

Último Trabajo de Bases de Datos 2 (20%) 2019-2

Punto 1 (50%)

Sea la tabla:

```
CREATE TABLE venta(  
codventa NUMBER(8) PRIMARY KEY,  
idvendedor NUMBER(8) NOT NULL,  
unidades NUMBER(8) NOT NULL CHECK (unidades > 0)  
);
```

Nota: En todas las consultas, el máximo valor que se puede presentar en la sumatoria del atributo unidades para un vendedor es 99999999.

Sean las siguientes **cinco** consultas:



Consulta 1 planteada por Kaia Gerber¹:

```
SELECT sv, COUNT(*) AS cuantos  
FROM(SELECT CASE  
    WHEN SUM(unidades) BETWEEN 0 AND 50 THEN 1  
    WHEN SUM(unidades) BETWEEN 51 AND 100 THEN 2  
    WHEN SUM(unidades) BETWEEN 101 AND 150 THEN 3  
    ELSE 4  
    END AS sv  
FROM venta  
GROUP BY idvendedor  
)  
GROUP BY sv;
```

Hasta acá la Consulta 1.

Chucky el muñeco diabólico prefiere hacer lo siguiente. Primero crea esta vista:

```
CREATE VIEW visvend AS  
SELECT DISTINCT idvendedor  
FROM venta;
```

Luego ejecuta la siguiente consulta:

¹ Gerber es una modelo, hija de la también modelo Cindy Crawford.

Consulta 2 planteada por Chucky el muñeco



diabólico²:

```
SELECT sv, COUNT(*) AS cuantos
FROM
(SELECT CASE
  WHEN sumin BETWEEN 0 AND 50 THEN 1
  WHEN sumin BETWEEN 51 AND 100 THEN 2
  WHEN sumin BETWEEN 101 AND 150 THEN 3
  ELSE 4
  END AS sv
FROM (SELECT (SELECT SUM(unidades)
              FROM venta v2
              WHERE v2.idvendedor = v1.idvendedor) AS sumin
      FROM visvend v1
      )
)
GROUP BY sv;
```

¡Qué Chucky tan malo!

Hasta acá la Consulta 2.

Daria Stavrovich hace lo siguiente: Crea una tabla auxiliar y unas vistas así:

```
CREATE TABLE rango(
sv NUMBER(1) PRIMARY KEY CHECK(sv IN (1,2,3,4)),
menor NUMBER(8) NOT NULL,
mayor NUMBER(8) NOT NULL);
```

```
INSERT INTO rango VALUES(1,0,50);
INSERT INTO rango VALUES(2,51,100);
INSERT INTO rango VALUES(3,101,150);
INSERT INTO rango VALUES(4,151,99999999);
```

```
CREATE VIEW totales AS
SELECT idvendedor, SUM(unidades) AS sumi
FROM venta
GROUP BY idvendedor;
```

² Chucky no necesita presentación.

```
CREATE VIEW clases AS
SELECT sv
FROM totales, rango
WHERE sumi BETWEEN menor AND mayor;
```

Consulta 3 planteada por Daria Stavrovich³:



```
SELECT sv, COUNT(*)
FROM clases
GROUP BY sv;
```

Hasta acá la Consulta 3.

Por su parte, Diamanda Galás crea una vista (la vista visvend la misma que usa el muñeco diabólico) y delega parte del trabajo a PL/SQL así:

```
CREATE VIEW visvend AS
SELECT DISTINCT idvendedor
FROM venta;
```

```
CREATE OR REPLACE FUNCTION clase(idv IN NUMBER)
RETURN NUMBER IS
sumi NUMBER(8);
clasi NUMBER(1);
BEGIN
SELECT SUM(unidades) INTO sumi
FROM venta
WHERE idvendedor = idv;
CASE WHEN sumi BETWEEN 0 AND 50 THEN clasi := 1;
      WHEN sumi BETWEEN 51 AND 100 THEN clasi := 2;
      WHEN sumi BETWEEN 101 AND 150 THEN clasi := 3;
      ELSE clasi := 4;
END CASE;
RETURN clasi;
END;
/
```

³ Puedes ver a Daria, e.g., acá: <https://www.youtube.com/watch?v=dwXm1J-EVKs>

Consulta 4 planteada por Diamanda Galás⁴:



```
SELECT clase(v.idvendedor) sv, COUNT(*) AS cuantos  
FROM visvend v  
GROUP BY clase(v.idvendedor);
```

Hasta acá la Consulta 4.

