

REQUERIMIENTO

RC3

RC4

RC9

RC10

RC11

RC12

1

2

3

1

1

2

3

4

SENTENCIA SQL

```
SELECT * FROM USUARIO WHERE CEDULA=id; (CON ID = 17)
```

```
SELECT * FROM (PREFERENCIAS NATURAL JOIN CATEGORIAPRODUCTO)  
natural join (ZONA INNER join TIPODECOMIDA ON IDTIPO=IDTIPOCOMIDA)  
WHERE CEDULA= id;
```

```
SELECT * FROM pedido WHERE CEDULA=id;
```

```
SELECT * FROM PRODUCTO NATURAL JOIN(SELECT * FROM (SELECT  
MAX(TOTALMENUS) AS TOTALMENUS FROM(SELECT IDPRODUCTO,  
COUNT(IDMENU) AS TOTALMENUS FROM (PRODUCTO P NATURAL JOIN  
TIENEPRODUCTO T) GROUP BY IDPRODUCTO)) NATURAL JOIN (SELECT  
IDPRODUCTO, COUNT(IDMENU) AS TOTALMENUS FROM (PRODUCTO P  
NATURAL JOIN TIENEPRODUCTO T) GROUP BY IDPRODUCTO));
```

```
SELECT CEDULA, NOMBRE, ROL, EMAIL  
FROM ( SELECT IDPRODUCTO, LOCAL  
      FROM (OFRECEPRODUCTO NATURAL JOIN RESTAURANTE))a1  
      INNER JOIN  
      ( SELECT USUARIO.CEDULA, NOMBRE, ROL, EMAIL, IDPRODUCTO, LOCAL,  
IDPEDIDO  
      FROM (USUARIO RIGHT OUTER JOIN PEDIDO ON USUARIO.CEDULA=  
PEDIDO.CEDULA) NATURAL JOIN PEDIDOPRODUCTO  
      WHERE ROL='CLIENTE' AND (FECHAYHORA BETWEEN  
TO_DATE('01/01/17') AND TO_DATE('31/12/17'))))us ON  
a1.LOCAL=us.LOCAL  
WHERE us.LOCAL=366520  
GROUP BY CEDULA, NOMBRE, ROL, EMAIL;
```

```

SELECT CEDULA, NOMBRE, ROL, EMAIL FROM USUARIO WHERE
ROL='CLIENTE' AND (CEDULA) NOT IN(SELECT CEDULA
    FROM ( SELECT IDPRODUCTO, LOCAL
        FROM (OFRECEPRODUCTO NATURAL JOIN RESTAURANTE))a1
    INNER JOIN
        ( SELECT USUARIO.CEDULA, NOMBRE, ROL, EMAIL, IDPRODUCTO,
LOCAL, IDPEDIDO
    FROM (USUARIO RIGHT OUTER JOIN PEDIDO ON
USUARIO.CEDULA= PEDIDO.CEDULA) NATURAL JOIN PEDIDOPRODUCTO
        WHERE ROL='CLIENTE' AND (FECHAYHORA BETWEEN
TO_DATE('01/01/17') AND TO_DATE('31/12/17'))))us ON
a1.LOCAL=us.LOCAL
    WHERE us.LOCAL=366520
    GROUP BY CEDULA, NOMBRE, ROL, EMAIL);

```

```

SELECT * FROM(SELECT IDPRODUCTO, COUNT(IDPRODUCTO) AS TOTAL
FROM ((SELECT IDPEDIDO, TO_CHAR(FECHAYHORA,'DY' ) AS DY FROM
PEDIDO )NATURAL JOIN PEDIDOPRODUCTO) NATURAL JOIN PRODUCTO
WHERE DY='DOM' GROUP BY IDPRODUCTO)NATURAL JOIN(SELECT
MAX(TOTAL)AS TOTAL FROM(SELECT IDPRODUCTO, COUNT(IDPRODUCTO)
AS TOTAL FROM ((SELECT IDPEDIDO, TO_CHAR(FECHAYHORA,'DY' ) AS DY
FROM PEDIDO )NATURAL JOIN PEDIDOPRODUCTO) NATURAL JOIN
PRODUCTO WHERE DY='DOM' GROUP BY IDPRODUCTO) );

