

## Laboratorio Consultas Indices-Estadísticas

Entregar para cada una la consulta, el resultado y el plan de ejecución.

### 1. Agrupar por estado la cantidad de clientes que tiene la empresa.

Consulta

```
SELECT state, COUNT(*)  
FROM customers  
GROUP BY state;
```

Resultado

NY	1019
TX	142
CA	284

Plan de ejecución

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				8
HASH		GROUP BY		8
TABLE ACCESS	CUSTOMERS	FULL		7

### 2. Agrupar por estado la cantidad de clientes que tiene la empresa y que su apellido sea "Vega".

Consulta

```
SELECT state, COUNT(*)  
FROM customers  
WHERE last_name = 'Vega'  
GROUP BY state;
```

Resultado

NY	4
CA	2

Plan de ejecución

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				8
HASH		GROUP BY		8
TABLE ACCESS	CUSTOMERS	FULL		7

### 3. La cantidad en stock por “año del modelo”.

Consulta

```
SELECT p.model_year, SUM(s.quantity)
FROM stocks s, products p
WHERE s.product_id = p.product_id
GROUP BY p.model_year;
```

Resultado

2017	3787
2016	1046
2018	8678

Plan de ejecución

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			939	6
HASH		GROUP BY	939	6
HASH JOIN			939	5
Access Predicates S.PRODUCT_ID=P.PRODUCT_ID				
VIEW	index\$_join\$_002		321	2
HASH JOIN				
Access Predicates ROWID=ROWID				
INDEX	IDX_MOLDE_YEAR	FAST FULL SCAN	321	1
INDEX	PK_PRODUCTS_47027DE54A8E86AD	FAST FULL SCAN	321	1
TABLE ACCESS	STOCKS	FULL	939	3

### 4. La cantidad en stock por “año del modelo” y que la categoría sea “Mountain Bikes” y “Road Bikes”.

Consulta

```
SELECT p.model_year, SUM(s.quantity)
FROM Products p, Stocks s, categories c
WHERE s.product_id = p.product_id
AND p.category_id = c.category_id
AND category_name in('Mountain Bikes', 'Road Bikes')
GROUP BY p.model_year;
```

Resultado

2017	1357
2016	275
2018	3113

Plan de ejecución

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST	LAST_CR_BUFFER_GETS	LAST_ELAPSED_TIME
SELECT STATEMENT				10	16	953
HASH		GROUP BY		268	10	953
HASH JOIN				268	9	639
Access Predicates S.PRODUCT_ID=P.PRODUCT_ID						
TABLE ACCESS	STOCKS	FULL	939	3	7	66
MERGE JOIN				92	6	168
TABLE ACCESS	CATEGORIES	BY INDEX RO...	2	2	2	19
Filter Predicates OR CATEGORY_NAME='Mountain Bikes' CATEGORY_NAME='Road Bikes'						
INDEX	PK_CATEGORY_DS_...	FULL SCAN	7	1	1	9
Sort						
Access Predicates P.CATEGORY_ID=C.CATEGORY_ID						
Filter Predicates P.CATEGORY_ID=C.CATEGORY_ID						
TABLE ACCESS	PRODUCTS	FULL	321	3	7	51

## 5. Qué tienda tiene el mayor stock.

Consulta

```
SELECT * FROM (
SELECT store_name, SUM(quantity)
FROM Stocks s, Stores st
WHERE s.store_id = st.store_id
GROUP BY store_name
ORDER BY 2 DESC)
WHERE rownum = 1;
```

Resultado

Rowlett Bikes	4620
---------------	------

Plan de ejecución

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	8
COUNT		STOPKEY		
VIEW			939	8
SORT		ORDER BY STOPKEY	939	8
HASH		GROUP BY	939	8
MERGE JOIN			939	6
TABLE ACCESS	STORES	BY INDEX ROWID	3	2
INDEX	PK_STORES_A2E2A30CF92043A1	FULL SCAN	3	1
SORT		JOIN	939	4
Access Predicates				
S.STORE_ID=ST.STORE_ID				
Filter Predicates				
S.STORE_ID=ST.STORE_ID				
TABLE ACCESS	STOCKS	FULL	939	3

## 6. Empleados con su cantidad de ventas (mostrar el nombre completo y la cantidad).

Consulta

```
SELECT first_name, last_name, COUNT(order_id)
FROM staffs sf, orders o
WHERE sf.staff_id = o.staff_id
GROUP BY first_name, last_name;
```

Resultado

Marcelene	Boyer	553
Mireya	Copeland	164
Genna	Serrano	184
Kali	Vargas	88
Layla	Terrell	86
Venita	Daniel	540

Plan de ejecución

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1615	9
HASH		GROUP BY	1615	9
MERGE JOIN			1615	8
TABLE ACCESS	STAFFS	BY INDEX ROWID	10	2
INDEX	PK_STAFFS_1963DD9C67F51F64	FULL SCAN	10	1
SORT		JOIN	1615	6
Access Predicates				
SF.STAFF_ID=O.STAFF_ID				
Filter Predicates				
SF.STAFF_ID=O.STAFF_ID				
TABLE ACCESS	ORDERS	FULL	1615	5

## 7. Empleados con su cantidad de ventas (mostrar el nombre completo y la cantidad) y que sea de la tienda “Rowlett Bikes”.

