

Consideraciones para optimización de consultas SQL



> 01 | Links Tutoriales SQL Server y Oracle

- <https://www.sqlsvertutorial.net/>



HOME START HERE BASICS ADVANCED ∨ FUNCTIONS ∨ ADMINISTRATION

- <https://www.oracletutorial.com/>



HOME START HERE BASICS PL/SQL ADVANCED ∨ FUNCTIONS ∨ API ∨

> 01 | Otros links de gran apoyo

- <https://use-the-index-luke.com>
- <https://www.sqlshack.com>
- <https://oracle-base.com>
- <https://www.dba-oracle.com/>
- <https://es.linkedin.com/pulse/32-consejos-para-oracle-sql-pueden-mejorar-el-suarez-carrillo>

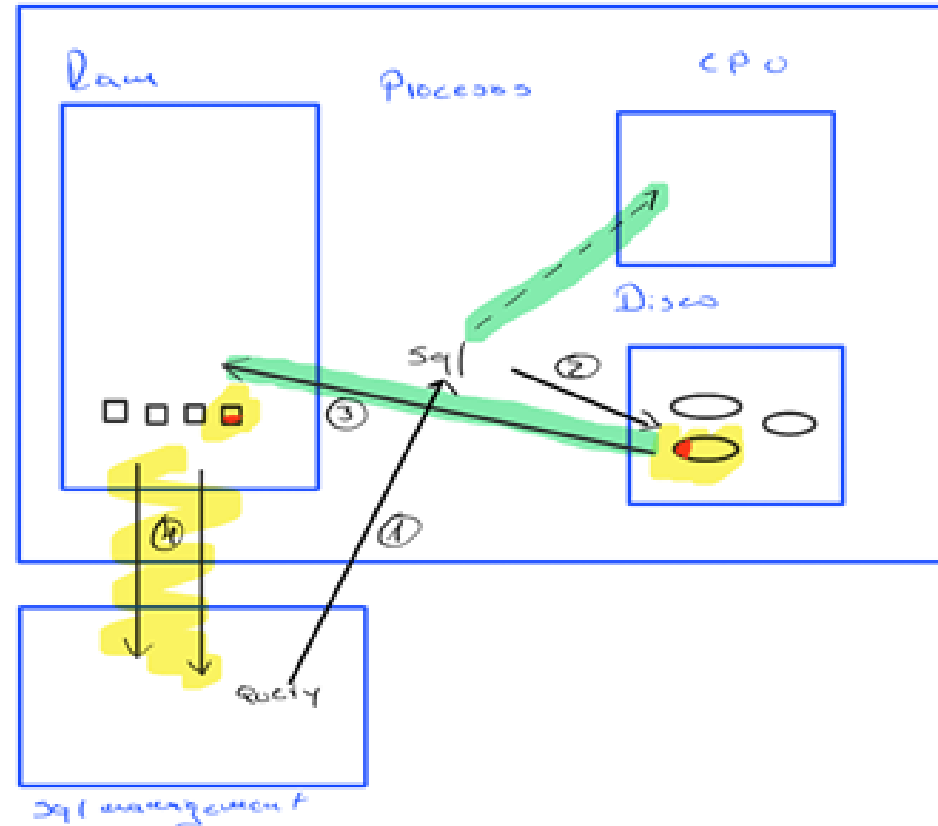
> 02 | Recomendaciones Generales

- Probar una y otra vez,
 - Varias alternativas
 - Monitoreando los resultados
- Concepto de divide y vencerás
- Enfoque en el rendimiento, no solo en lo funcional
- Desarrollar considerando que no hay muchos recursos (Evitar mejoras con fuerza bruta - HW)
- A veces resulta mejor empezar de nuevo
- Líder debe estar pendiente de la buena codificación.
 - La rotación de desarrolladores es más alta que de los líderes.

> 03

Ejecución de una consulta óptima

- La consulta solo obtiene lo justo y necesario
- Consultas basadas en índices de las tablas (tablas correctamente indexadas, particionadas)
- Consultas bien estructuradas.
- Evitar
 - Subqueries, en especial para columnas
 - Like, Order by, Distinct, Or



> 03

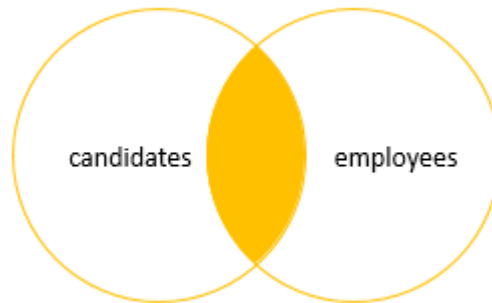
- Debe desalojar lo anterior para cargar lo nuevo (Ej. Datos que se consumen constantemente)



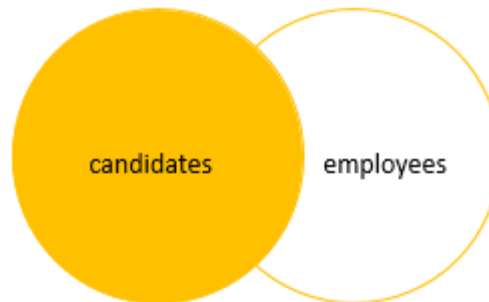
> 04 | Joins

<https://www.sqlservertutorial.net/sql-server-basics/sql-server-joins/>

<https://www.sqlservertutorial.net/sql-server-basics/sql-server-inner-join/>

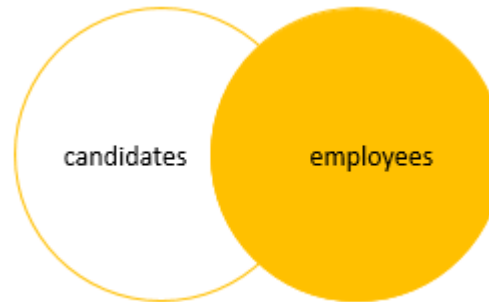


<https://www.sqlservertutorial.net/sql-server-basics/sql-server-left-join/>

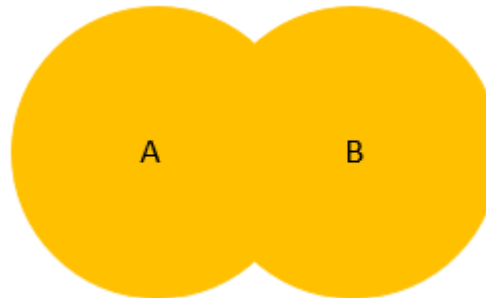


> 04 | Joins

<https://www.sqlservertutorial.net/sql-server-basics/sql-server-right-join/>



<https://www.sqlservertutorial.net/sql-server-basics/sql-server-full-outer-join/>



- En caso de mezclar Inner con otros Joins, **siempre debe ir primero el INNER JOIN**

> 05

Tips para crear consultas:

- Pueden explorar la estructura de la tabla con “EXEC sp_help tablename” (SQL SERVER)

```
EXEC sp_help 'sales.customers'
```

Results									
phone	varchar	no	20			yes	no	yes	
email	varchar	no	255			no	no	no	
street	varchar	no	255			yes	no	yes	
city	varchar	no	50			yes	no	yes	
state	varchar	no	25			yes	no	yes	

Identity	Seed	Increment	Not For Replication
customer_id	1	1	0

RowGuidCol
No rowguidcol column defined.

Data_located_on_filegroup
PRIMARY

index_name	index_description	index_keys
PK_customer_CD65CB8573CA3D02	clustered, unique, primary key located on PRIMARY	customer_id

constraint_type	constraint_name	delete_action	update_action	status
PRIMARY KEY	PK_customer_CD65CB8573CA3D02	(C)	(C)	(C)

> 05

Tips para crear consultas:

- Estructura de la tabla - Doble clic en la tabla
- (Oracle – SQL Developer)

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane shows the 'oracle_ot' connection with a tree view of tables including INVENTORIES. The main pane shows the 'INVENTORIES' table structure with columns: PRODUCT_ID, WAREHOUSE_ID, and QUANTITY. Below this, the 'Constraints' tab shows the primary key constraint 'PK_INVENTORIES' on columns 'PRODUCT_ID' and 'WAREHOUSE_ID'. A red arrow points to the 'Indexes' tab, which shows the index 'OT' on the 'PRODUCT_ID' column.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
PRODUCT_ID	NUMBER(12,0)	No	(null)	1	(null)
WAREHOUSE_ID	NUMBER(12,0)	No	(null)	2	(null)
QUANTITY	NUMBER(8,0)	No	(null)	3	(null)

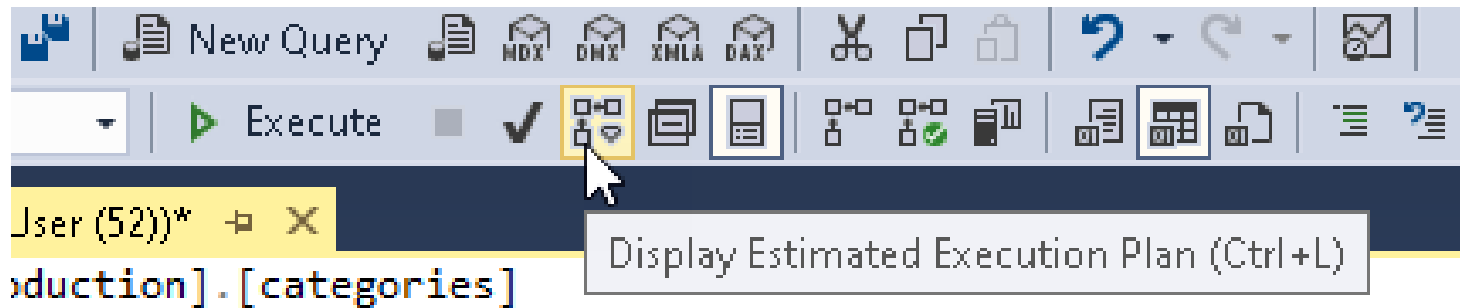
INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCTIONALITY	JOIN_INDEX	COLUMNS
OT	PK_INVENTORIES	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	PRODUCT_ID, WAREHOUSE_ID

INDEX_OWNER	INDEX_NAME	TABLE_OWNER	TABLE_NAME	COLUMN_NAME	COLUMN_POSITION	COLUMN_LENGTH	CHAR_LENGTH	DESCEND	COLLATED_COLUMN_ID
OT	PK_INVENTORIES	OT	INVENTORIES	PRODUCT_ID	1	22	0	ASC	(null)
OT	PK_INVENTORIES	OT	INVENTORIES	WAREHOUSE_ID	2	22	0	ASC	(null)

> 06

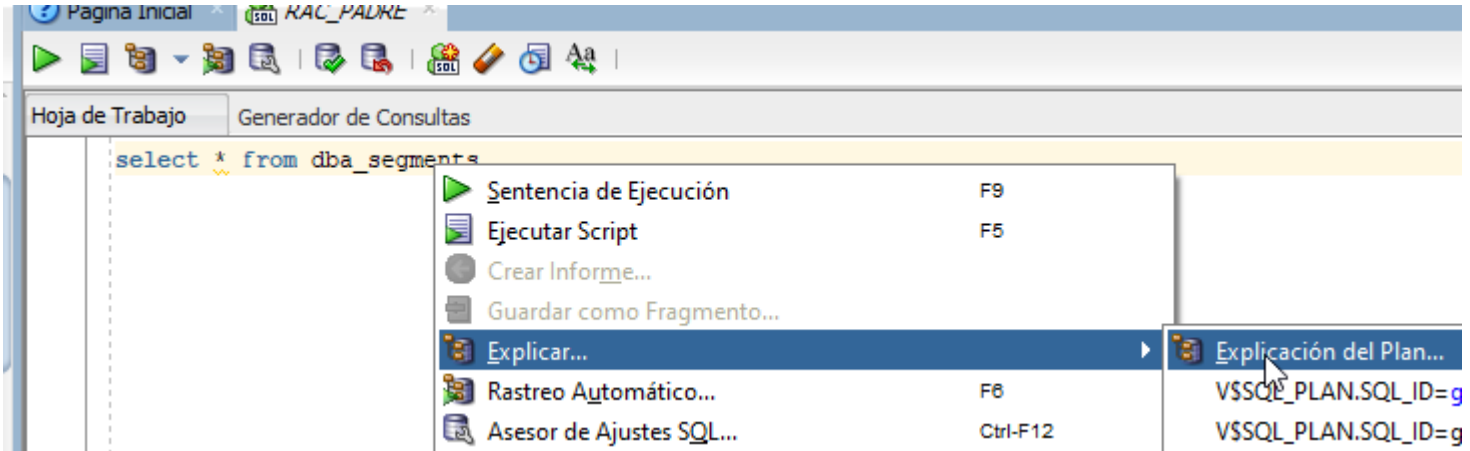
Tips para crear consultas:

- Ver plan de ejecución (SQL SERVER)



> 06 | Tips para crear consultas:

- Plan de ejecución en SQL DEVELOPER (ORACLE)



Script Output x Explain Plan x

SQL 0.065 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				
TABLE ACCESS	INVENTORIES	FULL	124	3
Filter Predicates				
WAREHOUSE_ID=4			124	3
Other XML				

> 06

Tips para crear consultas:

- Plan de ejecución ORACLE

```
EXPLAIN PLAN FOR  
SELECT * FROM INVENTORIES  
WHERE WAREHOUSE_ID=4 ;
```

```
SELECT PLAN_TABLE_OUTPUT  
FROM TABLE(DBMS_XPLAN.DISPLAY());
```

PLAN_TABLE_OUTPUT							
1	Plan hash value: 3778774871						
2							
3	-----						
4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
5	-----						
6	0	SELECT STATEMENT		124	1240	3 (0)	00:00:01
7	* 1	TABLE ACCESS FULL	INVENTORIES	124	1240	3 (0)	00:00:01
8	-----						
9							
10	Predicate Information (identified by operation id):						
11	-----						

> 07

Tips para crear consultas:

- Para revisar el consumo, habilitar las estadísticas (SQL SERVER)

```
set statistics io on  
set statistics time on
```

```
SQL Server parse and compile time:
```

```
  CPU time = 0 ms, elapsed time = 0 ms.
```

```
(1303 rows affected)
```

```
Table 'customers'. Scan count 2, logical reads 42, physical reads 0, read-ahead reads 0, lob logical
```

```
SQL Server Execution Times:
```

```
  CPU time = 0 ms,  elapsed time = 59 ms.
```

```
Completion time: 2022-07-06T10:45:11.4873178-05:00
```

> 08

Tips para crear consultas:

- No usar “ select * ”, sino especificar las columnas necesarias
 - Especialmente para los aplicativos
- Mantener siempre una indentación correcta

```
Select *  
from production.brands
```

```
Select  brand_id,  
        brand_name  
from production.brands
```

> 09

Tips para crear consultas:

- Evitar el uso de “**Distinct**”, preferiblemente usar los filtros “Where” necesarios para obtener el dato único.
- Usar el “**Order By**” solo en casos estrictamente necesarios.
 - **No** usarlo en combinación con “Distinct” para obtener un dato único.

```
Select top 1 *
from
(
    Select distinct first_name, last_name
    FROM sales.Customers
    where state='NY'
)x
order by first_name, last_name
```

first_name	last_name
Aaron	Knapp

```
Select first_name, last_name
FROM sales.Customers
where state='NY' and
      zip_code='10701' and
      street='807 Grandrose Ave. '
```

first_name	last_name
Aaron	Knapp

> 10 | Tips para crear consultas:

- No usar “Order by” por **número de columna**. Especificar el nombre de la columna o el alias para ordenar.

```
select State,
       city,
       first_name,
       last_name
from sales.customers
where phone is null
order by 1,2
```

State	city	first_name	last_name
CA	Apple Valley	Hedwig	Paul
CA	Apple Valley	Angelina	Lloyd
CA	Apple Valley	Hayden	Cross
CA	Apple Valley	Justin	Newton
CA	Apple Valley	Majorie	Glover
CA	Apple Valley	Hue	Dalton

```
select State,
       city,
       first_name,
       last_name
from sales.customers
where phone is null
order by State,
       city
```

State	city	first_name	last_name
CA	Apple Valley	Hedwig	Paul
CA	Apple Valley	Angelina	Lloyd
CA	Apple Valley	Hayden	Cross
CA	Apple Valley	Justin	Newton
CA	Apple Valley	Majorie	Glover
CA	Apple Valley	Hue	Dalton

> 11

Tips para crear consultas:

- Evitar en lo posible el uso de “Or”, “In”

```
select  State,  
        city,  
        first_name,  
        last_name  
from sales.customers  
where   state = 'CA' or  
        state = 'NY'
```

```
select  State,  
        city,  
        first_name,  
        last_name  
from sales.customers  
where state in ('CA', 'NY')
```

> 12 | Tips para crear consultas:

- Los “Joins” no hacerlos con comas, sino especificar claramente el tipo de “Join” a realizar

```
SELECT customers.state,  
       Count(*)  
FROM sales.customers,sales.orders  
WHERE Customers.Customer_ID = orders.customer_id AND -- DEL JOIN  
       customers.state='NY' --FILTRO GENERAL  
GROUP BY customers.state
```