```

```

SELECT * FROM(SELECT LOCAL, COUNT(LOCAL) AS TOTAL FROM ((SELECT
IDPEDIDO, TO_CHAR(FECHAYHORA,'DY' ) AS DY FROM PEDIDO )NATURAL
JOIN PEDIDOPRODUCTO) NATURAL JOIN RESTAURANTE WHERE DY='DOM'
GROUP BY LOCAL)NATURAL JOIN(SELECT MAX(TOTAL)AS TOTAL
FROM(SELECT LOCAL, COUNT(LOCAL) AS TOTAL FROM ((SELECT IDPEDIDO,
TO_CHAR(FECHAYHORA,'DY' ) AS DY FROM PEDIDO )NATURAL JOIN
PEDIDOPRODUCTO) NATURAL JOIN RESTAURANTE WHERE DY='DOM'
GROUP BY LOCAL) );

```

```
SELECT * FROM (SELECT IDPRODUCTO, COUNT(IDPRODUCTO) AS TOTAL  
FROM ((SELECT IDPEDIDO, TO_CHAR(FECHAYHORA, 'DY') AS DY FROM  
PEDIDO) NATURAL JOIN PEDIDOPRODUCTO) NATURAL JOIN PRODUCTO  
WHERE DY = 'DOM' GROUP BY IDPRODUCTO) NATURAL JOIN (SELECT  
MIN(TOTAL) AS TOTAL FROM (SELECT IDPRODUCTO, COUNT(IDPRODUCTO)  
AS TOTAL FROM ((SELECT IDPEDIDO, TO_CHAR(FECHAYHORA, 'DY') AS DY  
FROM PEDIDO) NATURAL JOIN PEDIDOPRODUCTO) NATURAL JOIN  
PRODUCTO WHERE DY = 'DOM' GROUP BY IDPRODUCTO));
```

```
SELECT * FROM (SELECT LOCAL, COUNT(LOCAL) AS TOTAL FROM ((SELECT  
IDPEDIDO, TO_CHAR(FECHAYHORA, 'DY') AS DY FROM PEDIDO) NATURAL  
JOIN PEDIDOPRODUCTO) NATURAL JOIN RESTAURANTE WHERE DY = 'DOM'  
GROUP BY LOCAL) NATURAL JOIN (SELECT MIN(TOTAL) AS TOTAL  
FROM (SELECT LOCAL, COUNT(LOCAL) AS TOTAL FROM ((SELECT IDPEDIDO,  
TO_CHAR(FECHAYHORA, 'DY') AS DY FROM PEDIDO) NATURAL JOIN  
PEDIDOPRODUCTO) NATURAL JOIN RESTAURANTE WHERE DY = 'DOM'  
GROUP BY LOCAL));
```

```

SELECT CEDULA, NOMBRE, EMAIL, ROL
FROM (((SELECT CEDULA,NOM AS NOMBRE, ROL, EMAIL
FROM ((PEDIDO NATURAL JOIN (SELECT CEDULA, ROL, EMAIL, NOMBRE AS
NOM FROM USUARIO)) INNER JOIN (PEDIDOPRODUCTO NATURAL JOIN
(PRODUCTO NATURAL JOIN OFRECEPRODUCTO)) ON
PEDIDO.IDPEDIDO=PEDIDOPRODUCTO.IDPEDIDO)
WHERE IDCATEGORIA=2 AND PRECIO>=36885.855 )
UNION ALL (SELECT CEDULA, NOMBRE,EMAIL,ROL FROM(SELECT
SUM(NUMWEEKS) AS TOTALWEEKS FROM (SELECT
COUNT(DISTINCT(WEEK)) AS NUMWEEKS,YEAR FROM(SELECT IDPEDIDO,
EXTRACT(YEAR FROM FECHAYHORA) AS YEAR,
to_number(to_char(FECHAYHORA,'WW')) AS WEEK FROM (PEDIDO ))
GROUP BY YEAR)) NATURAL JOIN (SELECT NOMBRE, ROL, CEDULA, EMAIL,
SUM(NUMWEEKS) AS TOTALWEEKS FROM (SELECT COUNT(*) AS
NUMWEEKS, NOMBRE, CEDULA, EMAIL, ROL, YEAR FROM (SELECT
EXTRACT(YEAR FROM FECHAYHORA)AS YEAR, CEDULA, ROL, NoMBRE,
EMAIL , to_number(to_char(FECHAYHORA,'WW')) AS WEEK FROM (PEDIDO
NATURAL JOIN USUARIO)) GROUP BY NOMBRE, ROL, CEDULA, EMAIL, YEAR)
GROUP BY NOMBRE, ROL, CEDULA, EMAIL))))
UNION ALL SELECT CEDULA, NOMBRE, ROL, EMAIL FROM (PEDIDO
NATURAL JOIN PEDIDOPRODUCTO) NATURAL JOIN USUARIO
WHERE CEDULA not IN (SELECT CEDULA FROM (PEDIDO NATURAL JOIN
PEDIDOMENU) NATURAL JOIN USUARIO CEDULA )
GROUP BY CEDULA, NOMBRE, EMAIL, ROL ;