Finalmente, N'Dea Davenport hace lo siguiente:

```
CREATE OR REPLACE FUNCTION clasin(sumi IN NUMBER)  
RETURN NUMBER IS  
clasi NUMBER(1);  
BEGIN  
CASE WHEN sumi BETWEEN 0 AND 50 THEN clasi := 1;  
      WHEN sumi BETWEEN 51 AND 100 THEN clasi := 2;  
      WHEN sumi BETWEEN 101 AND 150 THEN clasi := 3;  
      ELSE clasi := 4;  
END CASE;  
RETURN clasi;  
END;  
/
```

Consulta 5 planteada por N'Dea



Davenport⁵:

⁴ Galás no es para alguien ¡de corazón débil! Acá con una **tenebrosa** canción que lamentablemente aplica para la época que vivimos de cuarentena: <https://www.youtube.com/watch?v=iBiFm-JI7IM>

⁵ Davenport lo intentó en solitario, pero es con The Brand New Heavies, cuando ha tenido más reconocimiento; acá con uno de sus primeros temas: <https://www.youtube.com/watch?v=FB8nIGKbpJY>

```

SELECT sv, COUNT(*) AS cuantos
FROM
(
SELECT DISTINCT idvendedor,
      clasin(SUM(unidades) OVER (PARTITION BY idvendedor))
      AS sv
FROM venta
)
GROUP BY sv;

```

Hasta acá la Consulta 5.

Haciendo caso omiso del orden en que salen los resultados (las filas), ¿son estas cinco consultas equivalentes? Es decir, ¿generan los mismos resultados si se ejecutan con los mismos datos? Justifique su respuesta explicando brevemente el funcionamiento de cada consulta.

Nota: Dos consultas son NO-Equivalentes si existe *al menos* una muestra de datos, tal que al ejecutar las consultas con esta muestra de datos, generan resultados (filas) diferentes.

Para cada una de las cinco consultas anteriores, haga pruebas con diversas muestras de datos (desarrolle un programa PL/SQL para cargar la tabla con datos aleatorios). Aumente gradualmente el tamaño de las muestras **hasta llegar a volúmenes de datos que permitan observar el comportamiento (rendimiento) de cada consulta.**

Presente el *explain plan* para cada consulta y una tabla resumen donde **incluya los tamaños de las muestras que usó** y los **costos** y tiempos obtenidos en cada caso.

Aplice también el TKPROF (está explicado en los **vídeos** que les envié) y presente para **cada** consulta, una tabla resumen donde incluya los **tamaños** de las muestras que usó y los resultados obtenidos para las siguientes tasas:

- i) LIOs (f+g) sobre filas procesadas (h).
- ii) Filas retornadas (i) sobre traídas (*fetches*) (j).
- iii) Lecturas de disco (k) sobre LIOs (f+g).

Presente conclusiones (análisis de resultados) sobre los resultados obtenidos.

Punto 2 (50%)

Repita **todo** el Punto 1 pero esta vez, inmediatamente luego de crear la tabla venta, cree este índice:

CREATE INDEX idven ON venta(idvendedor);

Ahora proceda a hacer las pruebas con las mismas consultas y con los mismos tamaños de muestras de datos del punto 1.

¿Hay diferencias en los planes y en el rendimiento con respecto a las consultas correspondientes del punto 1? Explique.

Notas:

- **No modifique** la estructura de las tablas, **ni** de las vistas, **ni** de las funciones PL/SQL **ni** de las consultas.
- Fecha de entrega: **lunes 6 de abril hasta las 4 pm**. Se debe enviar **todo el trabajo** a fjmoreno@unal.edu.co. Envíe también el programa que desarrolló para cargar la tabla venta con datos aleatorios. Envíe todo (informe y programa) en un archivo comprimido.
- No se reciben trabajos en hora posterior. A las 4:15pm de ese día, les enviaré un mensaje con la lista de los trabajos recibidos hasta las 4pm.
- No se reciben versiones “mejoradas”.
- Grupos máximo de **tres** personas.
- Los trabajos deben ser independientes entre los grupos. Trabajos copiados **así sea parcialmente** se califican con 0 (cero) para todos los integrantes de **todos** los grupos involucrados en la copia.
- Cualquier duda consultarla personalmente o por correo electrónico con el profesor. El monitor también les puede ayudar, pero NO está autorizado para cambiar las condiciones del trabajo.

Pregunta frecuente

¿Con cuántos datos se deben hacer las pruebas?

Rta. Usted decide con cuantos datos iniciar las pruebas y de a cuantos datos va aumentando gradualmente, hasta llegar a volúmenes de datos que permitan observar el comportamiento (rendimiento) de cada consulta.

En el trabajo deben quedar claro los tamaños de muestras usados.

F. Moreno

27 de marzo de 2020.