100 %		
Results Messages		
	state	(No column name)
1	NY	1093

```
SELECT customers.state,  
       Count(*)  
FROM sales.customers  
INNER JOIN sales.orders  
  ON Customers.Customer_ID = orders.customer_id --FILTRO DEL JOIN  
WHERE customers.state='NY' --FILTRO GENERAL  
GROUP BY customers.state
```

100 %		
Results Messages		
	state	(No column name)
	NY	1093

> 13 | Tips para crear consultas:

- Usar “Where” en lugar de “Having” para definir filtros

```
|SELECT  customers.state,  
         Count(*)  
FROM    sales.customers  
INNER JOIN sales.orders  
        ON Customers.Customer_ID = orders.customer_id  
GROUP BY customers.state  
HAVING  customers.state='NY'
```

```
=|SELECT  customers.state,  
         Count(*)  
FROM    sales.customers  
INNER JOIN sales.orders  
        ON Customers.Customer_ID = orders.customer_id  
WHERE   customers.state='NY'  
GROUP BY customers.state
```

> 13

Tips para crear consultas:

- “Having” usarlo solo para filtrar campos agregados

```
SELECT customers.state,  
       Count(*)  
FROM sales.customers  
INNER JOIN sales.orders  
       ON Customers.Customer_ID = orders.customer_id  
GROUP BY customers.state  
HAVING count(*)>300
```

%	
Results	Messages
state	(No column name)
NY	1093
CA	348

<https://www.sqlservertutorial.net/sql-server-basics/sql-server-having/>

> 14 | Tips para crear consultas:

- Evitar las subconsultas, especialmente para obtener el valor de una columna. En su lugar usar “Join”

```
SELECT PP.product_id,  
  (SELECT BRAND_NAME  
   FROM [production].[brands] PB  
   WHERE PB.brand_id=PP.brand_id) BRAND,  
  (SELECT CATEGORY_NAME  
   FROM [production].[categories] PC  
   WHERE PC.category_id=PP.category_id) CATEGORY,  
  PP.product_name,  
  PP.model_year,  
  PP.list_price  
FROM [production].[products] PP
```

product_id	BRAND	CATEGORY	product_name
1	Trek	Mountain Bikes	Trek 820 - 2016
2	Ritchey	Mountain Bikes	Ritchey Timberwolf Frameset - 2016
3	Surly	Mountain Bikes	Surly Wednesday Frameset - 2016
4	Trek	Mountain Bikes	Trek Fuel EX 8 29 - 2016
5	Heller	Mountain Bikes	Heller Shagmaw Frame - 2016
6	Surly	Mountain Bikes	Surly Ice Cream Truck Frameset

```
SELECT PP.product_id,  
  BRAND_NAME BRAND,  
  CATEGORY_NAME CATEGORY,  
  PP.product_name,  
  PP.model_year,  
  PP.list_price  
FROM [production].[products] PP  
INNER JOIN [production].[brands] PB  
  ON PB.brand_id=PP.brand_id  
INNER JOIN [production].[categories] PC  
  ON PC.category_id=PP.category_id
```

product_id	BRAND	CATEGORY	product_name
1	Trek	Mountain Bikes	Trek 820 - 2016
2	Ritchey	Mountain Bikes	Ritchey Timberwolf Frameset - 2016
3	Surly	Mountain Bikes	Surly Wednesday Frameset - 2016
4	Trek	Mountain Bikes	Trek Fuel EX 8 29 - 2016
5	Heller	Mountain Bikes	Heller Shagmaw Frame - 2016
6	Surly	Mountain Bikes	Surly Ice Cream Truck Frameset
7	Trek	Mountain Bikes	Trek Slash 29 - 2016
8	Trek	Mountain Bikes	Trek Remedy 29 - 2016

> 14 | Tips para crear consultas:

- Evitar las subconsultas, especialmente para obtener el valor de una columna. En su lugar usar “Join”

```
(321 rows affected)
Table 'Workfile'. Scan count 0, logical reads 0, physi
Table 'Worktable'. Scan count 0, logical reads 0, phys
Table 'products'. Scan count 1, logical reads 5, physi
Table 'brands'. Scan count 1, logical reads 2, physica
Table 'categories'. Scan count 1, logical reads 2, phy
```

```
SQL Server Execution Times:
    CPU time = 0 ms,  elapsed time = 147 ms.
```

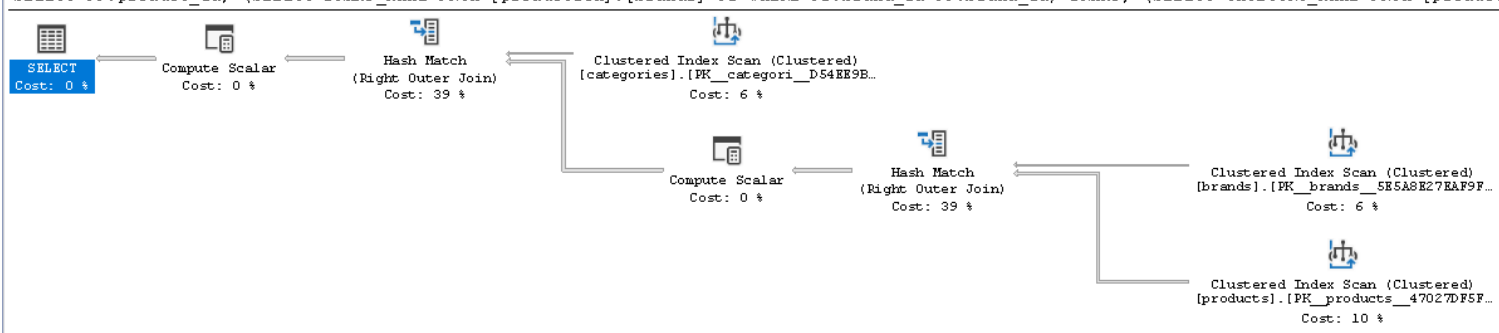
```
(321 rows affected)
Table 'Workfile'. Scan count 0, logical reads 0, physi
Table 'Worktable'. Scan count 0, logical reads 0, phys
Table 'products'. Scan count 1, logical reads 5, physi
Table 'categories'. Scan count 1, logical reads 2, phy
Table 'brands'. Scan count 1, logical reads 2, physica
```

```
SQL Server Execution Times:
    CPU time = 0 ms,  elapsed time = 103 ms.
```

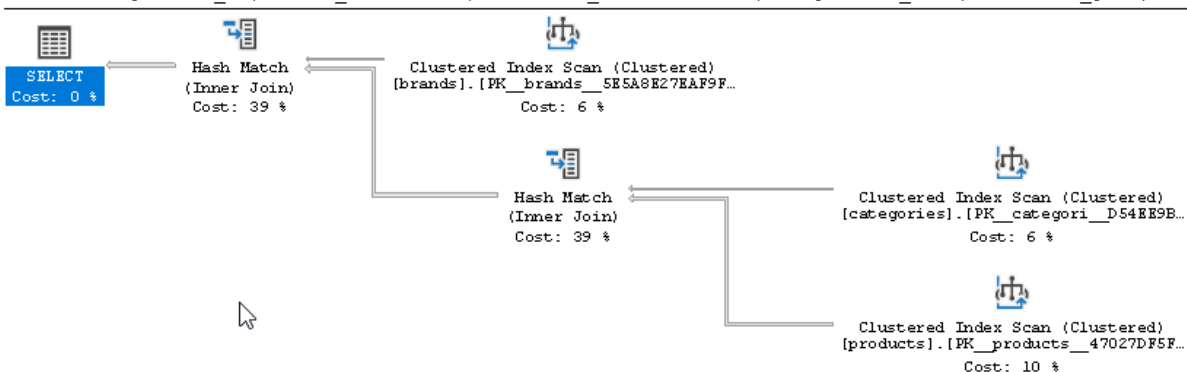
> 14 | Tips para crear consultas:

- Evitar las subconsultas, especialmente para obtener el valor de una columna. En su lugar usar “Join”

Query 1: Query cost (relative to the batch): 100%
SELECT PP.product_id, (SELECT BRAND_NAME FROM [production].[brands] PB WHERE PB.brand_id=PP.brand_id) BRAND, (SELECT CATEGORY_NAME FROM [product

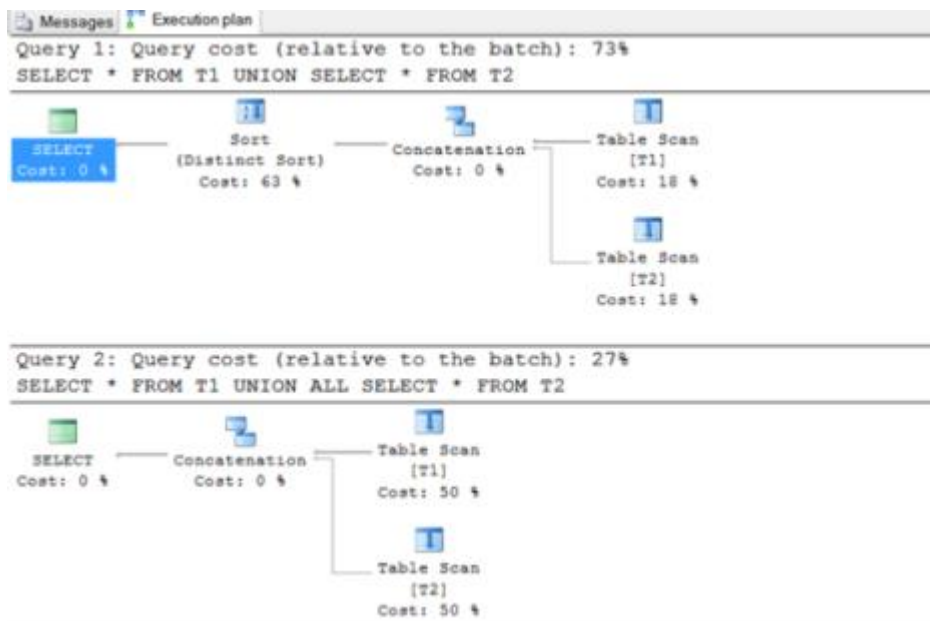


Query 1: Query cost (relative to the batch): 100%
SELECT PP.product_id, BRAND_NAME BRAND, CATEGORY_NAME CATEGORY, PP.product_name, PP.model_year, PP.lis



> 15 | UNION vs UNION ALL

- **UNION**
 - Remueve duplicados
 - Necesita ordenar el resultado para remover los duplicados
 - **Costo adicional en el rendimiento del motor de BD**
- **UNION ALL**
 - Mantiene duplicados
 - No necesita ordenar

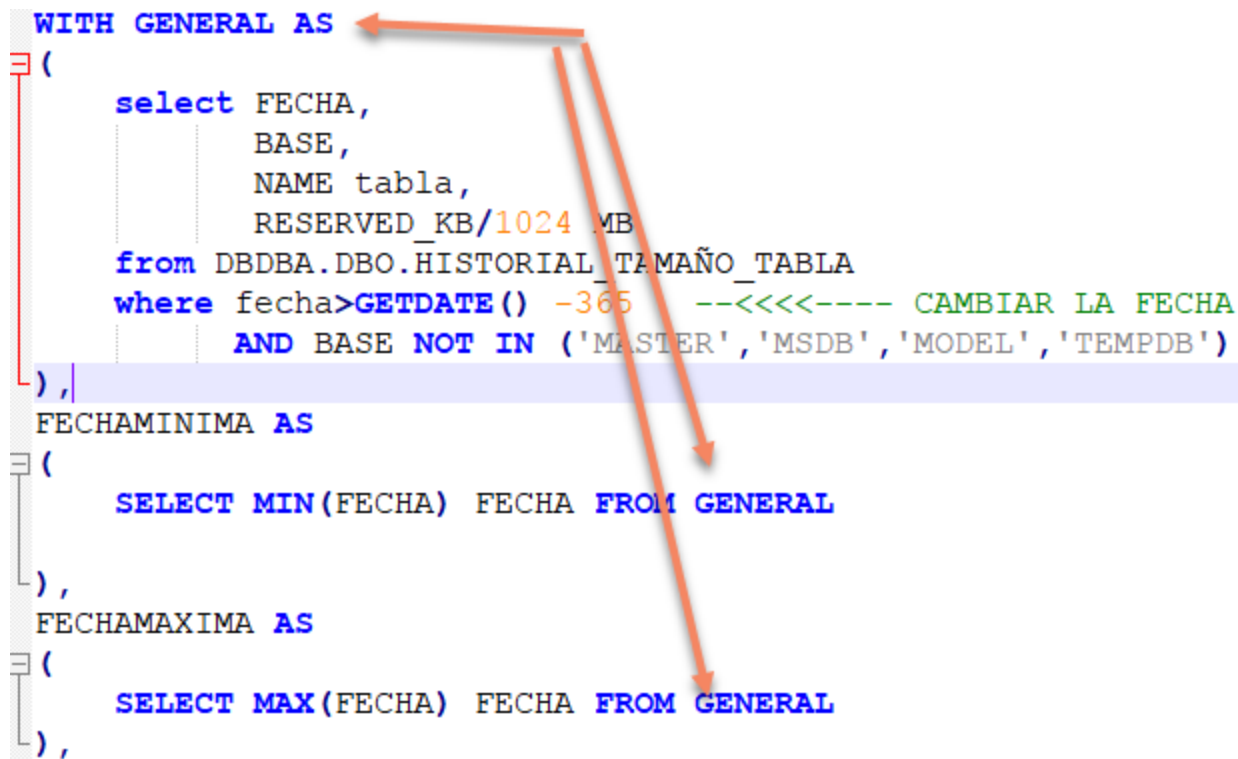


> 16

CTEs

Ejemplo: Utilizar una CTE en otra CTE

```
WITH GENERAL AS  
(  
    select FECHA,  
           BASE,  
           NAME tabla,  
           RESERVED_KB/1024 MB  
    from DBDBA.DBO.HISTORIAL_TAMAÑO_TABLA  
    where fecha>GETDATE() -365  --<<<<----- CAMBIAR LA FECHA  
           AND BASE NOT IN ('MASTER', 'MSDB', 'MODEL', 'TEMPDB')  
) ,  
FECHAMINIMA AS  
(  
    SELECT MIN(FECHA) FECHA FROM GENERAL  
) ,  
FECHAMAXIMA AS  
(  
    SELECT MAX(FECHA) FECHA FROM GENERAL  
) ,
```



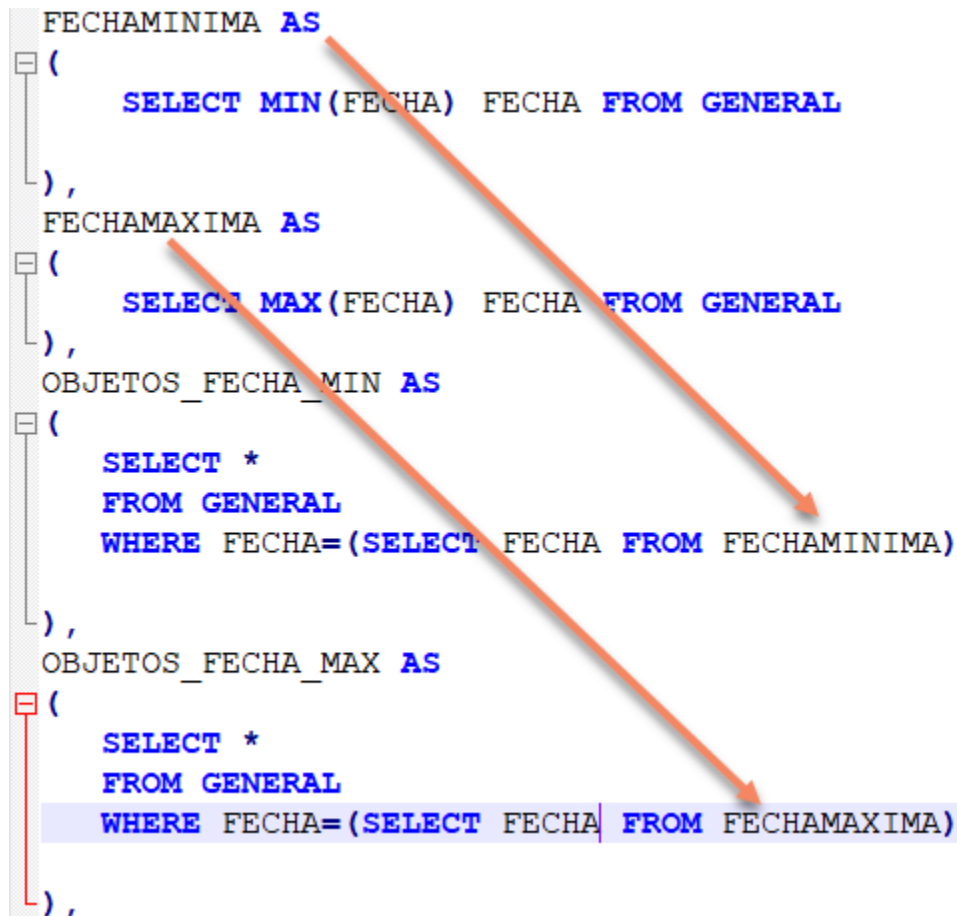
<https://www.sqlservertutorial.net/sql-server-basics/sql-server-cte/>