```

DISTRIBUCIÓN DE LOS PARÁMETROS DE ENTRADA

El parámetro es el id por lo que es único, no se repite en la búsqueda, y debido a la cantidad de datos en esta tabla, su distribución es igual a la de cualquier otra tupla. Respecto a la posibilidad de cambiar el tamaño de la respuesta lo unico que se podría hacer es provocar un resultado vacío con un id inexistente, como por ejemplo 5 (millones). No se puede incrementar ya que el requerimiento solo habla de la información de un solo cliente.

La distribución de los resultados de este requerimiento dependerían de la cantidad de menús existentes, como también de la cantidad de productos y finalmente la intersección de estos dos, es decir, el contenido de los menus respecto a los productos. No hay escenarios de prueba, ya que como ya fue explicado, este resultado es constante.

La distribución de los datos de entrada se estima uniforme para cada producto de cada restaurante, puesto que al tener en cuenta el tamaño de las tablas de los productos, de los restaurantes y de los pedidos, es bastante poco probable que más de un cliente hayan consumido lo mismo en el mismo rango de fechas.

La distribución de los datos de entrada se estima uniforme para cada producto de cada restaurante, puesto que al tener en cuenta el tamaño de las tablas de los productos, de los restaurantes y de los pedidos, es bastante poco probable que más de un cliente hayan consumido lo mismo en el mismo rango de fechas.

La distribución de los parámetros de entrada en este requerimiento esta determinada gracias a los pedidos, ya que en ellos y la cantidad de veces que un producto haga parte de una de estas tuplas es donde se ve realmente la cardinalidad.

La distribución de los parámetros de entrada en este requerimiento esta determinada gracias a los pedidos, ya que en ellos y la cantidad de veces que un producto haga parte de una de estas tuplas es donde se ve realmente la cardinalidad.

La distribución de los parámetros de entrada en este requerimiento esta determinada gracias a los pedidos, ya que en ellos y la cantidad de veces que un producto haga parte de una de estas tuplas es donde se ve realmente la cardinalidad.

La distribución de los parámetros de entrada en este requerimiento esta determinada gracias a los pedidos, ya que en ellos y la cantidad de veces que un producto haga parte de una de estas tuplas es donde se ve realmente la cardinalidad.

La distribución de los parámetros de entrada en este requerimiento esta determinada gracias a los pedidos, ya que en ellos se puede determinar los mejores clientes, lo que pidieron, cuando lo pidieron, el costo de lo que pidieron y cada cuanto lo pidieron.

VALORES DE PARÁMETROS

ID (USUARIO)

ID (USUARIO)

ID (USUARIO)

Este requerimiento no tiene parámetros de entrada ya que la respuesta es única en el momento de la consulta. En cuanto a los casos de prueba no habría ninguno debido a lo ya mencionado, no habría escenarios a analizar puesto que la respuesta es única y solo habría un único escenario.

