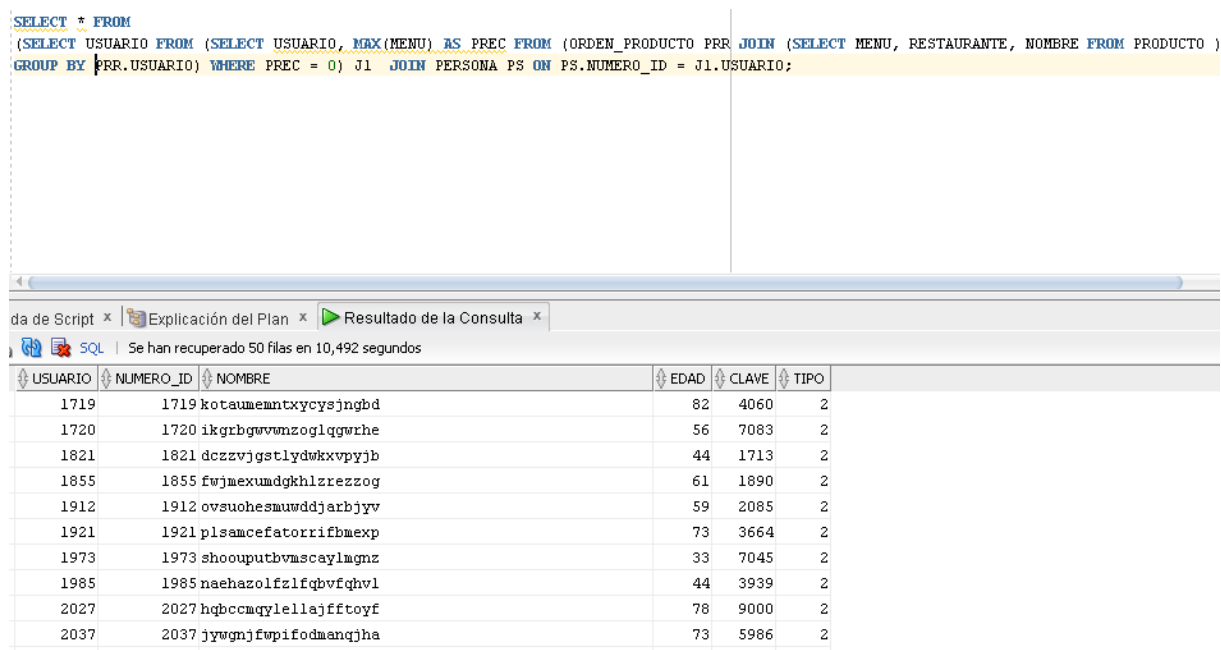
No se genera mucha variación, debido a que a la hora de insertar fuimos muy ordenados, ejemplo creamos para que a cada restaurante le correspondiera 1000 productos, que cada usuario tuviera 3 ordenes, por cosas como la anterior podemos notar que no se genera mucha variación.

- RFC12(C):

Para este requerimiento utilizamos la siguiente sentencia:

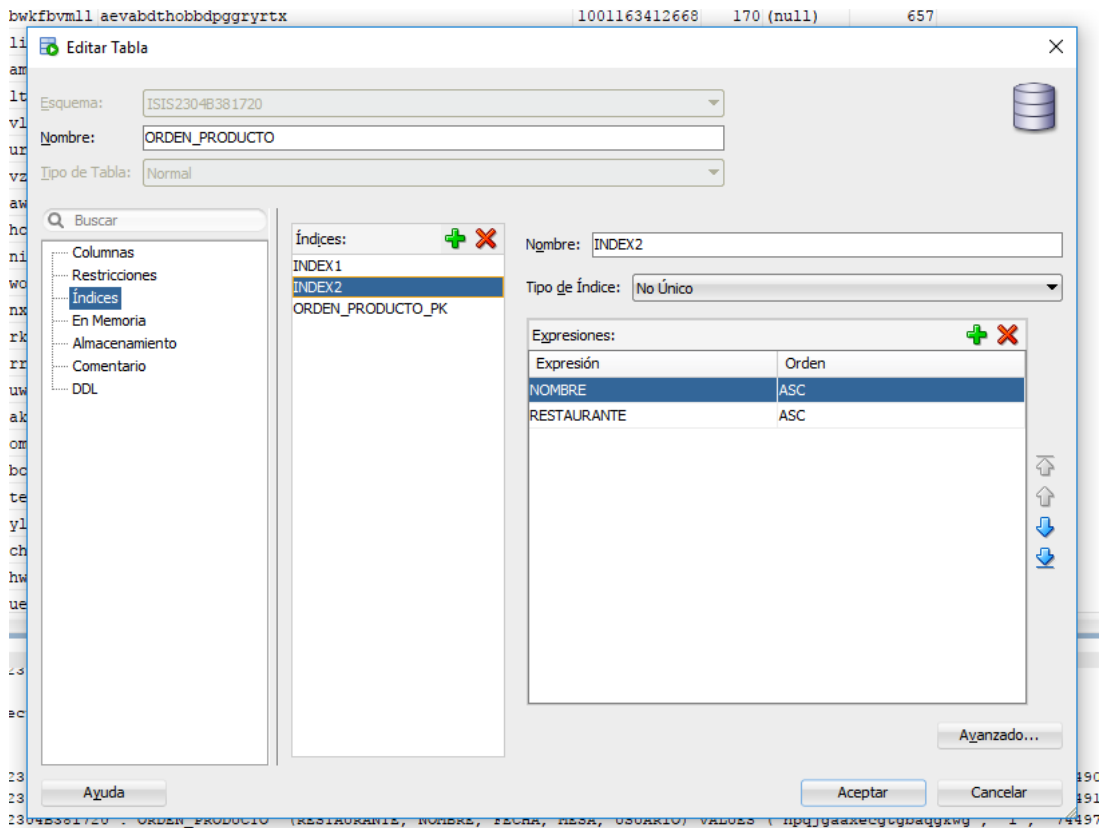


Tiempo:



Plan de consulta:

primary key NUMERO_ID y el JOIN se realiza por estas dos columnas, de manera que según lo visto en clase si ponemos índices sobre ambas columnas que realizan el JOIN se podrá realizar por el algoritmo de 2-pass index sort lo cual nos optimiza mucho el tiempo.



Índice 2: Este índice se creó sobre el atributo NOMBRE, RESTAURANTE(B+,secundario), esto debido a que realizaremos un JOIN con la tabla PRODUCTO y está ya a su vez tiene un índice en su primary key NOMBRE, RESTAURANTE y el JOIN se realiza por estas dos columnas, de manera que según lo visto en clase si ponemos índices sobre ambas columnas que realizan el JOIN se podrá realizar por el algoritmo de 2-pass index sort lo cual nos optimiza mucho el tiempo. Además de que estamos buscando un RESTAURANTE en particular la selectividad es alta.

Para las demás TABLAS

Debido a que supimos distribuir los datos de manera que en las tablas sobre las cuales se implementan los RFC9-RFC12 hubiera más cantidad de datos y en la otras una distribución de datos menor, nos bastó con poner los índices solamente para la tabla ORDEN_PRODUCTO de una manera inteligente tal que las búsquedas lográramos reducir bien los tiempos. Vale aclarar que usamos a nuestro favor para JOINS los índices que se creaban por defecto para las llaves primarias.