> 16

CTEs

Ejemplo: Utilizar una CTE en otra CTE

```
FECHAMINIMA AS
(
    SELECT MIN(FECHA) FECHA FROM GENERAL
),
FECHAMAXIMA AS
(
    SELECT MAX(FECHA) FECHA FROM GENERAL
),
OBJETOS_FECHA_MIN AS
(
    SELECT *
    FROM GENERAL
    WHERE FECHA=(SELECT FECHA FROM FECHAMINIMA)
),
OBJETOS_FECHA_MAX AS
(
    SELECT *
    FROM GENERAL
    WHERE FECHA=(SELECT FECHA FROM FECHAMAXIMA)
),
```



> 16

CTEs

Ejemplo: Utilizar una CTE en otra CTE

```
OBJETOS_FECHA_MIN AS
(
    SELECT *
    FROM GENERAL
    WHERE FECHA=(SELECT FECHA FROM FECHAMINIMA)
),
OBJETOS_FECHA_MAX AS
(
    SELECT *
    FROM GENERAL
    WHERE FECHA=(SELECT FECHA FROM FECHAMAXIMA)
),
OBJETOS_CRECIENTES AS
(
    SELECT ISNULL(MIN.BASE,MAX.BASE) BASE,
           ISNULL(MIN.TABLA,MAX.TABLA) TABLA,
           MIN.MB MB_MIN,
           MAX.MB MB_MAX,
           ISNULL(MAX.MB,0) - ISNULL(MIN.MB,0) DIFF
    FROM OBJETOS_FECHA_MIN MIN
    FULL OUTER JOIN OBJETOS_FECHA_MAX MAX
        ON MIN.BASE=MAX.BASE AND
           MIN.TABLA=MAX.TABLA
    WHERE ABS(ISNULL(MAX.MB,0) - ISNULL(MIN.MB,0))>100
) --AQUI YA SE OBTIENE EL HISTORIAL DE OBJETOS QUE HAN !
SELECT G.FECHA,G.BASE,G.TABLA,G.MB
FROM GENERAL G
INNER JOIN OBJETOS_CRECIENTES O
    ON G.BASE=O.BASE AND
       G.TABLA=O.TABLA
```



> 17 | Row number

- SQL Server ROW_NUMBER Function
- <https://www.sqlservertutorial.net/sql-server-window-functions/sql-server-row-number-function/>

first_name	last_name	city	row_num
Douglass	Blankenship	Albany	1
Mi	Gray	Albany	2
Priscilla	Wilkins	Albany	3
Andria	Rivers	Amarillo	1
Delaine	Estes	Amarillo	2
Jonell	Rivas	Amarillo	3
Luis	Tyler	Amarillo	4
Narcisa	Knapp	Amarillo	5
Abby	Gamble	Amityville	1
Barton	Cox	Amityville	2
Genny	Fields	Amityville	3
Hubert	Reilly	Amityville	4
Kylee	Dickson	Amityville	5
Marisa	Chambers	Amityville	6
Myron	Ruiz	Amityville	7
Tenisha	Lyons	Amityville	8
Thalia	Home	Amityville	9

> 18 | Precaución con tablas temporales

- Tablas temporales utilizan la TempDB (SQL SERVER) o tablespaces temporales (Oracle)
- Insertar solo los datos necesarios
 - Caso contrario se puede hacer crecer el espacio temporal desmesuradamente
 - Puede causar problemas a otras sesiones en la BD que necesiten Ordenar, Agrupar o crear nuevas tablas temporales
- Evitar colocar tipos de datos **BLOB (Binary Large Object)**
- Eliminarlas lo más pronto posible

> 19 | No usar vistas para los queries

- Una vista puede tener un propósito diferente al que se está buscando
- De por si la vista puede estar mal optimizada

Recomendable

- Extraer lo necesario de la vista
- Optimizar lo extraído.
- Si se requiere usar la vista, probar si el rendimiento con la vista es igual que al extraer el código y ponerla en su query

> 20 | Actualización de Estadísticas

- Proveen información para el motor de optimización
 - Se estima el número de filas involucradas en una consulta
 - Para preparar un plan de ejecución efectivo
- Deben actualizarse frecuentemente
 - Caso contrario se puede generar un plan de ejecución incorrecto y costoso
 - Ej: **Usar un Table Full Scan en lugar de un Index Seek**
 - Consumo de Memoria, CPU e IO
 - Incluso podrían presentarse bloqueos y deadlocks

> 20 | Actualización de Estadísticas

Campo 1	Campo 2	...
1	a	
2	b	
3	c	
4	d	

Table Full Scan



Campo 1	Campo 2	...
1	a	...
2	b	...
3	c	...
4	d	...
...
...
...
1000000	xxxx	...
1000001	yyyy	...
1000002	zzzz	...

Table  Full Scan

- Obtener estadísticas mediante (SQL SERVER):
 - **UPDATE STATISTICS NombreTabla [WITH FULLSCAN]**
- Ejemplo de impacto de las estadísticas en el plan de ejecución:
 - <https://www.brentozar.com/archive/2020/11/how-bad-statistics-cause-bad-sql-server-query-performance/>

> 20 | Actualización de Estadísticas

- Obtener estadísticas mediante (Oracle):
 - `dbms_stats.gather_schema_stats(ownname=>'OT')`

```
select num_rows from dba_tables where owner='OT'
```

	NUM_ROWS
1	(null)
2	(null)
3	(null)
4	(null)
5	(null)
6	(null)
7	(null)
8	(null)
9	(null)
10	(null)
11	(null)
12	(null)

```
select num_rows from dba_tables where owner='OT'
```

	NUM_ROWS
1	4
2	25
3	23
4	9
5	107
6	5
7	288
8	319
9	319
10	105
11	665
12	1112

> 21 | Conversiones Implícitas

- Evitar las conversiones implícitas ya que afectan al rendimiento

```
declare @año varchar(4)='2017'
```

```
select brand_id,  
       category_id,  
       product_name  
from production.products  
where model_year=@año
```

Results Messages Execution plan

Query 1: Query cost (relative to the select brand_id, category_id, product

Clustered Index Scan (Clustered Index [products].[PK_products_470])

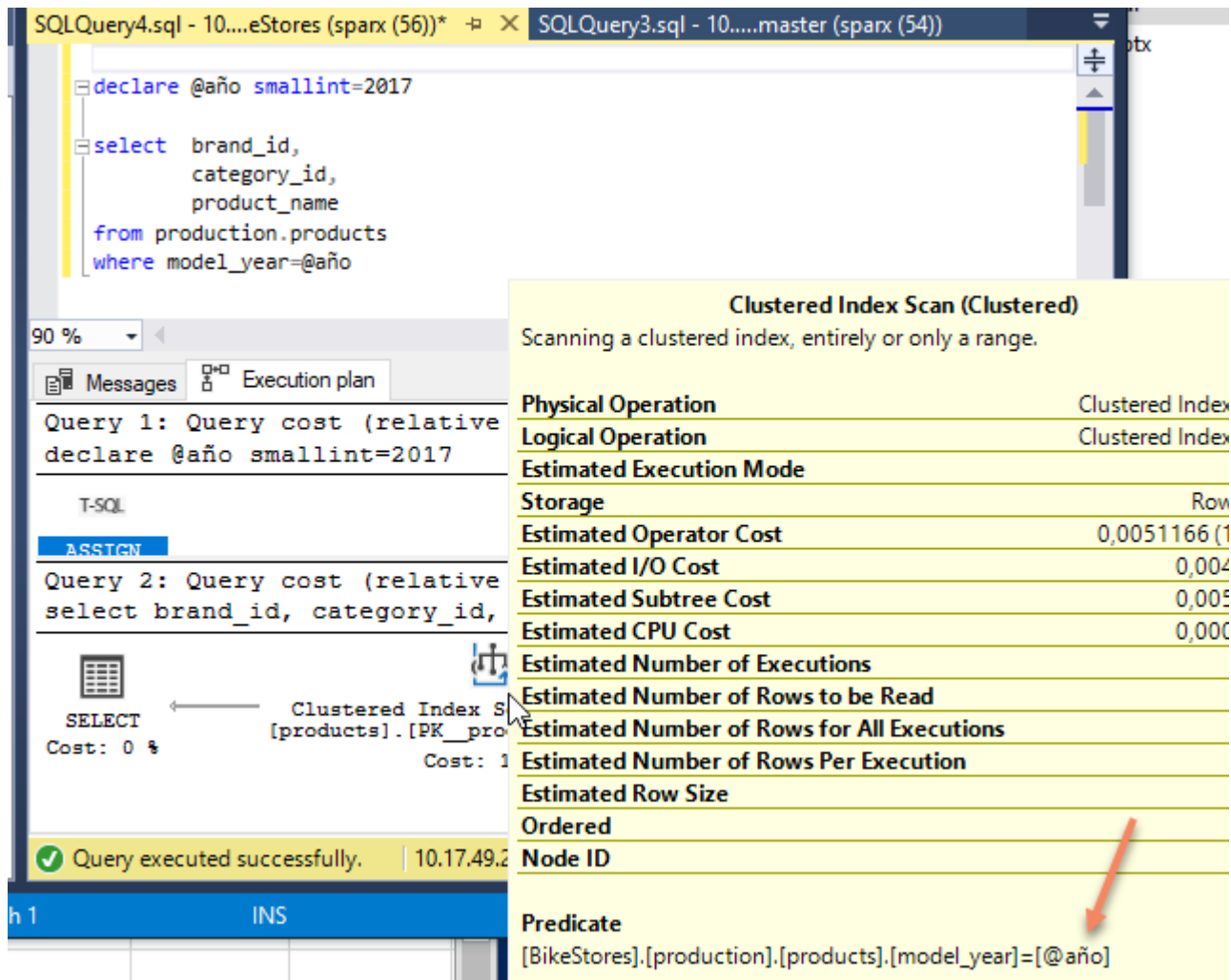
Cost: 100 %
0.000s
85 of
80 (106%)

Physical Operation Clustered Index
Logical Operation Clustered Index
Actual Execution Mode
Estimated Execution Mode
Storage Row
Number of Rows Read
Actual Number of Rows for All Executions
Actual Number of Batches
Estimated I/O Cost 0.004
Estimated Operator Cost 0.0051166 (1
Estimated Subtree Cost 0.005
Estimated CPU Cost 0.000
Estimated Number of Executions
Number of Executions
Estimated Number of Rows for All Executions
Estimated Number of Rows Per Execution
Estimated Number of Rows to be Read
Estimated Row Size
Actual Rebinds
Actual Rewinds
Ordered
Node ID

Predicate
[BikeStores].[production].[products].[model_year]=CONVERT_IMPLICIT
(smallint,[@año],0)

> 21 Conversiones Implícitas

- Mantener el mismo tipo de dato para las comparaciones



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window with the following T-SQL code:

```
declare @año smallint=2017
select brand_id,
       category_id,
       product_name
from production.products
where model_year=@año
```

The bottom pane shows the execution plan for the query. The plan consists of a single operation: "Clustered Index Scan (Clustered)". A tooltip is displayed over this operation, providing the following details:

- Clustered Index Scan (Clustered)**: Scanning a clustered index, entirely or only a range.
- Physical Operation**: Clustered Index
- Logical Operation**: Clustered Index
- Estimated Execution Mode**
- Storage**: Row
- Estimated Operator Cost**: 0,0051166 (1)
- Estimated I/O Cost**: 0,004
- Estimated Subtree Cost**: 0,005
- Estimated CPU Cost**: 0,000
- Estimated Number of Executions**
- Estimated Number of Rows to be Read**
- Estimated Number of Rows for All Executions**
- Estimated Number of Rows Per Execution**
- Estimated Row Size**
- Ordered**
- Node ID**
- Predicate**: [BikeStores].[production].[products].[model_year]=[@año]

The status bar at the bottom indicates "Query executed successfully." and "10.17.49.2".

> 22 | Índices

Recomendaciones

- Crear las consultas basadas en los índices
- Mantener el orden especificado en el índice
- Para conocer que índices tienen las tablas y su composición, utilizar sp_helpindex (SQL SERVER)

```
sp_helpindex "[sales].[order_items]"
```

index_name	index_description	index_keys
PK__order_it__837942D4546F51ED	clustered, unique, primary key located on PRIMARY	order_id, item_id

> 22 | Índices

Recomendaciones

- Muchos índices aceleran las consultas, pero ralentizan otras operaciones (INSERT, UPDATE)
- Script para observar el uso de los índices (Index usage)

```
select object_name(i.[object_id]) as TABLA,
       i.[name] as INDICE,
       i.type_desc TIPO,
       isnull(user_seeks,0)user_seeks,
       isnull(user_scans,0)user_scans,
       isnull(user_lookups,0)user_lookups,
       isnull(user_updates,0)user_updates
from sys.tables t with(nolock)
inner join sys.indexes i with(nolock)
  on object_name(i.[object_id])=t.name
  and t.type='u'
left join sys.dm_db_index_usage_stats as s with(nolock)
  on i.[object_id] = s.[object_id]
  and i.index_id = s.index_id
where i.[name] is not null
order by TABLA
```

results

Messages

Consultas

Modificaciones

TABLA	INDICE	TIPO	user_seeks	user_scans	user_lookups	user_updates
brands	PK__brands__5E5A8E27EAF9F5B3	CLUSTERED	0	0	0	0
categories	PK__categori__D54EE9B4752FAA35	CLUSTERED	12	47	82	0
customers	PK__customer__CD65CB8573CA3D02	CLUSTERED	0	0	0	0
order_items	PK__order_it__837942D4546F51ED	CLUSTERED	0	0	0	0
orders	PK__orders__46596229836D15A6	CLUSTERED	0	0	0	0
products	PK__products__47027DF5F8277386	CLUSTERED	0	0	0	0
staffs	PK__staffs__1963DD9CF8ED3C04	CLUSTERED	0	0	0	0
staffs	UQ__staffs__AB6E6164649207B1	NONCLUSTERED	0	0	0	0
stocks	PK__stocks__E68284D38CBFA1E6	CLUSTERED	0	0	0	0

> 22 | Índices Recomendaciones

- Recomendación de índices en el plan de ejecución

```
select p.product_name,  
       oi.quantity,  
       oi.list_price  
from sales.order_items_2 oi  
inner join dbo.products_nuevo p  
on oi.product_id=p.product_id  
where quantity=2
```

121 %

Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

select p.product_name, oi.quantity, oi.list_price from sales.order_items_2 oi inner join dbo.products_nuevo p

Missing Index (Impact 50.9568): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [sale:

```
graph LR
    SELECT[SELECT  
Cost: 0 %] --> HashMatch[Hash Match  
(Inner Join)  
Cost: 38 %]
    TableScanP[Table Scan  
[products_nuevo] [p]  
Cost: 0 %] --> HashMatch
    HashMatch --> TableScanOI[Table Scan  
[order_items_2] [oi]  
Cost: 62 %]
```

CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>]
ON [sales].[order_items_2] ([quantity])
INCLUDE ([product_id],[list_price])

> 22 | Índices

Recomendaciones

- Observar el uso de los índices en el plan de ejecución
- Table Scan: lee toda la tabla para obtener el dato



- Index Scan: lee todo el índice para encontrar el dato



- Index Seek: Recorre el índice (en forma de árbol) para encontrar el dato, sin leerlo completamente



> 22 | Índices Recomendaciones

- No poner funciones en los campos porque no se utiliza el índice apropiadamente
- En ocasiones ni siquiera utiliza el índice, sino realiza un TableScan

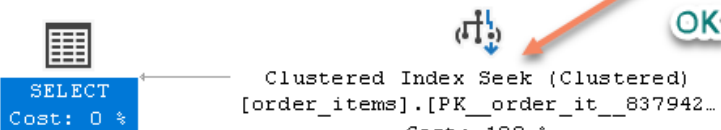
```
select product_id,
list_price
from [sales].[order_items]
where order_id=658 and
item_id=3
```

21 %

Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

select product_id, list_price from [sales].[order_items] wh



The execution plan shows a 'SELECT' operation with a cost of 0% and a 'Clustered Index Seek (Clustered)' operation on '[order_items].[PK_order_it__837942...]' with a cost of 100%. An orange arrow points from the 'item_id=3' condition in the SQL query to the index seek operation, and a green 'OK' bubble is next to it.

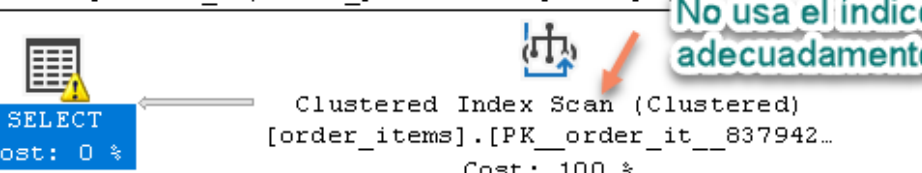
```
select product_id,
list_price
from [sales].[order_items]
where convert(varchar(10),order_id)='658' and
convert(varchar(10),item_id)='3'
```

%

Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

select product_id, list_price from [sales].[order_items] wh



The execution plan shows a 'SELECT' operation with a cost of 0% and a 'Clustered Index Scan (Clustered)' operation on '[order_items].[PK_order_it__837942...]' with a cost of 100%. An orange arrow points from the 'convert(varchar(10),item_id)='3'' condition in the SQL query to the index scan operation. A green bubble with the text 'No usa el índice adecuadamente' (Does not use the index appropriately) is next to it.