En este caso los valores que entran por parámetros son el ROL (usuario), el local del restaurante, el producto del determinado restaurante y los rangos de fechas. Estos resultados varían según todos menos el rol, ya que este tiene que ser una constante debido a lo requerido. Por otra parte, la respuesta se agrandaría si el rango de fechas es mayor, sin embargo, lo que más determina este tamaño de respuesta es el producto y el restaurante ingresados.

En este caso los valores que entran por parametros son el ROL (usuario), el local del restaurante, el producto del determinado restaurante y los rangos de fechas. Estos resultados varían según todos menos el rol, ya que esta tiene que ser una constante debido a lo requerido. Por otra parte, la respuesta se agrandaría si el rango de fechas es mayor, sin embargo, lo que más determina este tamaño de respuesta es el producto y el restaurante ingresados.

Este requerimiento no tiene parámetros de entrada ya que la respuesta es única en el momento de la consulta. En cuanto a los casos de prueba no habría ninguno debido a lo ya mencionado, no habría escenarios a analizar puesto que la respuesta es única y solo habría un único escenario.

Este requerimiento no tiene parámetros de entrada ya que la respuesta es única en el momento de la consulta. En cuanto a los casos de prueba no habría ninguno debido a lo ya mencionado, no habría escenarios a analizar puesto que la respuesta es única y solo habría un único escenario.

Este requerimiento no tiene parámetros de entrada ya que la respuesta es única en el momento de la consulta. En cuanto a los casos de prueba no habría ninguno debido a lo ya mencionado, no habría escenarios a analizar puesto que la respuesta es única y solo habría un único escenario.

Este requerimiento no tiene parámetros de entrada ya que la respuesta es única en el momento de la consulta. En cuanto a los casos de prueba no habría ninguno debido a lo ya mencionado, no habría escenarios a analizar puesto que la respuesta es única y solo habría un único escenario.

Los parámetros de entrada en este caso es el costo de 1.5 SMDV solamente, puesto que el resto se toma como una constante que se da gracias a este valor de entrada.