Consulta

```
SELECT first_name, last_name, COUNT(order_id)
FROM staffs sf, orders o, stores s
WHERE sf.staff_id = o.staff_id
AND s.store_id = sf.store_id
AND store_name = 'Rowlett Bikes'
GROUP BY first_name, last_name;
```

Resultado

Kali	Vargas	88
Layla	Terrell	86

Plan de ejecución

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST	LAST_CR_BUFFER_GETS	LAST_ELAPSED_TIME
SELECT STATEMENT						
HASH						
HASH JOIN						
Access Predicates						
S.PRODUCT_ID=P.PRODUCT_ID						
TABLE ACCESS	STOCKS	FULL	939	3	7	68
MERGE JOIN						
TABLE ACCESS	BRANDS	BY INDEX ROWID...	204	6	6	360
INDEX	PK_BRANDS_SESA...	FULL SCAN	9	2	2	173
INDEX			9	1	1	163
JOIN			204	4	4	160
SORT						
Access Predicates						
Filter Predicates						
P.BRAND_ID=B.BRAND_ID						
TABLE ACCESS	PRODUCTS	BY INDEX ROWID...	204	3	4	81
INDEX	IDX_MOLDE_YEAR	RANGE SCAN	204	1	1	34
Access Predicates						
MODEL_YEAR=2018						

## 8. Stock de bicicletas por marca.

Consulta

```
SELECT b.brand_name, SUM(s.quantity)
FROM stocks s, products p, brands b
WHERE s.product_id = p.product_id
AND p.brand_id = b.brand_id
GROUP BY b.brand_name;
```

Resultado

Strider	136
Pure Cycles	104
Surly	1105
Electra	4998
Haro	454
Ritchey	45
Trek	5519
Heller	108
Sun Bicycles	1042

Plan de ejecución

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				
HASH				
HASH JOIN				
Access Predicates				
S.PRODUCT_ID=P.PRODUCT_ID				
TABLE ACCESS	STOCKS	FULL	939	3
MERGE JOIN				
TABLE ACCESS	BRANDS	BY INDEX ROWID	9	2
INDEX	PK_BRANDS_SESA...	FULL SCAN	9	1
JOIN			321	4
Access Predicates				
P.BRAND_ID=B.BRAND_ID				
Filter Predicates				
P.BRAND_ID=B.BRAND_ID				
TABLE ACCESS	PRODUCTS	FULL	321	3

## 9. Stock de bicicletas por marca y que el año del modelo sea 2018.

Consulta

```
SELECT b.brand_name, SUM(s.quantity)
FROM stocks s, products p, brands b
WHERE s.product_id = p.product_id
AND p.brand_id = b.brand_id
AND model_year = 2018
GROUP BY b.brand_name;
```

Resultado

Strider	136
Surly	524
Electra	3680
Trek	4256
Heller	82

Plan de ejecución

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST	LAST_CR_BUFFER_GETS	LAST_ELAPSED_TIME
SELECT STATEMENT						
HASH						
HASH JOIN						
Access Predicates						
S.PRODUCT_ID=P.PRODUCT_ID						
TABLE ACCESS	STOCKS	FULL	939	3	7	84
MERGE JOIN			204	6	6	258
TABLE ACCESS	BRANDS	BY INDEX RO...	9	2	2	24
INDEX	PK_BRANDS_SESA...	FULL SCAN	9	1	1	11
JOIN			204	4	4	208
SORT						
Access Predicates						
P.BRAND_ID=B.BRAND_ID						
Filter Predicates						
P.BRAND_ID=B.BRAND_ID						
TABLE ACCESS	PRODUCTS	BY INDEX RO...	204	3	4	132
INDEX	IDX_MODEL_YEAR	RANGE SCAN	204	1	1	78
Access Predicates						
MODEL_YEAR=2018						

De lo observado de las consultas anteriores contesta las siguientes preguntas

1. ¿Qué pasa cuando una consulta no tiene condiciones?

Cuando una consulta no tiene condiciones es más rápida, pues es más fácil el acceso. Por lo tanto el costo también es menor.

2. ¿Qué pasa cuando a una consulta se le agregan condiciones?

Pues el costo aumenta, ya que el plan de ejecución es más complejo.

3. ¿Para qué sirve un plan de ejecución SQL?

Para poder decidir por la opción que nos genere menos costos y menor tiempo de ejecución.

4. ¿Cómo podemos modificar el plan de ejecución?

Creando índices.

5. ¿Cuál es el motor de búsqueda? Y ¿Quién lo implementa?

Es lo que nos permite buscar y obtener alguna especificación que nosotros definimos, en la base de datos.

6. ¿Qué son las estadísticas y para qué sirven las mismas en el RDBMS?

Son cálculos basados en el manejo de los datos, generalmente se implementan para mejorar las consultas.

**Crea los índices necesarios que se pueden usar en Q2, Q4, Q7 y Q9. Ejecuta nuevamente tu consulta y compara el plan de ejecución “original” vs el “nuevo”.**

¿Qué diferencias encontraste?

Reducción del tiempo, gracias a los índices.

¿Ayuda o no ayuda la creación de índices?

Si, si ayuda la creación de índices.

**Genera estadísticas para optimizar y vuelve a ejecutar la consulta Q1 - Q9**

```
ANALYZE TABLE Brands COMPUTE STATISTICS;  
ANALYZE TABLE categories COMPUTE STATISTICS;  
ANALYZE TABLE customers COMPUTE STATISTICS;  
ANALYZE TABLE order_items COMPUTE STATISTICS;  
ANALYZE TABLE orders COMPUTE STATISTICS;  
ANALYZE TABLE products COMPUTE STATISTICS;  
ANALYZE TABLE staffs COMPUTE STATISTICS;  
ANALYZE TABLE stocks COMPUTE STATISTICS;  
ANALYZE TABLE stores COMPUTE STATISTICS;
```

¿Qué diferencias encontraste?

Pues los costos disminuyen por el uso de las condiciones.

¿Ayuda o no ayuda la generación de estadísticas para el RDBMS?

Si, si ayuda la generación de estadísticas.