

Grupo #3

Informe de optimización de consultas

Consultas:

1) ¿Qué tipo de residuo se genera con mayor frecuencia en cada ubicación?

Antes de la optimización:

```
-- Consultas
-- 1) ¿Qué tipo de residuo se genera con mayor frecuencia en cada ubicación?
SELECT
    ubicacion,      residuo,      Total -- Seleccionar Ubicación, residuo, total
FROM (
    -- Primero hacemos un query para mostrar por cada Ubicación un ranking con el tipo de residuo
    SELECT
        CONCAT(u.nombre, ' ', u.ciudad) AS Ubicacion,
        tr.nombre AS residuo,
        SUM(rd.cantidad_kg) AS Total,
        ROW_NUMBER() OVER(PARTITION BY u.id_ubicacion ORDER BY SUM(rd.cantidad_kg) DESC) AS ranking
    FROM
        recoleccion AS r
    INNER JOIN recoleccion_detalle AS rd ON r.id_recoleccion = rd.id_recoleccion
    INNER JOIN ubicacion AS u ON u.id_ubicacion = r.id_ubicacion
    INNER JOIN tipo_residuo AS tr ON tr.id_tipo_residuo = rd.id_tipo_residuo
    GROUP BY
        u.id_ubicacion, tr.nombre
) AS T1
WHERE
    T1.ranking = 1 -- Filtramos solo el primer ranking de cada Ubicación
ORDER BY
    T1.Total DESC;
```

Result Grid Filter Rows: Export: Wrap Cell Content: A												
	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
	1	PRIMARY	<derived2>	NULL	ref	<auto_key0>	<auto_key0>	8	const	2	100.00	Using filesort
	2	DERIVED	rd	NULL	ALL	uq_detalle_id_tipo_residuo	NULL	NULL	NULL	23	100.00	Using temporary; Using filesort
	2	DERIVED	tr	NULL	eq_ref	PRIMARY	PRIMARY	4	proyecto.rd.id_tipo_residuo	1	100.00	NULL
	2	DERIVED	r	NULL	eq_ref	PRIMARY,uq_ubic_fecha	PRIMARY	4	proyecto.rd.id_recoleccion	1	100.00	NULL
	2	DERIVED	u	NULL	eq_ref	PRIMARY	PRIMARY	4	proyecto.r.id_ubicacion	1	100.00	NULL

2) ¿Qué días de la semana tienen mayor volumen de recolección?

Antes de la optimización:

```
149
150 -- 2) ¿Qué días de la semana tienen mayor volumen de recolección?
151 • SELECT
152     DATE_FORMAT(r.fecha, '%W') AS Dia,
153     SUM(rd.cantidad_kg) AS TotalRecolectado
154 FROM recoleccion AS r
155 INNER JOIN recoleccion_detalle AS rd
156     ON r.id_recoleccion = rd.id_recoleccion
157 GROUP BY Dia
158 ORDER BY TotalRecolectado DESC;
159
```

Result Grid										
Filter Rows:										
Export: Wrap Cell Content: I A										
	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	Extra
▶	1	SIMPLE	r	NULL	index	PRIMARY,uq_ubic_fecha	uq_ubic_fecha	7	NULL	Using index; Using temporary; Using filesort
	1	SIMPLE	rd	NULL	ref	uq_detalle,IDX_RD_FKId_recoleccion,idx_rd_re...	idx_rd_recoleccion_cantidad	4	proyecto.r.id_recoleccion	Using index

```
152     DATE_FORMAT(r.fecha, '%W') AS Dia,
153     SUM(rd.cantidad_kg) AS TotalRecolectado
154 FROM recoleccion AS r
155 INNER JOIN recoleccion_detalle AS rd
156     ON r.id_recoleccion = rd.id_recoleccion
157 GROUP BY Dia
158 ORDER BY TotalRecolectado DESC;
159
160 -- Optimizacion de datos con index
161 • CREATE INDEX ix_recoleccion_fecha_recol
162     ON recoleccion (fecha, id_recoleccion);
163
164 • CREATE INDEX ix_detalle_recol_cant
165     ON recoleccion_detalle (id_recoleccion, cantidad_kg);
```

```
167 --
168 • EXPLAIN
169 SELECT
170     DATE_FORMAT(r.fecha, '%W') AS Dia,
171     SUM(rd.cantidad_kg) AS TotalRecolectado
172 FROM recoleccion AS r
173 JOIN recoleccion_detalle AS rd
174     ON r.id_recoleccion = rd.id_recoleccion
175 GROUP BY Dia
176 ORDER BY TotalRecolectado DESC;
```

Result Grid										
Filter Rows:										
Export: Wrap Cell Content: I A										
	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	Extra
▶	1	SIMPLE	r	INDEX	index	PRIMARY,uq_ubic_fecha,ix_recolect...	uq_ubic_fecha	7	INDEX	Using index; Using temporary; Using ...
	1	SIMPLE	rd	INDEX	ref	uq_detalle,ix_detalle_recol_tpo,ix_de...	uq_detalle	4	proyecto.r.id_recoleccion	Using index

Antes de la optimización:

Antes de la optimización:

SELECT

```
ROUND(
    SUM(CASE WHEN Clasificacion = 'Reciclable' THEN Cantidad ELSE 0 END)
    / SUM(Cantidad) * 100, 2) AS Tasa_Eficiencia -- round
```

```
(
SELECT CONCAT(u.nombre, ' ', u.ciudad) AS Ubicacion,
       tr.clasificacion AS Clasificacion,
       SUM(rd.cantidad_kg) AS Cantidad
FROM
    recoleccion AS r
    INNER JOIN recoleccion_detalle AS rd ON r.id_recoleccion = rd.id_recoleccion
    INNER JOIN ubicacion AS u ON u.id_ubicacion = r.id_ubicacion
    INNER JOIN tipo_residuo AS tr ON tr.id_tipo_residuo = rd.id_tipo_residuo
GROUP BY
    Ubicacion,
    Clasificacion
)
```

GROUP BY

ORDER BY

Tasa_Eficiencia DESC;

Después de la optimización agregando INDEX:

```
236 • CREATE INDEX IX_recoleccion_detalle_completa
237 ON recoleccion_detalle (id_recoleccion, id_tipo_residuo, cantidad_kg);
238
```

Result Grid		Filter Rows:		Export:		Wrap Cell Content:						
	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	<derived2>	HOLE	ALL	HOLE		HOLE	HOLE	HOLE	22	100.00	Using temporary; Using filesort
2	DERIVED	tr	HOLE	ALL	PRIMARY		HOLE	HOLE	HOLE	6	100.00	Using temporary
2	DERIVED	rd	HOLE	ref	uq_detalle_id_tpo_residuo,IDX_RD_FKId_recole...	id_tpo_residuo	4		projecto.tr.id_tpo_residuo	3	100.00	HOLE
2	DERIVED	r	HOLE	eq_ref	PRIMARY,uq_ubic_fecha		PRIMARY	4	projecto.rd.id_recoleccion	1	100.00	HOLE
2	DERIVED	u	HOLE	eq_ref	PRIMARY		PRIMARY	4	projecto.r.id_ubicacion	1	100.00	HOLE