PLANES OBTENIDOS

SQL | 0,063 segundos

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				1 3
TABLE ACCESS	USUARIO	BY INDEX ROWID	1	3
INDEX	USUARIO_PK	UNIQUE SCAN	1	2
Access Predicates				
CEDULA=17				
Other XML				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				1 4
HASH JOIN				1 4
Access Predicates				
IDTIPO=IDTIPOCOMIDA				
NESTED LOOPS			1	4
NESTED LOOPS			1	4
STATISTICS COLLECTOR				
NESTED LOOPS			1	3
NESTED LOOPS			1	3
TABLE ACCESS	PREFERENCIAS	FULL	1	3
Filter Predicates				
PREFERENCIAS.CEDULA=17				
TABLE ACCESS	CATEGORIAPRODUCTO	BY INDEX ROWID	1	0
INDEX	CATEGORIAPRODUCTO_PK	UNIQUE SCAN	1	0
Access Predicates				
PREFERENCIAS.IDCATEGORIA=CATEGORIAPRODUCTO.IDCATEGORIA				
TABLE ACCESS	ZONA	BY INDEX ROWID	1	0
INDEX	ZONA_PK	UNIQUE SCAN	1	0
Access Predicates				
PREFERENCIAS.IDZONA=ZONA.IDZONA				
INDEX	TIPODECOMIDA_PK	UNIQUE SCAN	1	0
Access Predicates				
IDTIPO=IDTIPOCOMIDA				
TABLE ACCESS	TIPODECOMIDA	BY INDEX ROWID	1	1
TABLE ACCESS	TIPODECOMIDA	FULL	1	1
Other XML				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				1 1028
TABLE ACCESS	PEDIDO	FULL	1	1028
Filter Predicates				
CEDULA=17				
Other XML				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				1 5
HASH JOIN				1 5
Access Predicates	PRODUCTO.IDPRODUCTO=from\$_subquery\$_008.IDPRODUCTO			
NESTED LOOPS				1 5
NESTED LOOPS				1 5
STATISTICS COLLECTOR				
NESTED LOOPS				1 4
VIEW				1 2
SORT				1 1
VIEW		AGGREGATE		1 2
HASH		GROUP BY		1 2
HASH JOIN				1 1
Access Predicates				
AND				
P.IDCATEGORIA=T.IDCATEGORIA				
P.IDPRODUCTO=T.IDPRODUCTO				
NESTED				1 1
NES				1 1
index\$_join\$_006				1 0
Access Predicates	ROWID=ROWID			
TIENEPRODUCTO_UK1	FAST FULL SCAN			1 0
TIENEPROD_PK	FAST FULL SCAN			1 0
PRODUCTO_PK	UNIQUE SCAN			1 1
Access Predicates	P.IDPRODUCTO=T.IDPRODUCTO			
TABLEPRODUCTO	BY INDEX ROWID			1 1
Filter Predicates	P.IDCATEGORIA=T.IDCATEGORIA			
TABLEPRODUCTO	FULL			1 1
VIEW				1 2
Filter Predicates	from\$_subquery\$_003.TOTALMENUS=from\$_subquery\$_008.TOTALMENUS			
SORT		GROUP BY		1 2
HASH JOIN				1 1
Access Predicates				
AND				
P.IDCATEGORIA=T.IDCATEGORIA				
P.IDPRODUCTO=T.IDPRODUCTO				
NESTED LOOPS				1 1
NESTED LOC				1 1
STATIST				
VIEWindex\$_join\$_010				1 0
Access Predicates	ROWID=ROWID			
TIENEPRODUCTO_UK1	FAST FULL SCAN			1 0
TIENEPROD_PK	FAST FULL SCAN			1 0
PRODUCTO_PK	UNIQUE SCAN			1 1
Access Predicates	P.IDPRODUCTO=T.IDPRODUCTO			
TABLEACCEPRODUCTO	BY INDEX ROWID			1 1
Filter Predicates	P.IDCATEGORIA=T.IDCATEGORIA			
TABLEACCESS	PRODUCTO	FULL		1 1
INDEX	PRODUCTO_PK	UNIQUE SCAN		1 1
Access Predicates	PRODUCTO.IDPRODUCTO=from\$_subquery\$_008.IDPRODUCTO			
TABLEACCESS	PRODUCTO	BY INDEX ROWID		1 1
TABLEACCESS	PRODUCTO	FULL		1 1

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				1 1926
HASH		GROUP BY		1 1926
FILTER				
Filter Predicates	TO_DATE('31/12/17')>=TO_DATE('01/01/17')			
HASH JOIN				4 1925
Access Predicates	USUARIO.CEDULA=PEDIDO.CEDULA			
NESTED LOOPS				4 1925
NESTED LOOPS				4 1925
STATISTICS COLLECTOR				
HASH JOIN				4 1917
Access Predicates	PEDIDO.IDPEDIDO=PEDIDOPRODUCTO.IDPEDIDO			
NESTED LOOPS				4 1917
STATISTICS COL				
HASH JOIN		SEMI		4 1909
Access Predicates	OFRECEPRODUCTO.LOCAL=PEDIDOPRODUCTO.LOCAL			
TABLEAPEDIDOPRODUCTO	FULL			4 859
Filter Predicates	PEDIDOPRODUCTO.LOCAL=366520			
INDEX	OFRECEPRODUCTO_PK	FAST FULL SCAN		2 1050
Filter Predicates	OFRECEPRODUCTO.LOCAL=366520			
TABLEACCESS	PEDIDO	BY INDEX ROWID		1 2
Filter Predicates				
AND				
PEDIDO.FECHAYHORA>=TO_DATE('01/01/17')				
PEDIDO.FECHAYHORA<=TO_DATE('31/12/17')				
INDEX	PEDIDO_PK	UNIQUE SCAN		1 1

