RC3

RC4

RC9

RC10

RC11

N° DE SENTENCIA (DEBIDO A QUE EN ALGUNAS OCASIONES SE USÓ MÁS DE UNA)

1

2





SENTENCIA SQL
SERVE I CONTINUE DE LA CONTINUE DE L
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 vacio 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 cantidada 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 cantidada 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 cantidada 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 cantidada 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)	
D (USUANIU)	
ID (USUARIO)	
ID (USUARIO)	p

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 escensarios a analizar puesto que la respuesta es única y solo habría un único escenario.

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.

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 escensarios a analizar puesto que la respuesta es única y solo habría un único escenario.

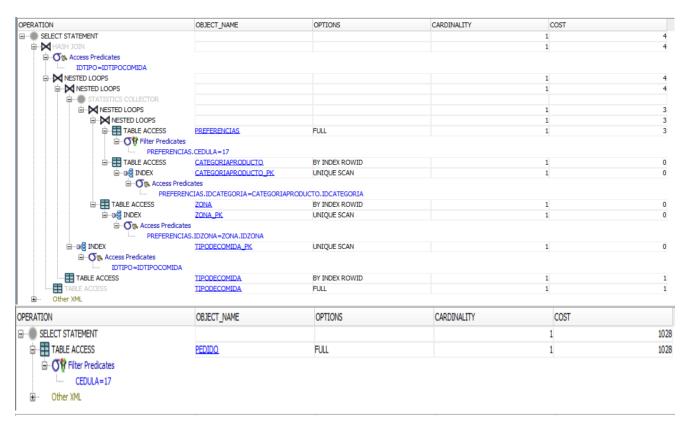
Este requerimiento no tiene parámetros de entrada ya que la respuesta es única en el momento de la consulta. En cuanto a los casos de prueba no habría ninguno debido a lo ya mencionado, no habría escensarios a analizar puesto que la respuesta es única y solo habría un único escenario.

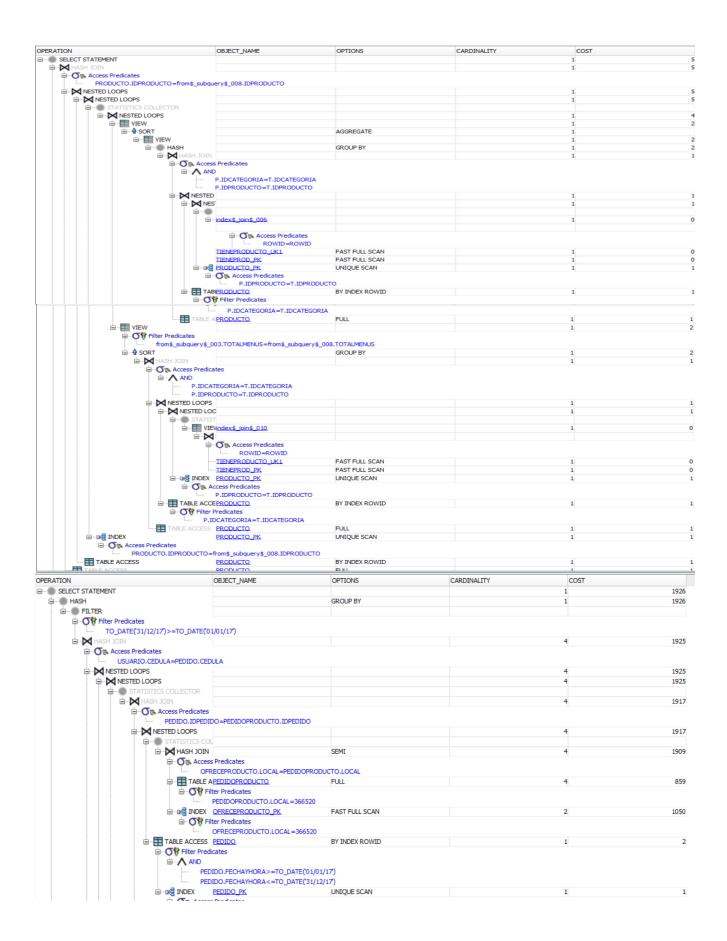
Este requerimiento no tiene parámetros de entrada ya que la respuesta es única en el momento de la consulta. En cuanto a los casos de prueba no habría ninguno debido a lo ya mencionado, no habría escensarios a analizar puesto que la respuesta es única y solo habría un único escenario.