> 22 | Índices fragmentados sql server

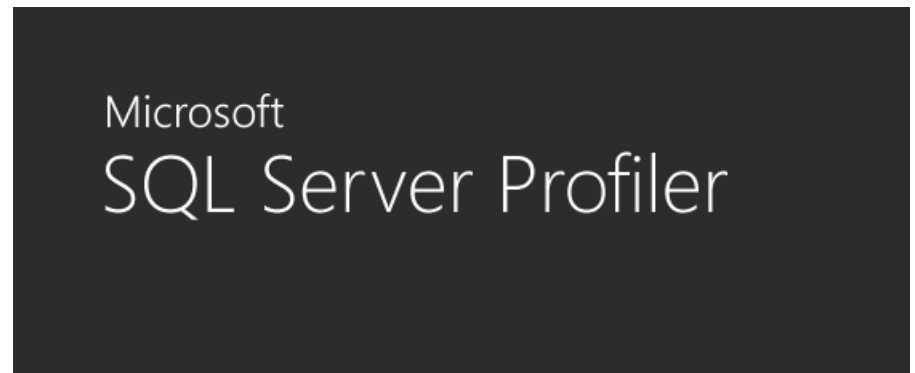
- <https://www.sqlshack.com/how-to-identify-and-resolve-sql-server-index-fragmentation/>
- Tarea administrativa por parte de los DBAs

Results		Messages			
	Schema	Table	Index	avg_fragmentation_in_percent	page_count
1	Sales	OrderTracking	IX_OrderTracking_CarrierTrackingNumber	99.1047919957873	1899
2	Sales	Store	PXML_Store_Demographics	98.4375	64
3	Sales	OrderTracking	IX_OrderTracking_SalesOrderID	98.394495412844	436
4	HumanResources	Employee	PK_Employee_BusinessEntityID	98.3870967741936	62
5	dbo	SalesOrderHeader	PK_SalesOrderHeader_SalesOrderID	94.9206349206349	630
6	Production	ProductModel	PXML_ProductModel_Instructions	91.6666666666667	12
7	Production	ProductModel	PXML_ProductModel_CatalogDescription	80	5
8	Sales	Store	AK_Store_rowguid	66.6666666666667	3
9	Production	ProductCostHistory	PK_ProductCostHistory_ProductID_StartDate	66.6666666666667	3
10	Production	ProductDescription	AK_ProductDescription_rowguid	66.6666666666667	3
11	Production	ProductListPriceHistory	PK_ProductListPriceHistory_ProductID_StartDate	66.6666666666667	3
12	Sales	SpecialOfferProduct	PK_SpecialOfferProduct_SpecialOfferID_ProductID	66.6666666666667	3

> 23

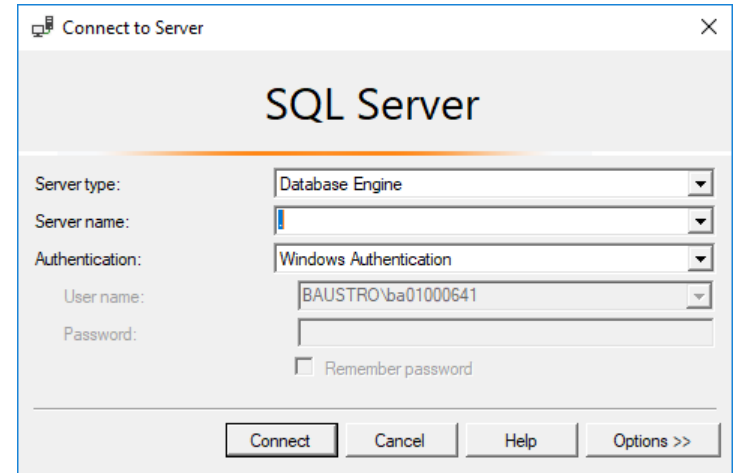
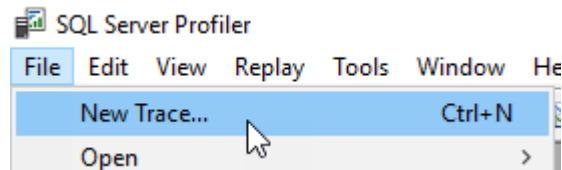
MONITOREO DE SENTENCIAS - SQL SERVER Sql Profiler

- Posibilidad de rastrear las sentencias SQL y su consumo de recursos
- Se puede observar:
 - Consumo de CPU
 - Consumo de IO
 - Bloqueos
 - Deadlocks
 - Tiempo de respuesta
 - La consulta como tal
- Se puede almacenar el resultado:
 - En una tabla BD
 - En un archivo tipo trace
- Ejecutar **al menos unas 10 veces** el sql para capturar las estadísticas correctas.

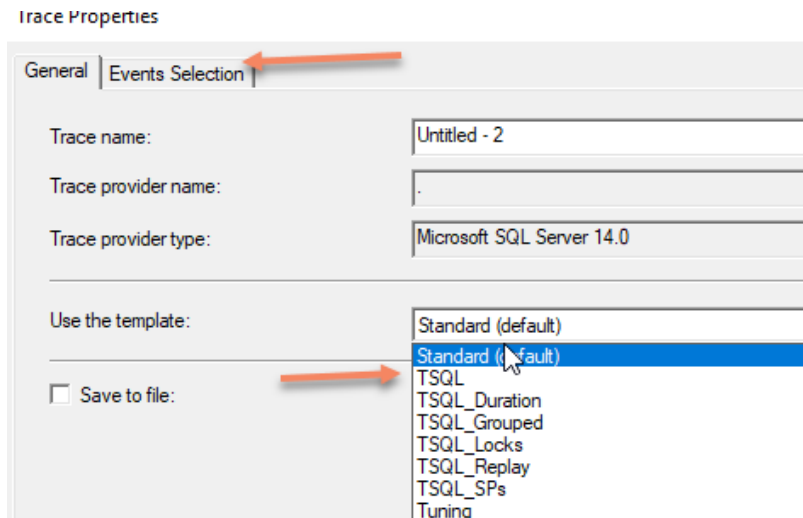


> 23 | Sql Profiler

- Crear un nuevo trace



- Se puede utilizar una plantilla



> 23 | Sql Profiler

- Eventos y columnas en base a una plantilla

Trace Properties

General Events Selection

Review selected events and event columns to trace. To see a complete list, select the "Show all events" and "Show all columns" options.

Events	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProcess
Security Audit									
<input checked="" type="checkbox"/> Audit Login	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Audit Logout		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sessions									
<input checked="" type="checkbox"/> ExistingConnection	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input checked="" type="checkbox"/>
Stored Procedures									
<input checked="" type="checkbox"/> RPC:Completed	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TSQL									
<input checked="" type="checkbox"/> SQL:BatchCompleted	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SQL:BatchStarting	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>

< >

Security Audit
Includes event classes that are used to audit server activity.

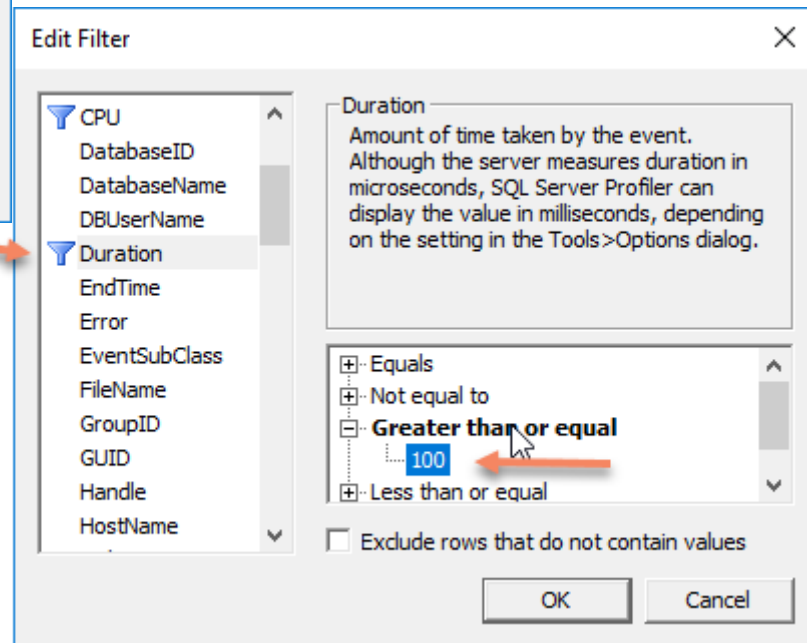
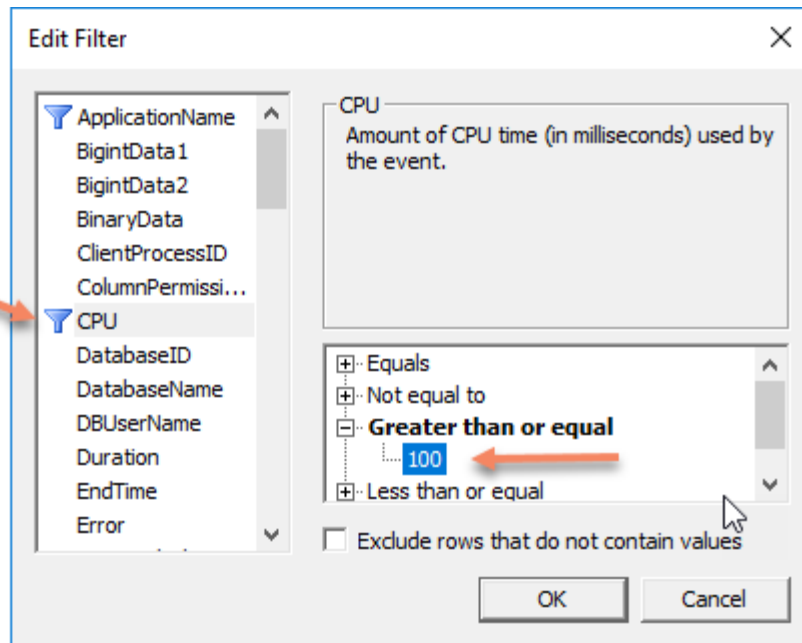
No data column selected.

☐ Show all events
☐ Show all columns

Column Filters...
Organize Columns...

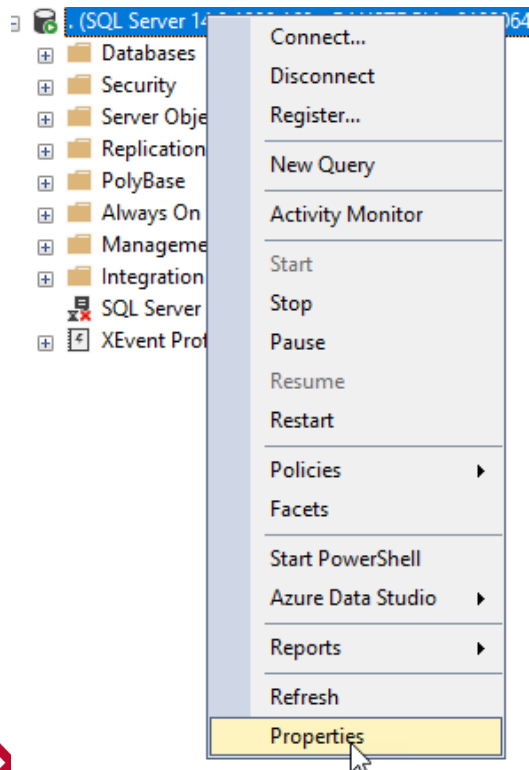
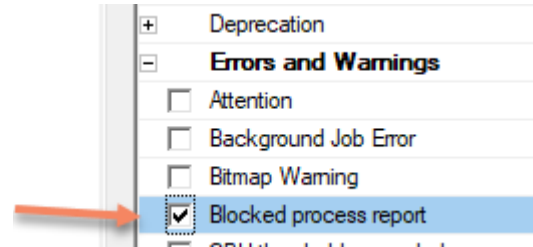
Run Cancelar Ayuda

> 23 | Sql Profiler Filtros



> 23 | Sql Profiler Bloqueos

- Detectar bloqueos
- Cambiar el umbral de tiempo de bloqueos en la instancia de BD (en segundos)



FILESTREAM	
FILESTREAM Access Level	Disabled
FILESTREAM Share Name	MSSQLSERVER
Miscellaneous	
Allow Triggers to Fire Others	True
Blocked Process Threshold	0
Boost SQL Server Priority	False
Cursor Threshold	-1
Default Full-Text Language	1033
Default Language	English

> 23 | Sql Profiler Ejemplo Bloqueos

- Bloqueos

EventClass	TextData	Duration	App
Blocked process report	<blocked-process-report monitorLoop=...	2892	
Blocked process report	<blocked-process-report monitorLoop=...	2962	
Blocked process report	<blocked-process-report monitorLoop=...	3275	
Blocked process report	<blocked-process-report monitorLoop=...	3061	
Blocked process report	<blocked-process-report monitorLoop=...	4168	
Blocked process report	<blocked-process-report monitorLoop=...	3906	
Blocked process report	<blocked-process-report monitorLoop=...	3976	
Blocked process report	<blocked-process-report monitorLoop=...	4290	
Blocked process report	<blocked-process-report monitorLoop=...	4080	
Blocked process report	<blocked-process-report monitorLoop=...	5200	
Blocked process report	<blocked-process-report monitorLoop=...	5308	
Blocked process report	<blocked-process-report monitorLoop=...	5095	
Blocked process report	<blocked-process-report monitorLoop=...	1225	
Blocked process report	<blocked-process-report monitorLoop=...	4957	
Blocked process report	<blocked-process-report monitorLoop=...	5027	



Sql Profiler

Ejemplo Bloqueos

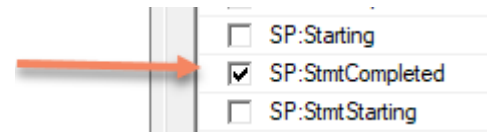
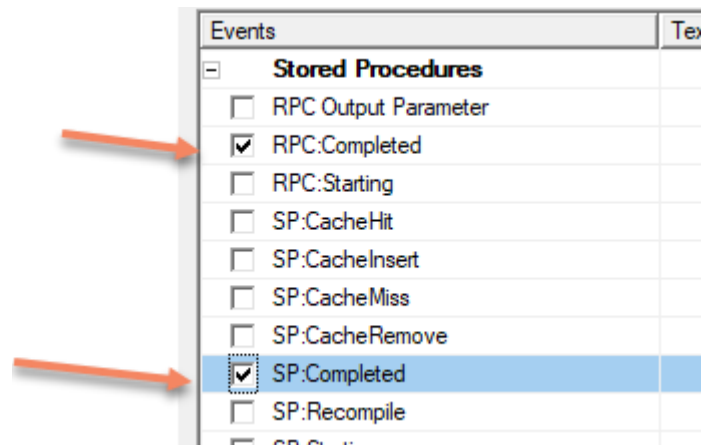
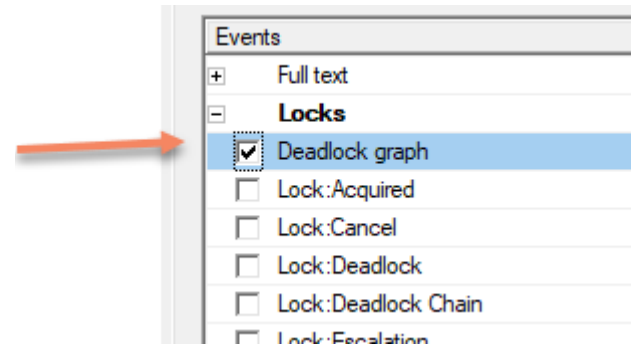
- Bloqueos

[illegible]

> 23

Sql Profiler Eventos

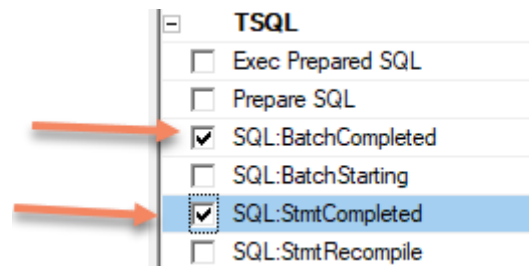
- Detectar interbloqueos
- Consumo de SPs
- Consumo de sentencias dentro del SP



> 23

Sql Profiler Eventos

- Consumo de sentencias en lote o individualmente



- Columnas típicas para revisar consumo

TextData	ApplicationName	LoginName	CPU	Reads	Writes	Duration
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

SPID	StartTime	EndTime	DatabaseName	HostName	Error
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