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				
HASH JOIN				
Access Predicates CEDULA=CEDULA		RIGHT ANTI	991397	4087
VIEW	SYS.VW_NSO_1		1	1926
HASH		GROUP BY	1	1926
FILTER				
Filter Predicates TO_DATE(31/12/17)>=TO_DATE(01/01/17)				
NESTED LOOPS		SEMI	4	1925
NESTED LOOPS			4	1917
HASH JOIN		SEMI	4	1909
Access Predicates OFRECEPRODUCTO.LOCAL=PEDIDOPRODUCTO.LOCAL				
TABLE ACCESS	PEDIDOPRODUCTO	FULL	4	859
Filter Predicates PEDIDOPRODUCTO.LOCAL=366520				
INDEX	OFRECEPRODUCTO_PK	FAST FULL SCAN	2	1050
Filter Predicates OFRECEPRODUCTO.LOCAL=366520				
TABLE ACCESS	PEDIDO	BY INDEX ROWID	1	2
Filter Predicates AND PEDIDO.FECHAYHORA>=TO_DATE(01/01/17) PEDIDO.FECHAYHORA<=TO_DATE(31/12/17)				
INDEX	PEDIDO_PK	UNIQUE SCAN	1	1
Access Predicates PEDIDO.IDPEDIDO=PEDIDOPRODUCTO.IDPEDIDO				
TABLE ACCESS	USUARIO	BY INDEX ROWID	991399	2
Filter Predicates USUARIO.ROL='CLIENTE'				
INDEX	USUARIO_PK	UNIQUE SCAN	1	1
Access Predicates USUARIO.CEDULA=PEDIDO.CEDULA				
TABLE ACCESS	USUARIO	FULL	991399	2158
Filter Predicates ROL='CLIENTE'				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				
NESTED LOOPS				
VIEW				
SORT				
VIEW				
HASH				
HASH JOIN				
Access Predicates IDPEDIDO=PEDIDOPRODUCTO.IDPEDIDO				
TABLE ACCESS	PEDIDO	FULL	355040	103
Filter Predicates TO_CHAR(INTERNAL_FUNCTION(FECHAYHORA),'DY')='DOM'				
TABLE ACCESS	PEDIDOPRODUCTO	FULL	1000000	85
VIEW				
Filter Predicates from\$_subquery\$_001.TOTAL=from\$_subquery\$_008.TOTAL				
SORT				
HASH JOIN				
Access Predicates IDPEDIDO=PEDIDOPRODUCTO.IDPEDIDO				
TABLE ACCESS	PEDIDO	FULL	355040	103
Filter Predicates TO_CHAR(INTERNAL_FUNCTION(FECHAYHORA),'DY')='DOM'				
TABLE ACCESS	PEDIDOPRODUCTO	FULL	1000000	85

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				
NESTED LOOPS				
VIEW				
SORT				
VIEW				
HASH				
HASH JOIN				
Access Predicates IDPEDIDO=PEDIDOPRODUCTO.IDPEDIDO				
TABLE ACCESS	PEDIDO	FULL	355040	10
Filter Predicates TO_CHAR(INTERNAL_FUNCTION(FECHAYHORA),'DY')='DOM'				
TABLE ACCESS	PEDIDOPRODUCTO	FULL	1000000	8
VIEW				
Filter Predicates from\$_subquery\$_001.TOTAL=from\$_subquery\$_008.TOTAL				
SORT				
HASH JOIN				
Access Predicates IDPEDIDO=PEDIDOPRODUCTO.IDPEDIDO				
TABLE ACCESS	PEDIDO	FULL	355040	10
Filter Predicates TO_CHAR(INTERNAL_FUNCTION(FECHAYHORA),'DY')='DOM'				
TABLE ACCESS	PEDIDOPRODUCTO	FULL	1000000	8





OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			773490	51776
UNION-ALL				
VIEW			509404	31765
UNION-ALL				
HASH JOIN			503526	14110
Access Predicates PEDIDO.CEDULA=CEDULA				
NESTED LOOPS			503526	14110
NESTED LOOPS				
STATISTICS COLLECTOR				
HASH JOIN			503526	7777
Access Predicates PEDIDO.IDPEDIDO=PEDIDOPRODUCTO.IDPEDIDO				
TABLE ACCESS PEDIDO		FULL	700000	1027
HASH JOIN			503526	5064
Access Predicates PEDIDOPRODUCTO.IDPRODUCTO=OFRECEPRODUCTO.IDPRODUCTO				
HASH JOIN			221423	2846
Access Predicates PRODUCTO.IDPRODUCTO=OFRECEPRODUCTO.IDPRODUCTO				
TABLE ACCESS PRODUCTO		FULL	100426	1642
Filter Predicates PRODUCTO.IDCATEGORIA=2				
TABLE ACCESS OFRECEPRODUCTO		FULL	966246	1201
Filter Predicates OFRECEPRODUCTO.PRECIO>=36885.855				
TABLE ACCESS PEDIDOPRODUCTO		FULL	1000000	858
INDEX USUARIO_PK		UNIQUE SCAN		
Access Predicates PEDIDO.CEDULA=CEDULA				
TABLE ACCESS USUARIO		BY INDEX ROWID	1	2157
TABLE ACCESS USUARIO		FULL	1000059	2157