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 escensarios 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.



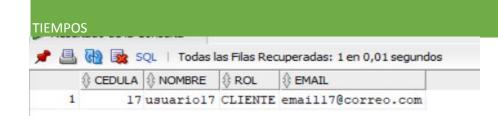


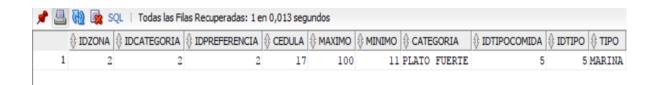


TON	ODWECT MANY	ONTONIC	CARDINALTY	
ITION SELECT STATEMENT	OBJECT_NAME	OPTIONS	CARDINALITY COST 991397	,
HASH JOIN		RIGHT ANTI	991397	,
⊕ O™ Access Predicates				
CEDULA=CEDULA G WIEW	SYS.VW_NSO_1		1	
□ □ HASH	SISTER TOWARD	GROUP BY	1	
⊕ ● FILTER				
Off Filter Predicates				
TO_DATE(31/12/17)>=TO_DATE(01/01/17)		CENT		
		SEMI	4	
⊕ MHASH JOIN		SEMI	4	
□ O™ Access Predicates				
OFRECEPRODUCTO.LOCAL =PEDIDOPRODUCTO.LOCAL				
☐	PEDIDOPRODUCTO	FULL	4	
□ OV Filter Predicates PEDIDOPRODUCTO.LOCAL=366520				
⊟-o-g INDEX	OFRECEPRODUCTO_PK	FAST FULL SCAN	2	
☐ O ♥ Filter Predicates				
OFRECEPRODUCTO.LOCAL=366520				
⊞ TABLE ACCESS	PEDIDO	BY INDEX ROWID	1	
□ ○ ○ ○ ○ ○ ○				
PEDIDO.FECHAYHORA>=TO_DATE('01/01/17')				
PEDIDO.FECHAYHORA<=TO_DATE(31/12/17)				
⊟-0-@ INDEX	PEDIDO_PK	UNIQUE SCAN	1	
☐ O™ Access Predicates				
PEDIDO.IDPEDIDO=PEDIDOPRODUCTO.IDPEDIDO	USUARIO.	BY INDEX ROWID	991399	
Fiter Predicates	usunnad.	DI INDEX KOWID	.aa13aa	
USUARIO.ROL='CLIENTE'				
⊞-a@ INDEX	USUARIO_PK	UNIQUE SCAN	1	
☐ O _M Access Predicates				
USUARIO.CEDULA=PEDIDO.CEDULA	USUARIO.	FULL	991399	
☐ TABLE ACCESS ☐ OF Fiter Predicates	USUARIO	rutt	39 1333	
ROL='CLIENTE'				
	ORIECT NAME	OPTIONS	CARDINALITY	
ATION SELECT STATEMENT	OBJECT_NAME	OPTIONS	CARDINALITY COST 4397	
→ MESTED LOOPS			4397	
⊕-⊞ VIEW			1	
i → SORT		AGGREGATE	1	
□ VIEW			439744	
⊟-		GROUP BY	439744 671772	
⊕ Ot Access Predicates			0/1//2	
IDPEDIDO=PEDIDOPRODUCTO.IDPEDIDO				
TABLE ACCESS	PEDIDO	FULL	355040	
☐ Of Filter Predicates				
TO_CHAR(INTERNAL_FUNCTION(FECHAYHORA), DY)=DOM' TABLE ACCESS	DEDIDODODUCTO	FULL	1000000	