> 23

Sql Profiler Ejemplos

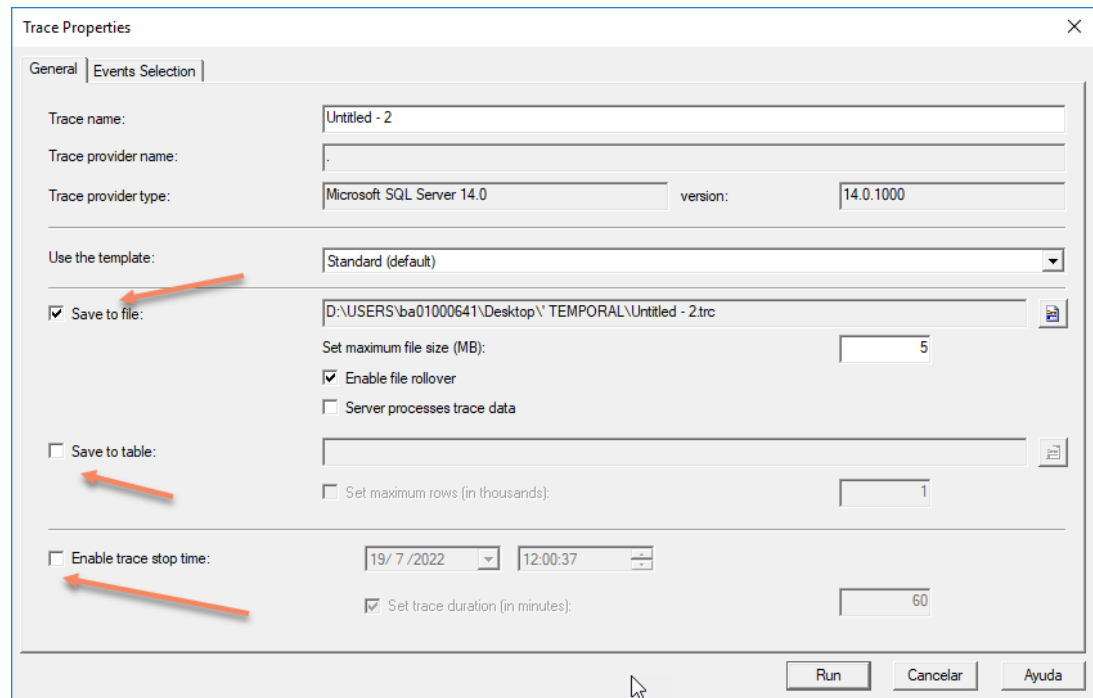
- Sentencias con alto consumo

SPID	Duration	StartTime	EndTime	Reads	Writes	CPU	IntegerData	Error
1374	3630	2020-07-09 16:20:04...	2020-07-09 16:20:08...	960224	0	3689		0 - 0

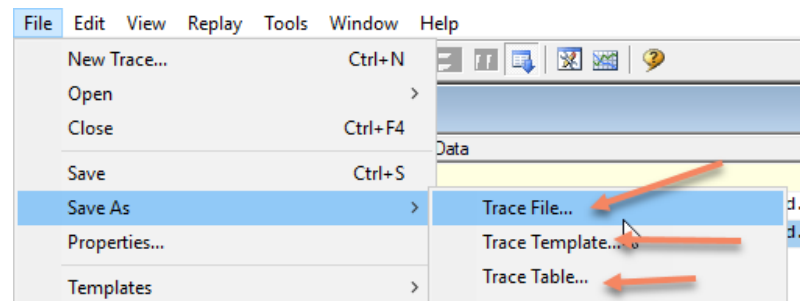
```
SELECT DISTINCT SO.ID_SOBRE FROM Contratos CO WITH (NOLOCK) INNER JOIN Sobres SO WITH (NOLOCK) ON CO.id_contrato = SO.ID_ID_Servicio = SRV.Id_Servicio LEFT OUTER JOIN Contratos_Ordenes_ModoProcesamiento COP WITH (NOLOCK) ON SO.Id_Contrato = CO.Id_Contrato FROM CONTRATOS WITH (NOLOCK) WHERE Proceso_Otros_Bancos = 'BC_SPI' AND Estado_Sobre = 'APROBADO' AND Fecha_PATEADD(day, 0, GETDATE()), 112), 112) AND Fecha_Vencimiento_Proceso >= CONVERT(datetime, CONVERT(varchar(10), DATEADD(day, 1, (select count (Id_Item) from Items where id_sobre = so.Id_Sobre and Items.Estado_Proceso = 'PENDIENTE'), and FormaPago and (select count (Id_Item) from Items where id_sobre = so.Id_Sobre and Items.Referencia_SWIFT = '' and Items.Banco = '01') and (select Id_Sobre from movimientos where CondicionProceso = 0 AND Mensaje = 'PROCESO OK' AND Id_Item = 0) or (SRV.Cod) by id_sobre desc
```

> 23 | Sql Profiler Guardar resultado

- Al configurar el trace
- Precaución porque puede ralentizar la operación de la BD



- Al finalizar la captura



> 23 | Sql Profiler

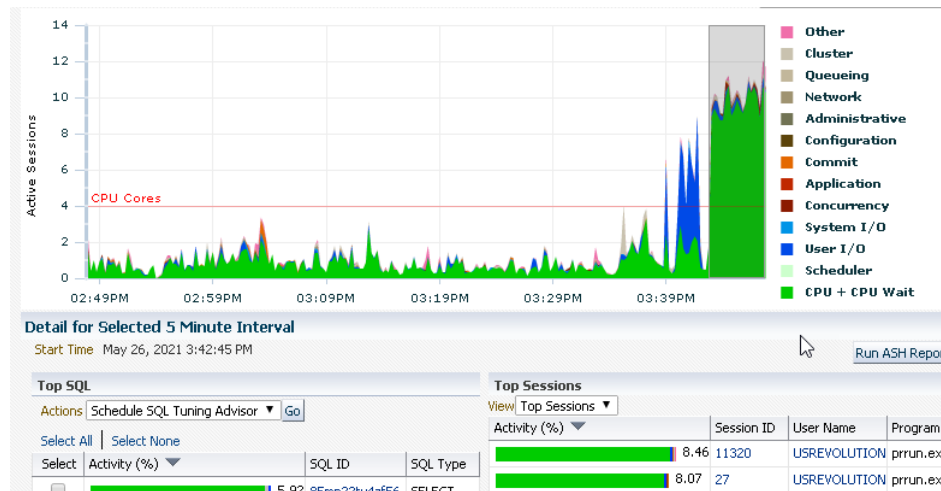
- La ejecución hacerla desde el **aplicativo**, y realizarla al menos unas **10 veces** cambiando los **parámetros** en cada ejecución.
- Escoger los eventos:
 - En la sección **Stored Procedures**:
 - **RPC:Completed**
 - **SP:STMTCompleted**
 - En la sección **TSQL**:
 - **SQL:BatchCompleted**
- Seleccionar **todas** las columnas
- **Filtrar por**
 - **Duración >=10 ms**
 - **CPU >= 10ms**
- Añadir cualquier filtro adicional que consideren necesario
- Considerar la **frecuencia de ejecución** de las sentencias, (por ejemplo: una vez cada semana, 1000 veces en un segundo, etc), y el horario estimado de ejecución.

> 24

MONITOREO DE SENTENCIAS – ORACLE ENTERPRISE MANAGER

> Propósito

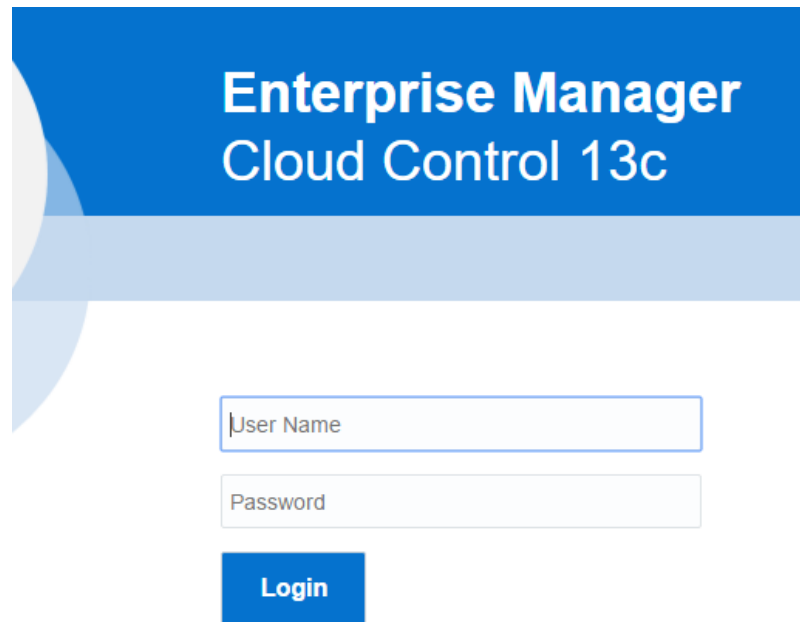
- Monitoreo de rendimiento
- Revisión de estructura de sentencias
- Revisión de planes de ejecución
- Tomar acción antes de producción



> 24 | Como Ingresar

Pantalla de login

- <https://10.16.8.46:7799/em>
 - "usuario: MONITORQA contraseña: MONITORQA01"
- <https://10.1.99.46:7799/em>
 - "usuario: MONITORDESA contraseña: MONITORDESA01"



Enterprise Manager
Cloud Control 13c

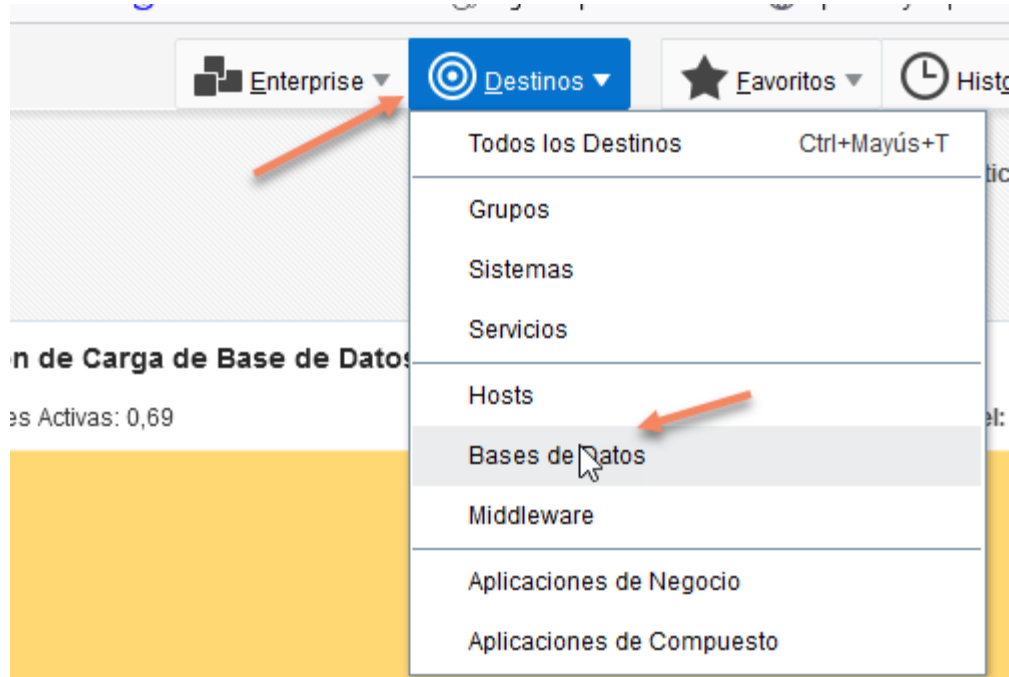
User Name

Password

Login

> 24 | Como Ingresar

Ir a destinos, luego a bases de datos

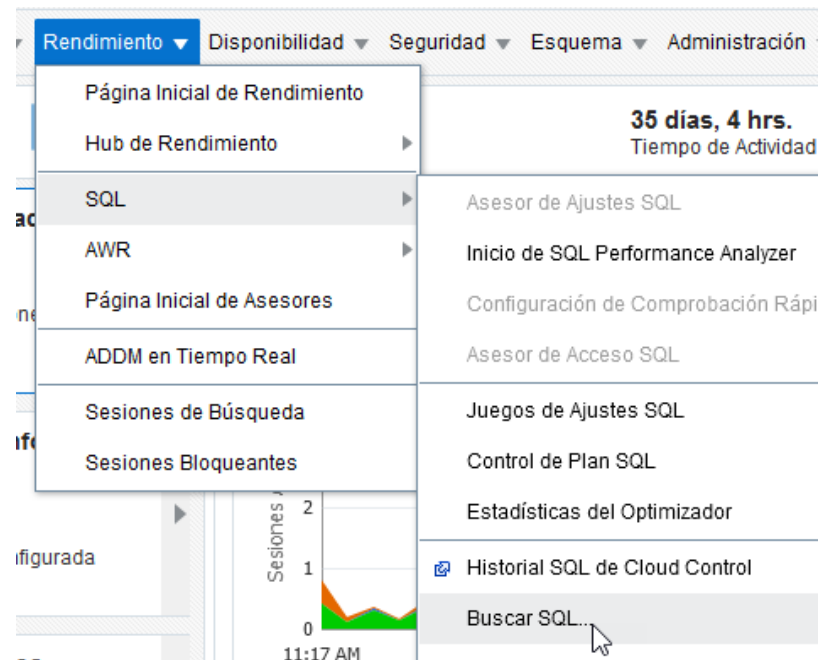
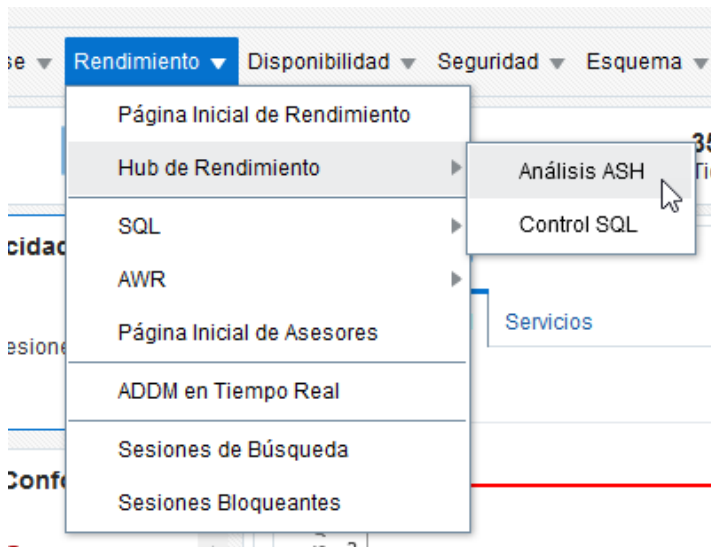


> 24 | Como Ingresar

Ingresar a la BD de desarrollo o QA



En rendimiento están las opciones para validar el consumo de los queries



> 24 | Como Ingresar

Ingresar credenciales para la conexión a la BD

Conexión a Base de Datos

Credencial ☒ Nombre ☐ Nuevo

Nombre de la Credencial NC_MONITORQA ▾

Detalles de Credencial

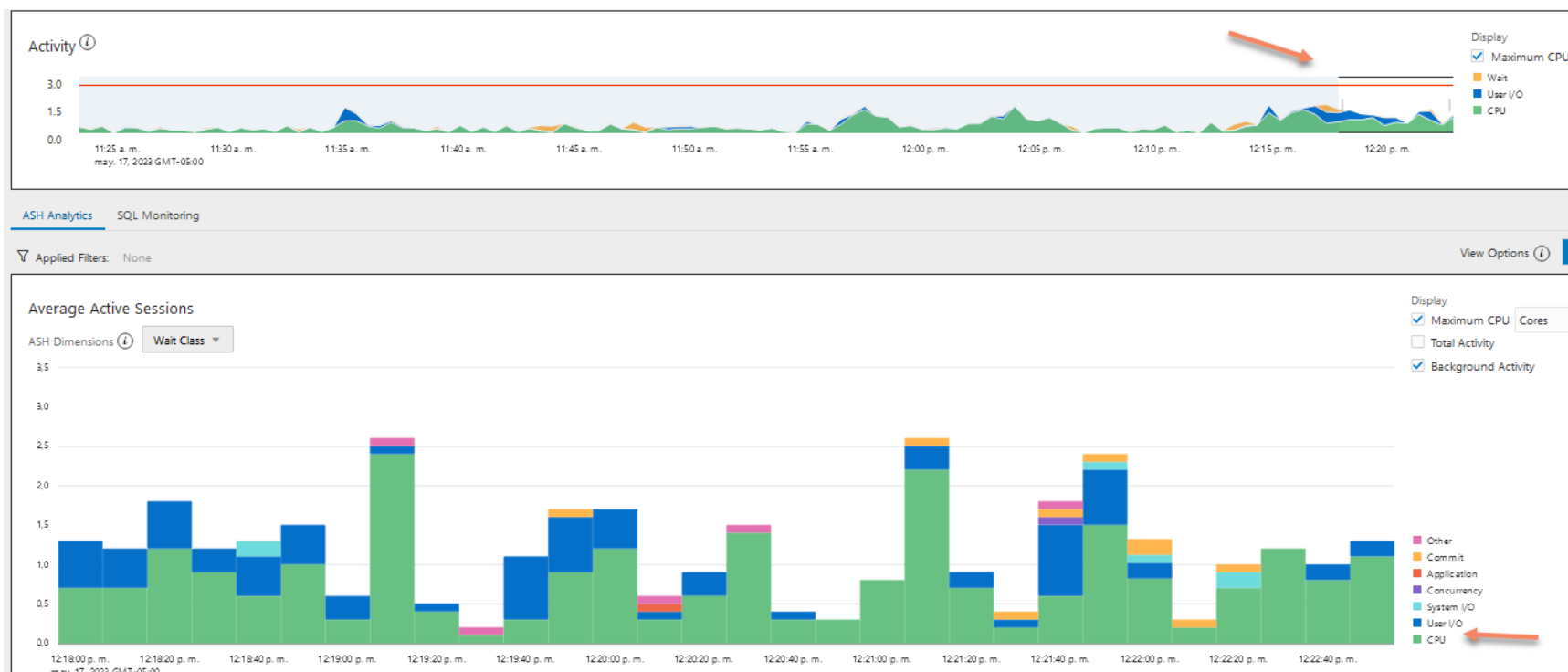
Atributo	Valor
Usuario	MONITORQA
Contraseña	*****
Rol	normal