TIEMPOS

SQL Todas las Filas Recuperadas: 1 en 0,01 segundos				
	CEDULA	NOMBRE	ROL	EMAIL
1	17	usuario17	CLIENTE	email17@correo.com


SQL
Todas las Filas Recuperadas: 1 en 0,013 segundos


	IDZONA	IDCATEGORIA	IDPREFERENCIA	CEDULA	MAXIMO	MINIMO	CATEGORIA	IDTIPOCOMIDA	IDTIPO	TIPO
1	2	2	2	17	100	11	PLATO FUERTE	5	5	MARINA



SQL

Todas las Filas Recuperadas: 1 en 0,042 segundos

	IDPEDIDO	IDPAGO	FECHAYHORA	CEDULA	SERVIDO	IDMESA	PRECIOPEDIDO
1	530316	3	19/07/16	17	F	35	497914

 Todas las Filas Recuperadas: 2 en 0,176 segundos						
IDPRODUCTO	NOMBRE	DESCRIPCIONESPANOL	DESCRIPCIONINGLES	IDCATEGORIA	TOTALMENUS	
1	11producto11	descripcion11 en español	description11 in English	5	1	
2	14producto14	descripcion14 en español	description14 in English	2	1	

 Todas las Filas Recuperadas: 3 en 0,151 segundos				
CEDULA	NOMBRE	ROL	EMAIL	
1	650549 usuario650549	CLIENTE	email650549@correo.com	
2	57523 usuario57523	CLIENTE	email57523@correo.com	
3	92061 usuario92061	CLIENTE	email92061@correo.com	

SQL | Se han recuperado 3.200 filas en 1,904 segundos

	CEDULA	NOMBRE	ROL	EMAIL
3177	104456	usuariol0...		56@correo...
3178	104457	usuariol0...		57@correo...
3179	104458	usuariol0...		58@correo...
3180	104459	usuariol0...		59@correo...
3181	104460	usuariol0...		60@correo...
3182	104461	usuariol0...		61@correo...

Recuento... X

999.996 Filas

Aceptar

SQL | Todas las Filas Recuperadas: 2 en 15,521 segundos

	TOTAL	IDPRODUCTO
1	10	388246
2	10	458028

SQL | Todas las Filas Recuperadas: 1 en 11,581 segundos

	TOTAL	LOCAL
1	10	269474

SQL | Se han recuperado 50 filas en 12,215 segundos

	TOTAL	IDPRODUCTO
1		
2		
3		
4		
5		
6	1	10

Recuento... X

178.590 Filas

Aceptar


SQL | Se han recuperado 50 filas en 11,24 segundos

	TOTAL	LOCAL
1	1	
2	1	
3	1	
4	1	
5	1	
6	1	17

Recuento... X

179.232 Filas

Aceptar

 SQL | Se han recuperado 50 filas en 0,662 segundos

	CEDULA	NOMBRE	EMAIL	ROL
1	584867	usuario584867	email584867@correo.com	CLIENTE
2	869547	usu	9547@correo.com	CLIENTE
3	869547	usu	9547@correo.com	CLIENTE
4	869547	usu	9547@correo.com	CLIENTE
5	130052	usu	0052@correo.com	CLIENTE
6	130052	usu	0052@correo.com	CLIENTE
7	477193	usuario477193	email477193@correo.com	CLIENTE

Recuento... X

468.130 Filas

Aceptar