□-■ VIEW	PEDIDOPRODUCTO	rott	4397	
⊕ O Filter Predicates				
from\$_subquery\$_001.TOTAL=from\$_subquery\$_008.TOTAL				
		GROUP BY	439744	
→ M HASH JOIN			671772	
☐ On Access Predicates IDPEDIDO=PEDIDOPRODUCTO.IDPEDIDO				
☐ TABLE ACCESS	PEDIDO	FULL	355040	
□ OF Filter Predicates	Maria de la companya della companya	555	Sold to	
TO_CHAR(INTERNAL_FUNCTION(FECHAYHORA),'DY')='DOM'				
TABLE ACCESS	PEDIDOPRODUCTO.	FULL	1000000	
RATION	OBJECT_NAME	OPTIONS	CARDINALITY COST	
SELECT STATEMENT			4397	
NESTED LOOPS			4397	
G. A SOUT		ACCRECATE	1	
		AGGREGATE	1 439744	
⊟ ⊕ HASH		GROUP BY	439744	
⊞ M HASH JOIN			671772	
⊕ Ot Access Predicates				
IDPEDIDO =PEDIDOPRODUCTO.IDPEDIDO	DEDITO	B	(Armai an)	
☐ TABLE ACCESS ☐ ♥ Filter Predicates	PEDIDO	FULL	355040	
TO_CHAR(INTERNAL_FUNCTION(FECHAYHORA),'DY')='DOM'				
TABLE ACCESS	PEDIDOPRODUCTO.	FULL	1000000	
B VIEW			4397	
☐ ○ ○ ○ Pilter Predicates		ong in the	7	
☐ OF Filter Predicates from\$_subquery\$_001.TOTAL=from\$_subquery\$_008.TOTAL		GROUP BY	439744	
☐ OF Filter Predicates ☐ from\$_subquery\$_001.TOTAL=from\$_subquery\$_008.TOTAL ☐ ◆ SORT		GROOF DI		
⊕ Of Filter Predicates		SKOOP DI	671772	
☐ OF Filter Predicates ☐ from\$_subquery\$_001.TOTAL=from\$_subquery\$_008.TOTAL ☐ OF SORT		GROOF	671772	
Filter Predicates from\$_subquery\$_001.TOTAL = from\$_subquery\$_008.TOTAL from\$_subquery\$_008.TOTAL from\$_subquery\$_008.TOTAL from\$_subquery\$_008.TOTAL from\$_subquery\$_008.TOTAL from\$_subquery\$_008.TOTAL from\$_subquery\$_008.TOTAL	PEDIDO	FULL	671772 355040	
Filter Predicates	PEDIDO			
Filter Predicates from\$_subquery\$_001.TOTAL = from\$_subquery\$_008.TOTAL from\$_subquery\$_008.TOTAL from\$_subquery\$_008.TOTAL from\$_subquery\$_008.TOTAL from\$_subquery\$_008.TOTAL from\$_subquery\$_008.TOTAL from\$_subquery\$_008.TOTAL	PEDIDO. PEDIDOPRODUCTO.			