[Más Detalles](#)

> 24 | Que revisar

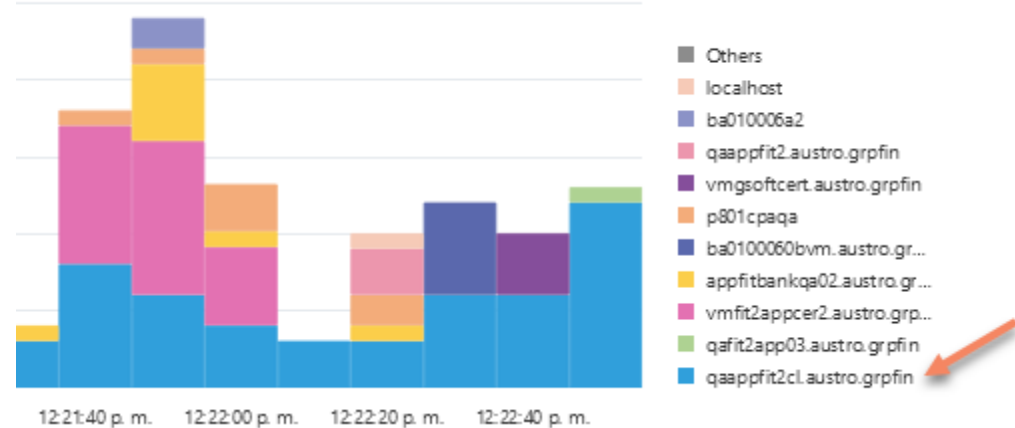
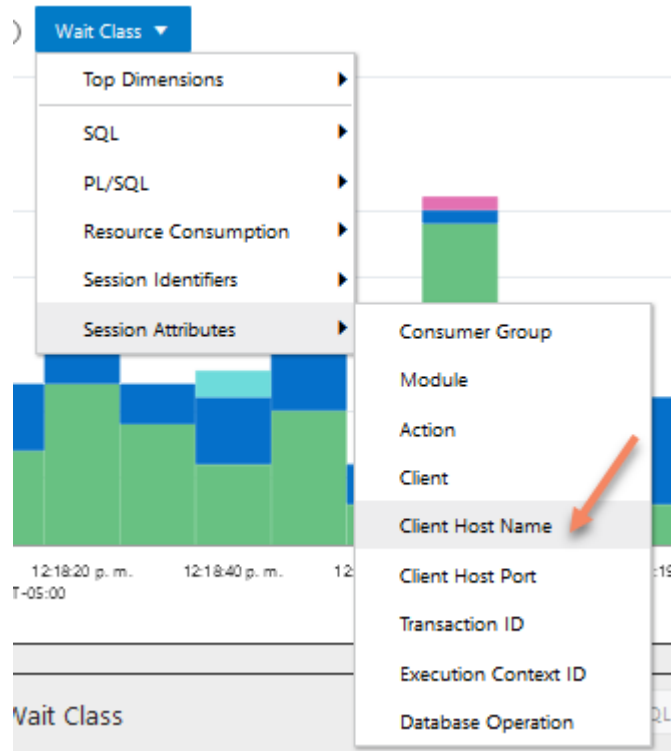
Análisis ASH

- CPU – verde
- IO – Azul
- Bloqueos – Rojo



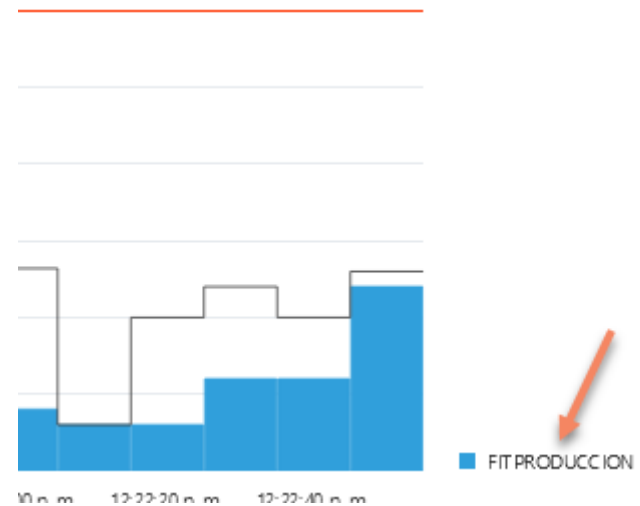
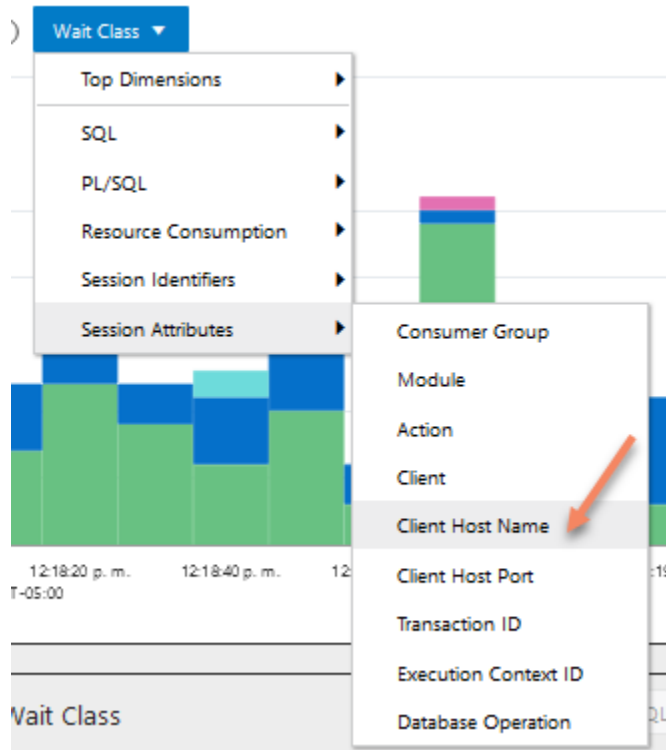
> 24 | Que revisar

Filtrar por varias opciones



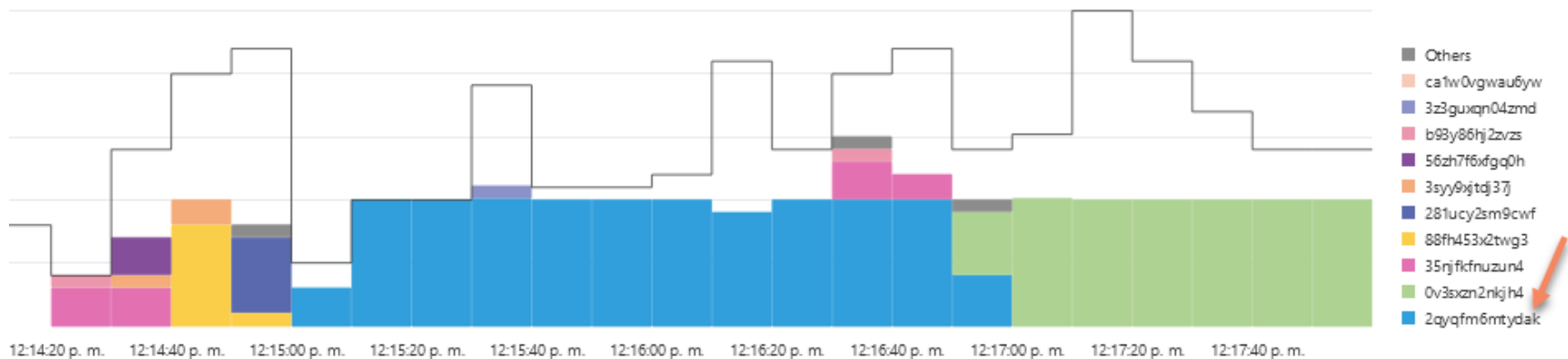
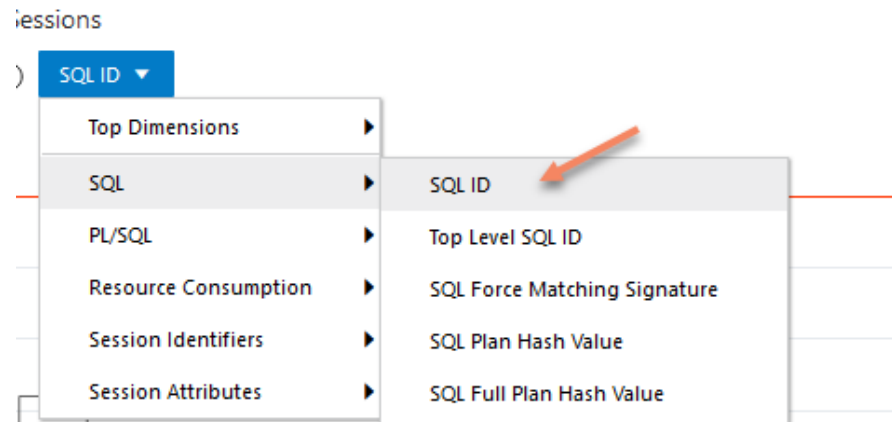
> 24 | Que revisar

Filtrar por varias opciones



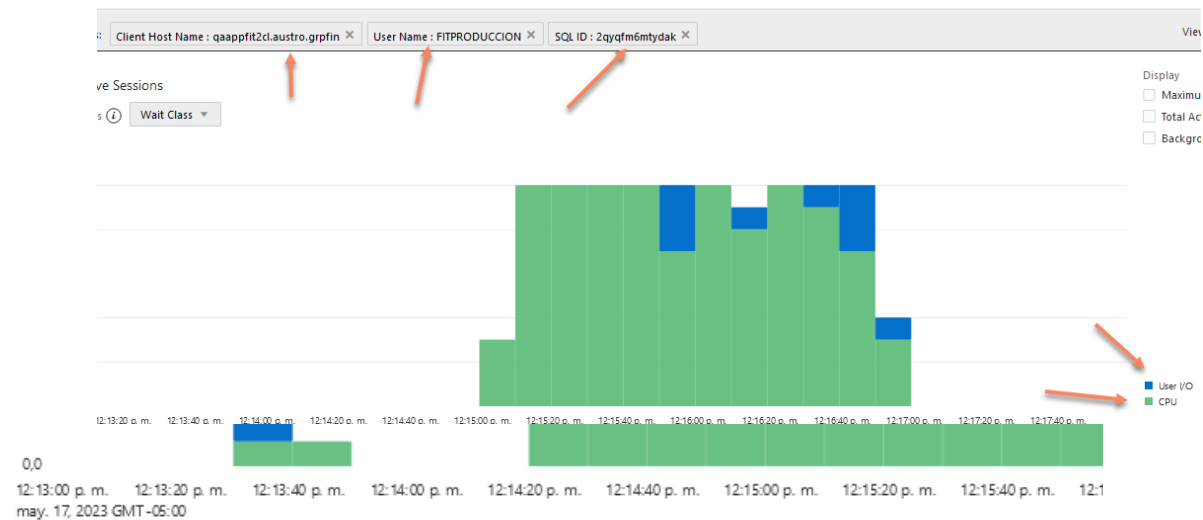
> 24 | Que revisar





Filtrar por varias opciones



> 24 | Que revisar

Filtrar por varias opciones



SQL ID by Wait Class					Tune SQL	Create SQL Tuning Set	Columns	
SQL ID	Activity (Average Active Sessions)	SQL Plan Hash Value	SQL Type	Service				
2qyqfm6mtydak	 0,35	433340160	SELECT	fitqa				
0v3sxn2nkjh4	 0,22	1264510536	UPDATE	fitqa				
35njfknuzun4	 0,04	793770647	SELECT	fitqa				
88fh453x2twg3	 0,03	3898379740	SELECT	fitqa				



> 24

Que revisar

Copiar el SQLID, y buscar

SQL ID	Activity (Average Active Sessions)	SQL Pla
2qyqfm6mtydak	0,35	433340
0v3sxn2nkjh4	0,22	126451
...

Rendimiento
Disponibilidad
Seguridad
Esquema
Administración

Página Inicial de Rendimiento
Hub de Rendimiento
SQL
AWR
Página Inicial de Asesores
ADDM en Tiempo Real
Sesiones de Búsqueda
Sesiones Bloqueantes

Asesor de Ajustes SQL
Inicio de SQL Performance Analyzer
Configuración de Comprobación Rápida de SQL
Asesor de Acceso SQL
Juegos de Ajustes SQL
Control de Plan SQL
Estadísticas del Optimizador
Historial SQL de Cloud Control
Buscar SQL...

Texto SQL

LIKE

SQL ID

=

2qyqfm6mtydak

Tiempo Transcurrido (seg)

>=

Por defecto, la búsqueda devuelve todas las coincidencias no sensibles a mayúsculas/minúsculas que comienzan por la cadena introducida. Para ejecutar una búsqueda de coincidencia exacta utilizar el símbolo comodín (%) en la cadena entrecomillada.

Buscar

Mostrar SQL

Resultados (1 sentencias)

Caché de Cursores (1 sentencias)

Instantáneas AWR

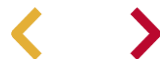
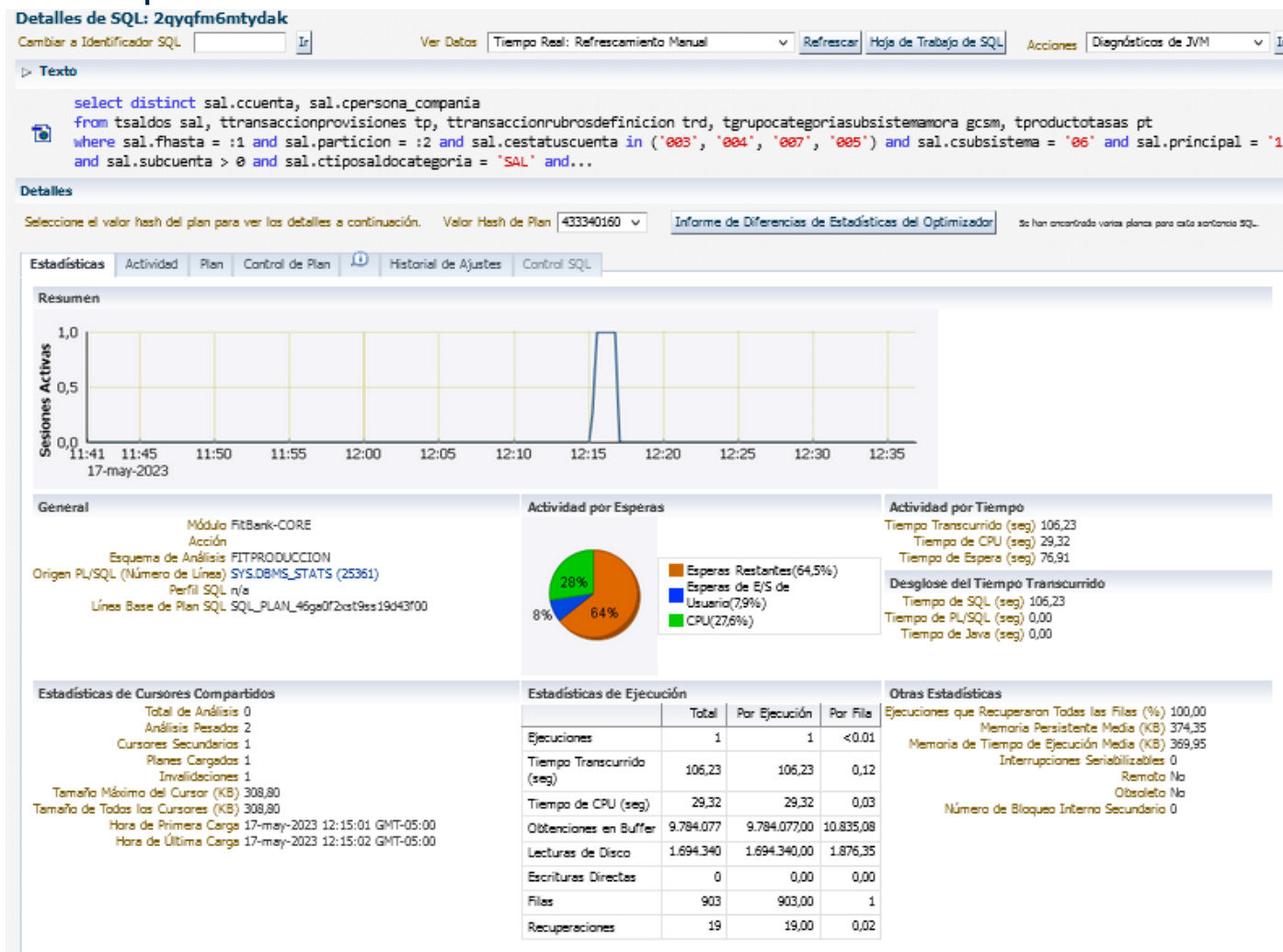
Línea Base de AWR

Juego de Ajustes SQL

SQL ID	Texto SQL	Valor Hash de Plan	Nombre de Esquema de
2qyqfm6mtydak	select distinct sal.cuenta, sal.cp persona_compania...	433340160	FITPRODUCCION

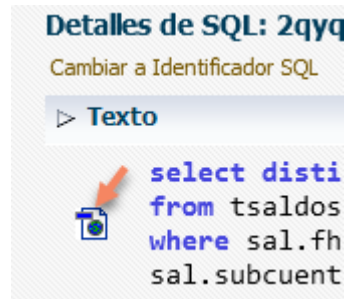
> 24 | Que revisar

Pantalla de información del query



> 24 | Que revisar

Revisar código del query



```
)
UNION
SELECT DISTINCT sal.ccuenta,
sal.cpersona_compania
FROM tsaldos sal
WHERE sal.fhasta = :8
AND sal.particion = :9
AND sal.cestatuscuenta IN ('003', '004', '007',
AND sal.csubsistema = '06'
AND sal.principal = '1'
AND sal.subcuenta > 0
AND sal.ctiposaldocategoria = 'SAL'
AND sal.cpersona_compania = '2'
AND sal.categoria IN ('CAPPRO', 'CAPCAS')
AND sal.saldomonedacuenta > 0
AND sal.fvencimiento <= :10
AND NOT EXISTS
(
SELECT 1
FROM tsaldos ts
WHERE ts.ccuenta = sal.ccuenta
AND ts.fhasta = :11
AND ts.subcuenta = sal.subcuenta
AND ts.categoria = :12
AND ts.subcuenta = sal.subcuenta
)
UNION
SELECT DISTINCT sal.ccuenta,
sal.cpersona_compania
FROM
```

> 24

Que revisar

Buscar sentencias que consuman

```
)
UNION
SELECT DISTINCT sal.ccuenta,
                sal.cpersona_compania
FROM tsaldos sal
WHERE sal.fhasta = :8
      AND sal.particion = :9
      AND sal.cestatuscuenta IN ('003', '004', '007',
                                '06')
      AND sal.csubsistema = '06'
      AND sal.principal = '1'
      AND sal.subcuenta > 0
      AND sal.ctiposaldocategoria = 'SAL'
      AND sal.cpersona_compania = '2'
      AND sal.categoria IN ('CAPPRO', 'CAPCAS')
      AND sal.saldomonedacuenta > 0
      AND sal.fvencimiento <= :10
      AND NOT EXISTS
      (
        SELECT 1
        FROM tsaldos ts
        WHERE ts.ccuenta = sal.ccuenta
              AND ts.fhasta = :11
              AND ts.subcuenta = sal.subcuenta
              AND ts.categoria = :12
              AND ts.subcuenta = sal.subcuenta
      )
UNION
SELECT DISTINCT sal.ccuenta,
                sal.cpersona_compania
```

like

0 de 0




> 04 | Que revisar

Revisar la composición del query en busca de excesivos selects anidados

```

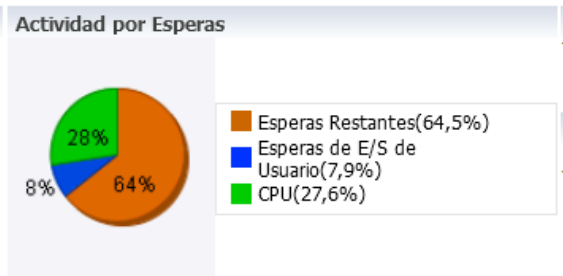
    ELSE
END estado
tc.fdesde fchUltimoCambio,
ts.saldo saldoAlCorte
NVL(ROUND(
(
    SELECT tpro.promediomensual
    FROM tcuentasaldospromedio tpro
    WHERE tpro.ccuenta = tc.ccuenta
      AND tpro.cpersona_compania = 2
      AND to_date(tpro.fpromedio, 'yyyymm') = ADD_MONTHS(to_date('202109', 'yyyymm'), -6 )
)
,2),0.000) saldoPromedio6,
NVL(ROUND(
(
    SELECT tpro.promediomensual
    FROM tcuentasaldospromedio tpro
    WHERE tpro.ccuenta = tc.ccuenta
      AND tpro.cpersona_compania = 2
      AND to_date(tpro.fpromedio, 'yyyymm') = ADD_MONTHS(to_date('202109', 'yyyymm'), -5 )
)
,2),0.0000) saldoPromedio5 ,
NVL(ROUND(
(
    SELECT tpro.promediomensual
    FROM tcuentasaldospromedio tpro
    WHERE tpro.ccuenta = tc.ccuenta
      AND tpro.cpersona_compania = 2
      AND to_date(tpro.fpromedio, 'yyyymm') = ADD_MONTHS(to_date('202109', 'yyyymm'), -4 )
)
,2),0.0000) saldoPromedio4 ,
NVL(ROUND(
(
    SELECT tpro.promediomensual
    FROM tcuentasaldospromedio tpro
    WHERE tpro.ccuenta = tc.ccuenta
```



> 24

Que revisar

Revisar las estadísticas y los posibles planes que puede utilizar



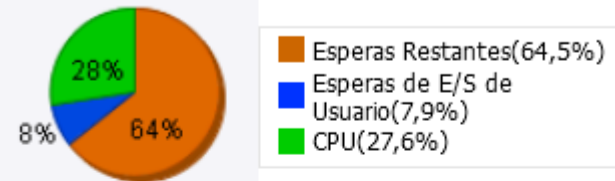
Estadísticas de Ejecución

	Total	Por Ejecución	Por Fila
Ejecuciones	1	1	<0.01
Tiempo Transcurrido (seg)	106,23	106,23	0,12
Tiempo de CPU (seg)	29,32	29,32	0,03
Obtenciones en Buffer	9.784.077	9.784.077,00	10.835,08
Lecturas de Disco	1.694.340	1.694.340,00	1.876,35
Escrituras Directas	0	0,00	0,00
Filas	903	903,00	1
Recuperaciones	19	19,00	0,02

- TIEMPO PROMEDIO
- CPU
- LECTURAS DE DISCO
- Al menos ejecutar 10 veces las consultas, para obtener un buen promedio.
 - Las primeras ejecuciones realiza la carga a memoria, parseo, selección de plan de ejecución, etc, lo que puede generar mayor consumo

<https://ronaldvargasquesada.wordpress.com/2009/07/23/el-factor-parseo-en-una-instruccion-sql/>

Actividad por Esperas

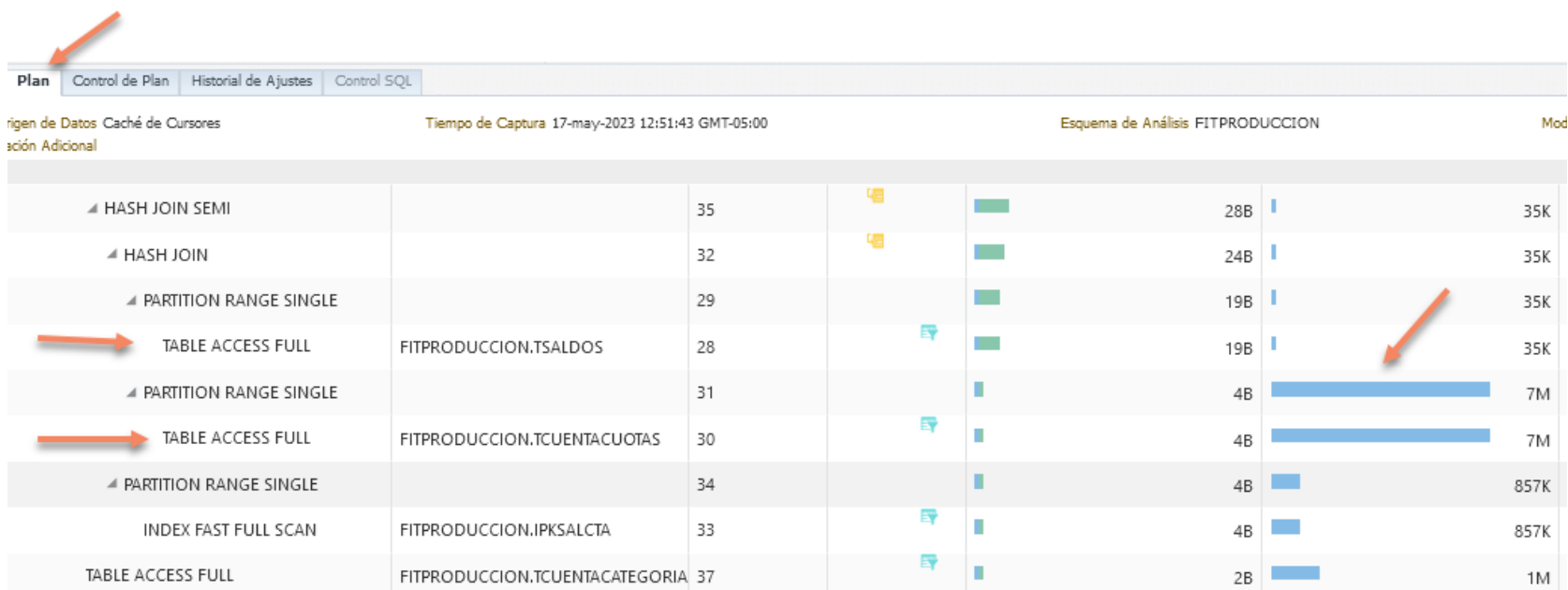


Estadísticas de Ejecución

	Total	Por Ejecución	Por Fila
Ejecuciones	1	1	<0.01
Tiempo Transcurrido (seg)	106,23	106,23	0,12
Tiempo de CPU (seg)	29,32	29,32	0,03
Obtenciones en Buffer	9.784.077	9.784.077,00	10.835,08
Lecturas de Disco	1.694.340	1.694.340,00	1.876,35
Escrituras Directas	0	0,00	0,00
Filas	903	903,00	1
Recuperaciones	19	19,00	0,02

> 24 | Que revisar

Revisar el plan de ejecución por table access full, u operaciones que estén causando demoras en el query



The screenshot shows a database query execution plan. An orange arrow points to the 'Plan' tab in the top navigation bar. Another orange arrow points to the 'TABLE ACCESS FULL' operation for the table 'FITPRODUCCION.TSALDOS'. A third orange arrow points to the 'TABLE ACCESS FULL' operation for the table 'FITPRODUCCION.TCUENTACUOTAS'. The plan shows various operations including HASH JOIN SEMI, HASH JOIN, PARTITION RANGE SINGLE, and INDEX FAST FULL SCAN. The 'TABLE ACCESS FULL' operations are highlighted with orange arrows, indicating they are the focus of the review.

Operation	Table	Cost	IO	Memory	Temp	Rows
HASH JOIN SEMI		35		28B		35K
HASH JOIN		32		24B		35K
PARTITION RANGE SINGLE		29		19B		35K
TABLE ACCESS FULL	FITPRODUCCION.TSALDOS	28		19B		35K
PARTITION RANGE SINGLE		31		4B		7M
TABLE ACCESS FULL	FITPRODUCCION.TCUENTACUOTAS	30		4B		7M
PARTITION RANGE SINGLE		34		4B		857K
INDEX FAST FULL SCAN	FITPRODUCCION.IPKSALCTA	33		4B		857K
TABLE ACCESS FULL	FITPRODUCCION.TCUENTACATEGORIA	37		2B		1M

> 24 | Que revisar

Revisar el plan de ejecución por table access full, u operaciones que estén causando demoras en el query

Text

```
SELECT *  
FROM TMOVIMIENTOS TM  
WHERE to_char(FREAL, 'yyyymmdd') = 20210927
```

Details



Select the plan hash value to see the details below. Plan Hash Value

Statistics Activity **Plan** Plan Control Tuning History SQL Monitoring

Data Source Cursor Cache Capture Time Sep 27, 2021 12:18:54 PM GMT-05:00 Parsing Schema I

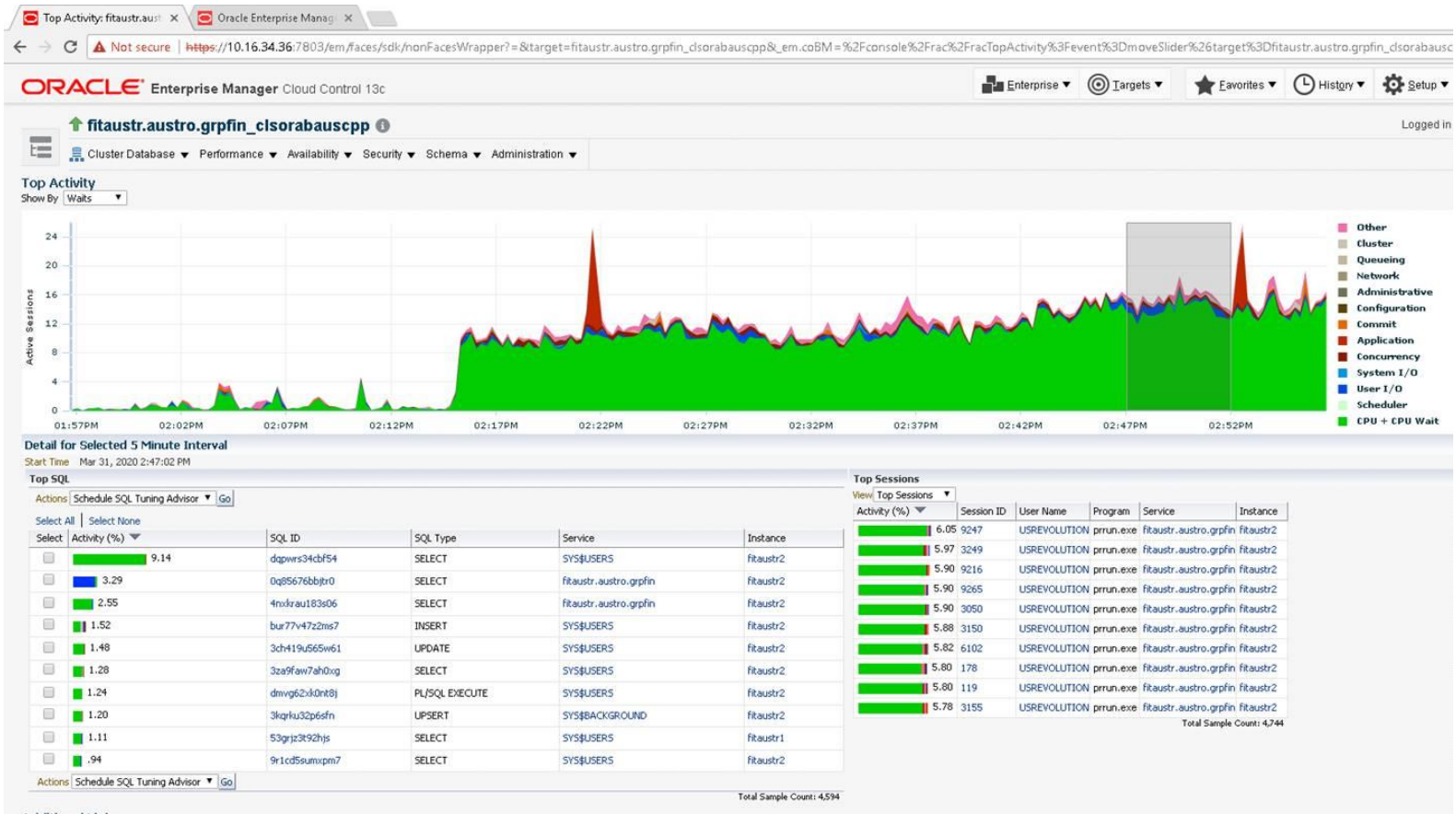
Additional Information

☐ Graphical ☒ Tabular

Operation	Object	Order	Predicate	Pruning	Operation Cost
SELECT STATEMENT		4			
PARTITION RANGE ALL		3			
PARTITION HASH ALL		2			
TABLE ACCESS FULL	FITCANALES.TMOVIMIENTOS	1			 1,902K

> 25 | Que se puede hacer

- Usar binds (variables enlazadas)



> 25 | Que se puede hacer

- Usar binds (variables enlazadas)
- Ejemplo de consulta que no usa bindings, la misma consulta donde solo varía el código de empleado genera 2 SQLID por este motivo.