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY COST	
■ SELECT STATEMENT			4397	138
⊕- NESTED LOOPS			4397	138
□ ■ VIEW			1	69
□ 4 SORT		AGGREGATE	1	
□ VIEW			439744	69
⊟-		GROUP BY	439744	69
⊞ MASH JOIN			671772	33
⊕ Ot Access Predicates				
IDPEDIDO =PEDIDOPRODUCTO.IDPEDIDO				
TABLE ACCESS	PEDIDO	FULL	355040	10
G Filter Predicates				
TO_CHAR(INTERNAL_FUNCTION(FECHAYHORA), 'DY')='DOM'				
TABLE ACCESS	PEDIDOPRODUCTO	FULL	1000000	8
□ III VIEW			4397	69
☐ OF Filter Predicates				
from\$_subquery\$_001.TOTAL=from\$_subquery\$_008.TOTAL				
⊕ 4 SORT		GROUP BY	439744	69
			671772	33
☐ Om Access Predicates				
IDPEDIDO=PEDIDOPRODUCTO,IDPEDIDO				
☐ ■ TABLE ACCESS	PEDIDO	FULL	355040	10
☐ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	NEW STATES			
TO_CHAR(INTERNAL_FUNCTION(FECHAYHORA), 'DY')='DOM'				
TABLE ACCESS	PEDIDOPRODUCTO.	FULL	1000000	8
PERATION	OBJECT_NAME	OPTIONS	CARDINALITY COST	
SELECT STATEMENT			4397	138
NESTED LOOPS			4397	138
⊕ ■ VIEW			1	69
i → SORT		AGGREGATE	1	
i VIEW			439744	693
⊟- ● HASH		GROUP BY	439744	69
⊞ M HASH JOIN			671772	336
☐ Oth Access Predicates				
IDPEDIDO =PEDIDOPRODUCTO.IDPEDIDO				
TABLE ACCESS	PEDIDO	FULL	355040	10:
□ OV Filter Predicates				
TO_CHAR(INTERNAL_FUNCTION(FECHAYHORA),'DY')='DOM'				
TABLE ACCESS	PEDIDOPRODUCTO	FULL	1000000	8
UIEW VIEW			4397	693
☐ ON Filter Predicates				
from\$_subquery\$_001.TOTAL=from\$_subquery\$_008.TOTAL				
		GROUP BY	439744	693
			671772	336
☐ On Access Predicates				
IDPEDIDO=PEDIDOPRODUCTO, IDPEDIDO				
☐ TABLE ACCESS	PEDIDO	FULL	355040	10:
☐-OF Filter Predicates				
☐ OF Filter Predicates TO_CHAR(INTERNAL_FUNCTION(FECHAYHORA), DY)=DOM ## TABLE ACCESS				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
■ SELECT STATEMENT			773490	51776
i UNION-ALL				
i			509404	31765
□ UNION-ALL				
i → M HASH JOIN			503526	14110
	dicates			
PEDIDO.	CEDULA=CEDULA			
■ NESTED LOOP	PS		503526	14110
□ M NESTED I	LOOPS			
- STAT	TISTICS COLLECTOR			
□ 🛛	HASH JOIN		503526	7777
- i - i - i - i - i - i - i - i - i - i	℧ ℅ Access Predicates			
	PEDIDO.IDPEDIDO=PEDIDOP	RODUCTO.IDPEDIDO		
	TABLE ACCESS PEDIDO	FULL	700000	1027
e-1	HASH JOIN		503526	5064
	PEDIDOPRODUCTO.IDPR	ODUCTO=OFRECEPRODUCTO.IDPRODUCTO		
	⇒ HASH JOIN		221423	2846
		JCTO=OFRECEPRODUCTO.IDPRODUCTO		
	TABLE APRODUCTO	FULL	100426	1642
	PRODUCTO.IDCA			
	TABLE A OFRECEPRODUC	TO FULL	966246	1201
	☐ ○ ○ ○ ○ Filter Predicates			
		TO.PRECIO>=36885.855		
	TABLE ACCEPEDIDOPRODUC	TO FULL	1000000	858
☐- 0 € INDE		UNIQUE SCAN		
	Access Predicates			
	PEDIDO.CEDULA=CEDULA			
TABLE AC		BY INDEX ROWID	1	
TABLE ACCES	SS USUARIO	FULL	1000059	2157





*		🔃 🗟 sql	. Todas la	as Filas Recuperad	las: 1 en 0,0	42 segundos		
	1	530316	3	19/07/16	17	F	35	497914

≠ 🚇	📵 🅦 SQL	Todas las Filas	Recuperadas: 2 en 0,176 segundos			
		♦ NOMBRE	♦ DESCRIPCIONESPANOL			∯ TOTALMENUS
1	11	productoll	descripcion11 en españo1	description11 in English	5	1
2	14	producto14	descripcion14 en español	description14 in English	2	1

	⊕ 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