SQL Details: gsjhb0h9xcw46

Switch Database Instance: fitaustr.austro.grpfin_clsorabauscpp_fitaustr2 Go

Switch to SQL ID: Go

Text

```
SELECT SUM ( A.VALOR )  
FROM PR_11 A , PR_33 B , PR_36 C  
WHERE B.COD_RUBRO =A.COD_RUBRO AND B.COD_GRUPO_RUB =3 AND A.CODIGO = '322' AND B.STATUS = 'A' AND  
YYYY HH24:MI:SS') AND TO_TIMESTAMP('30-06-2020 00:00:00','DD-MM-YYYY HH24:MI:SS') ) OR ( 2020 >
```

Details

SQL Details: ah6zu6k5c8q5u

Switch Database Instance: fitaustr.austro.grpfin_clsorabauscpp_fitaustr2 Go

Switch to SQL ID: Go

Text

```
SELECT SUM ( A.VALOR )  
FROM PR_11 A , PR_33 B , PR_36 C  
WHERE B.COD_RUBRO =A.COD_RUBRO AND B.COD_GRUPO_RUB =36 AND A.CODIGO = '9350' AND B.STATUS = 'A' AND  
YYYY HH24:MI:SS') AND TO_TIMESTAMP('30-06-2020 00:00:00','DD-MM-YYYY HH24:MI:SS') ) OR ( 2020 > 0
```

Details

> 25 | Que se puede hacer

- Usar binds
- Ejemplo de consulta que usa binds

```
SELECT *  
FROM  
(SELECT *  
FROM TLOGINTERFACES t  
WHERE t.TIPOMENSAJE = :1 AND t.NOMBREPERSONAINGRESO = :2 AND t.ESTADO = :3 AND t.SESION = :4 ORDER BY t.FREAL DESC)  
WHERE ROWNUM < 2
```

> 25 | Que se puede hacer

- Reestructurar las consultas
- Cambiar Likes por “=”
- Cambiar cruces de tablas con “,” por joins
- Fncfhasta reemplazar con `to_date('29991231','yyyymmdd')`
- Revisar los índices de las tablas y usarlos adecuadamente en las consultas
- No usar demasiados commits

> 25 | Que se puede hacer

- Usar tablas temporales
- Usar Tablas CTE
- Cambiar subselects por joins
- Concepto de Divide y vencerás

```
nv1((select tpd1.DIRECCION from FITAUSTROP.M.TPERSONADIR
where tpd1.CPERSONA = tp.cpersona AND tpd1.fhasta = to_
AND tpd1.CTIPODIRECCION = 'CE'and numerodireccion =
(select max(tpd2.numerodireccion) from FITAUSTROP.M.TPER
where tpd2.CPERSONA = tpd1.CPERSONA AND tpd2.fhasta =to
AND tpd2.CTIPODIRECCION = 'CE')),
' ') Correo,
```

```
nv1((SELECT NUMEROTELEFONO FROM FITAUSTROP.M.tpersonatel
INNER JOIN FITAUSTROP.M.TPERSONADIRECCIONES TPD ON TT.CP
AND TPD.NUMERODIRECCION = TT.NUMERODIRECCION AND TPD.FH
AND tpd.direccionprincipal = 1
WHERE TT.cpersona = TP.CPERSONA and TT.fhasta = to_date
and TT.CTIPOTELEFONO = 'TEL' AND TT.CPERSONA = TP.CPERS
and stelefono =
(sELECT max(stelefono) FROM FITAUSTROP.M.tpersonatelefon
INNER JOIN FITAUSTROP.M.TPERSONADIRECCIONES TPD ON TT.CP
AND TPD.NUMERODIRECCION = TT.NUMERODIRECCION AND TPD.FH
and tpd.direccionprincipal = 1
WHERE TT.cpersona = TP.CPERSONA and TT.fhasta = to_date
and TT.CTIPOTELEFONO = 'TEL' AND TT.CPERSONA = TP.CPERS
' ') Telefono_Fijo,
```

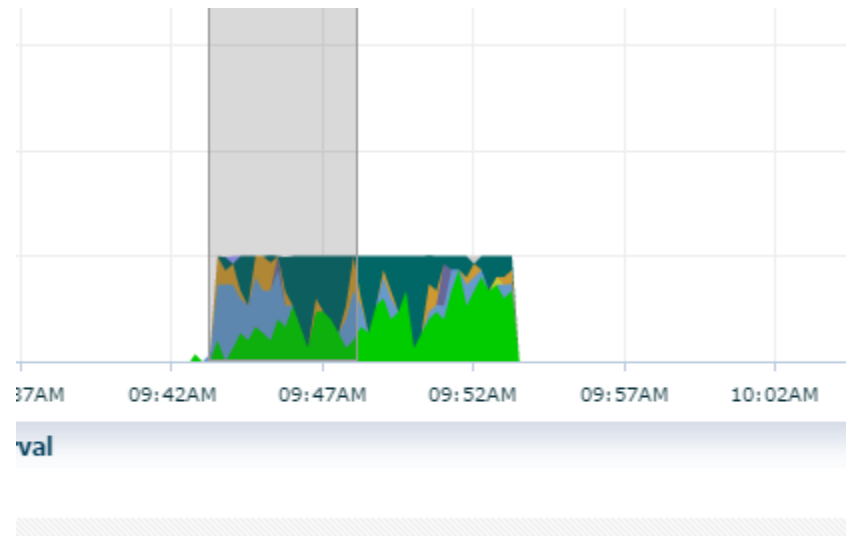
```
nv1((select TF.NUMEROTELEFONO from fitaustrop.m.tpersona
where TF.cpersona = TP.CPERSONA and fhasta = to_date('2
and CTIPOTELEFONO = 'CEL' and TF.stelefono =
(select max(TF2.stelefono) from fitaustrop.m.tpersonatel
where TF2.cpersona = TF.CPERSONA and TF2.fhasta = to_da
' ') Telefono_Movil,
```

> 25 | Que se puede hacer

- Usar tablas temporales
- Usar Tablas CTE
- Cambiar subselects por joins
- Concepto de Divide y vencerás

Antes de reestructurar

	NOMBRE	APELLIDO	TIPO_C
1	CARLOS VICENTE	MATUTE MATUTE	2
2	ROXANA MARIUXI	CONFORME ZAMBRANO	2
3	RUBEN DARIO	RUIZ ANDALUZ	2
4	JEANNETH PAULINA	ORTIZ OÑATE	2
5	ROSSY JIMABELL	AGUILAR MENDOZA	2
6	VANESSA ELIZABETH	SUAREZ DE LA A	2
7	CAROLINA CRISTINA	REYES ALVAREZ	2



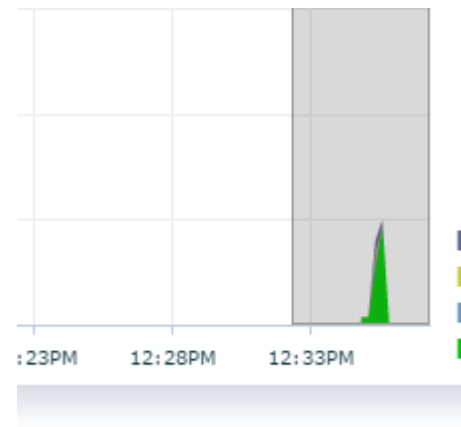
> 25 | Que se puede hacer

- Usar tablas temporales
- Usar Tablas CTE
- Cambiar subselects por joins
- Concepto de Divide y vencerás

Luego de reestructurar

	NOMBRE	APELLIDO	TIPO_CONTACTO
1	WILVER ROSEMBER	INTRIAGO LOOR	2
2	AURELIO RODRIGO	CAMPOVERDE CABRERA	2
3	ANDRES XAVIER	TORRES PALACIOS	2
4	ALEX JAVIER	PACHECO TITUANA	2
5	LUZ MARIA	CACERES VINUEZA	2
6	JEANETTE JESSENIA	OROZCO MURILLO	2

581:1 → 0:54 bcriollo@RAC_PADRE



> 25 | Que se puede hacer

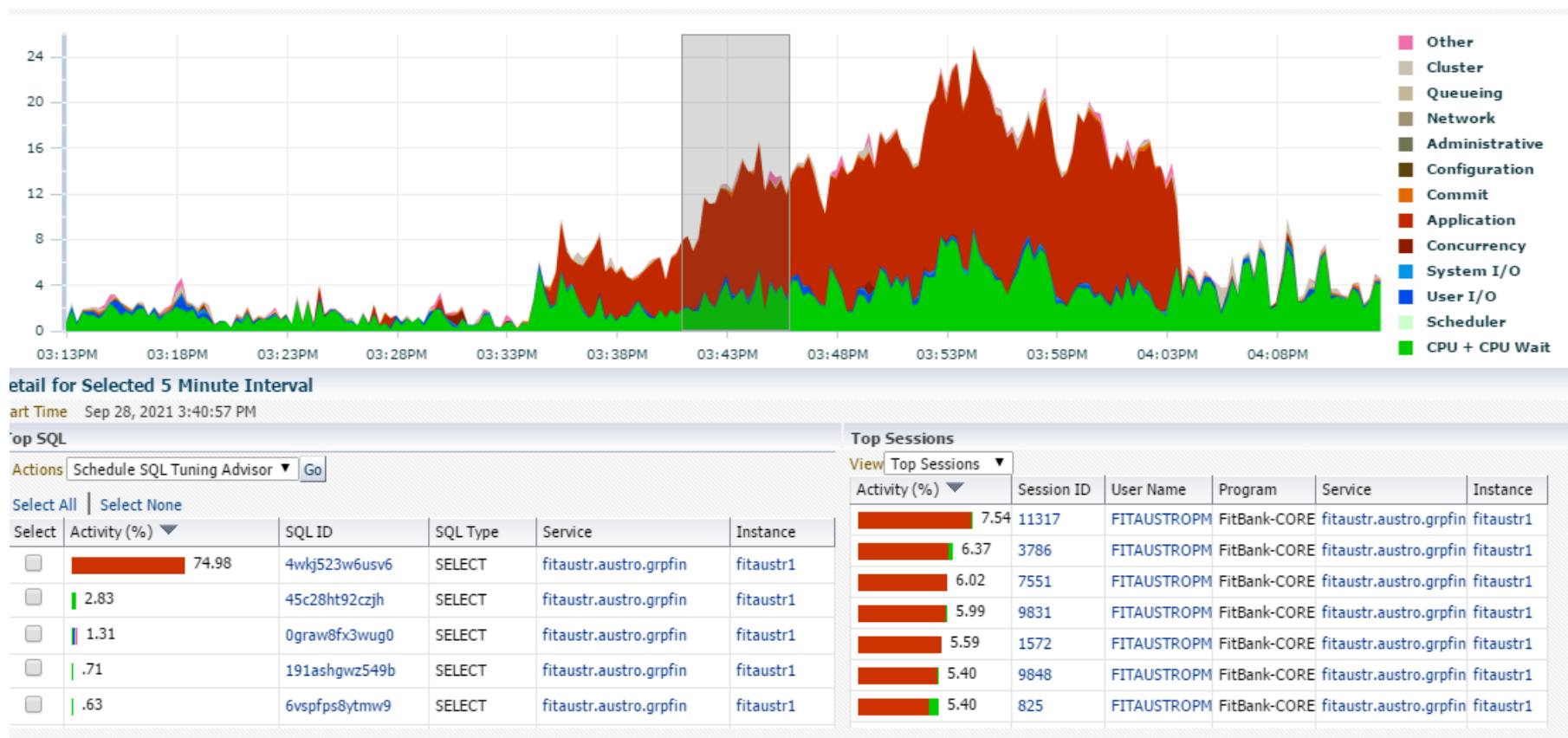
- Usar tablas temporales
- Usar Tablas CTE
- Cambiar subselects por joins
- Concepto de Divide y vencerás

Validar resultados

BORIS ALEJANDRO	CRIOLLO ZUÑIGA	2	CORISBRIOLLO@GMAIL.COM		0998076975	EC0101010009	1983-07-25	19	0104447453	TULCAN 1 22 Y 24 DE MAYO
BORIS ALEJANDRO	CRIOLLO ZUÑIGA	2	CORISBRIOLLO@GMAIL.COM		0998076975	EC0101010009	1983-07-25	19	0104447453	TULCAN 1 22 Y 24 DE MAYO
STEFANY ALEJANDRA	QUENGUAN DIAZ	2	STEFANY1996@HOTMAIL.COM		0990060936	EC2222010050	1996-05-03	20	0000523577	12 DE FEBRERO Y JOSE FELICIANO
STEFANY ALEJANDRA	QUENGUAN DIAZ	2	STEFANY1996@HOTMAIL.COM		0990060936	EC2222010050	1996-05-03	20	0000523577	12 DE FEBRERO Y JOSE FELICIANO
LAURA INES	VALDIVIESO MONTESINOS	2	VALDIVIEZO28@GMAIL.COM	2906924	0997195985	EC0101010000	1953-10-28	19	0100871219	AUTOPISTA PUEBLO AWA ENTRANC
LAURA INES	VALDIVIESO MONTESINOS	2	VALDIVIEZO28@GMAIL.COM	2906924	0997195985	EC0101010000	1953-10-28	19	0100871219	AUTOPISTA PUEBLO AWA ENTRANC
RODRIGO EDUARDO	LOPEZ ASTUDILLO	2	eddy@neoingenieria.com	2834101	0991105530	EC0101010000	1953-09-08	19	0101041739	GIL RAMIREZ DAVALOS HUMBOLT
RODRIGO EDUARDO	LOPEZ ASTUDILLO	2	eddy@neoingenieria.com	2834101	0991105530	EC0101010000	1953-09-08	19	0101041739	GIL RAMIREZ DAVALOS HUMBOLT

> 25 | Que se puede hacer

- No usar “Select For Update” para actualizar o ingresar datos las tablas
 - Provoca bloqueos



> 25 | Que se puede hacer

- No usar “Select For Update” para actualizar o ingresar datos las tablas
 - Se debe usar Updates e Inserts

```
select tbalance0_.CCUENTA as CCUENTA2180_, tbalance0_.FHASta as FHASta2180_, tbalance0_.PARTICION  
CATEGORIA2180_, tbalance0_.SUBCUENTA as SUBCUENTA2180_, tbalance0_.SSUBCUENTA as SSUBCUENTA2180_,  
tbalance0_.CSUCURSAL as CSUCURSAL2180_, tbalance0_.COFICINA as COFICINA2180_, tbalance0_.CMONEDA_C
```

```
FROM      TSALDOS tbalance0_  
WHERE     tbalance0_.CCUENTA=:1  
AND  
        (  
            :2 BETWEEN tbalance0_.FDESDE AND tbalance0_.FHASta  
        )  
AND tbalance0_.CPERSONA_COMPANIA=:3  
AND tbalance0_.PARTICION >=:4  
ORDER BY tbalance0_.SUBCUENTA,  
         tbalance0_.SSUBCUENTA DESC FOR UPDATE
```

> 26 | Continuar investigando:

- Seguir buscando buenas prácticas en la documentación de internet

how to improve performance sql

× | 🔍

🔍 Todo 📺 Videos 🖼️ Imágenes 📰 Noticias 📍 Maps ⋮ Más Herramientas

Cerca de 48.800.000 resultados (0,39 segundos)

How Can You Select Which Queries to Optimize?


1. Consistently Slow Queries. ...
2. Occasionally Slow Queries. ...
3. Queries With Red Flags. ...
4. Queries That Majorly Contribute to Total Execution Time. ...
5. Define Your Requirements. ...
6. Reduce Table Size. ...
7. Simplify Joins. ...
8. Use SELECT Fields FROM Instead of SELECT * FROM.

Más elementos... • 1 nov 2021

<https://www.tek-tools.com> > Systems

How To Optimize SQL Queries Performance - Best Practices

🔍 Información sobre los fragmentos destacados • 🗨️ Comentarios



<https://blog.devart.com> > how-to-o... ▼ Traducir esta página

SQL Query Optimization: 12 Useful Performance Tuning Tips ...

Query optimization tips for **better performance** — In the article, we are going to examine how to optimize SQL queries and improve query performance by using ...

[SQL query optimization basics](#) · [Query optimization tips for...](#)

<https://www.developer.com> > 10-w... ▼ Traducir esta página

10 Ways to Improve SQL Query Performance - Developer.com

10 sept 2015 — Database performance is one of the most crucial aspects of any application. Learn how to improve query performance by following these ten ...

<https://www.sqlshack.com> > query-... ▼ Traducir esta página

Query optimization techniques in SQL Server: tips and tricks

19 jun 2018 — Fixing bad queries and resolving performance problems can involve hours ... there is a potentially easy fix to improve query performance.

banco
del  austro