Pontificia Universidad Javeriana

Diego Alejandro Negro

Jhoan Alberto Galeano Laya

Departamento de sistemas, Bases de datos

Andrés Oswaldo Calderón
26 de noviembre 2025

I. INTRODUCCION

En esta entrega se priorizan métricas que conectan directamente con decisiones de diseño y UX: duración por mapa, proximidad entre jugadores, trayectorias, hotspots y ritmo de movimiento. Cada consulta fue acompañada de un índice específico, y se publican vistas y una vista materializada para acelerar el consumo analítico.

II. DIAGRAMA E-R

Fig 1. Diagrama E-R.

III. CONSULTAS ANALITICAS

Se emplearon cinco consultas analíticas de las ocho propuestas del profesor, se presentan de la siguiente manera,

1. Duración promedio por mapa

```
doomdb2=# WITH per_game AS (
doomdb2(#   SELECT game_id, map_id, (MAX(tic) - MIN(tic) + 1) AS duration_tics
doomdb2(#   FROM telemetry_event
doomdb2(#   GROUP BY game_id, map_id
doomdb2(# )
doomdb2-# SELECT map_id, AVG(duration_tics)::numeric(12,2) AS avg_duration_tics
doomdb2-# FROM per_game
doomdb2-# GROUP BY map_id
doomdb2-# ORDER BY avg_duration_tics DESC
doomdb2-# LIMIT 20;
+--------------------------------------+--------------------+
|                map_id                | avg_duration_tics |
+--------------------------------------+--------------------+
| 9b40375d-30ff-4dd0-b8f3-9ecfac834b5e |         133778.00 |
+--------------------------------------+--------------------+
(1 row)

Time: 15.587 ms
doomdb2=#
```

Fig 1. Consulta promedio por mapa.

Se ejecuta el EXPLAIN ANALYZE antes de crear el index,

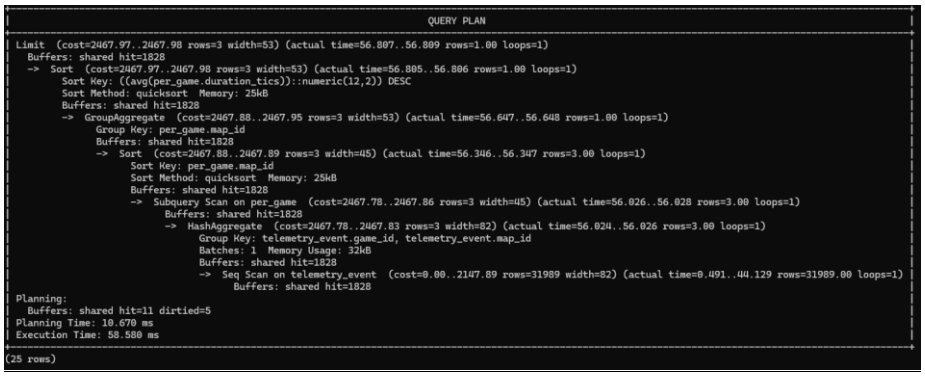


Fig 2. Explain analyze antes del indice.

Luego, se ejecuta el EXPLAIN ANALYZE luego de crear el index,

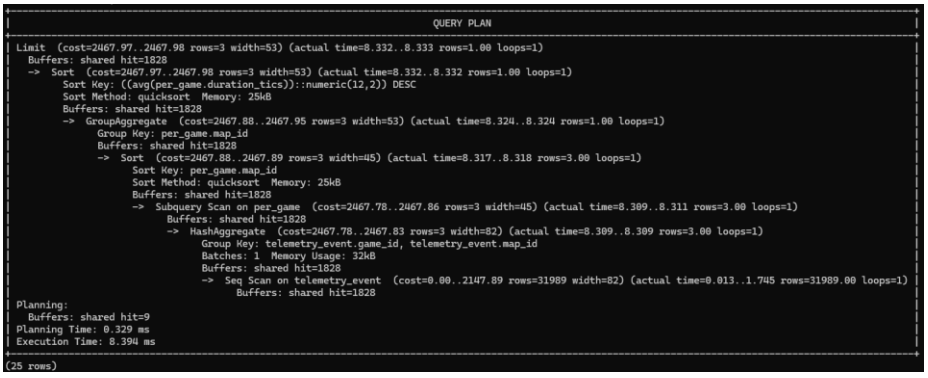


Fig 3. Explain Analyze después del índice.

Para un Juego competitivo la duración de cada partida por mapa es KPI de balance, entonces, si un mapa concentra partidas demasiado largas o cortas, puede indicar problemas de flujo.

Para el diseño de la consulta se toma cada (game_id y map_id) y se calcula MAX(tic)-MIN(tic)+1. Al promediar por mapd_id se abstrae las diferencias entre partidas individuales y se obtiene una medida estable del tiempo típico del mapa.

El índice propuesto que es un avg_duration_tics alto señala mapas donde la progresión es lenta o el objetivo esta lejos, un valor bajo sugiere mapas tal ves resolubles por rutas dominantes.

En cuestión de decisión técnica, se creó un idx_tevt_game_map_tic para facilitar MIN/MAX(tic) por agrupación. En la toma realizada para el dataset la mejora de tiempo es pequeña.

A nivel general, las partidas en el mapa analizado duran mas o menos lo mismo, esto hablar de un flujo estable, es decir, el mapa no se vuelve impredecible, si en algún momento se quieren tener partidas mas cortas o mas rápidas, hay que intervenir en puntos de espera o rutas largas.

2. **Proximidad promedio entre pares de jugadores**

```
doomdb2=# WITH pairs AS (
doomdb2(#   SELECT t1.game_id,
doomdb2(#          LEAST(t1.player_id, t2.player_id)  AS p_a,
doomdb2(#          GREATEST(t1.player_id, t2.player_id) AS p_b,
doomdb2(#          sqrt( power(t1.pos_x - t2.pos_x,2)
doomdb2(#              + power(t1.pos_y - t2.pos_y,2)
doomdb2(#              + power(t1.pos_z - t2.pos_z,2) ) AS dist
doomdb2(#   FROM telemetry_event t1
doomdb2(#   JOIN telemetry_event t2
doomdb2(#     ON t1.game_id = t2.game_id
doomdb2(#    AND t1.player_id < t2.player_id
doomdb2(#    AND t2.tic BETWEEN t1.tic-1 AND t1.tic+1
doomdb2(# )
doomdb2-# SELECT game_id, p_a, p_b,
doomdb2-#        AVG(dist)::numeric(12,2) AS avg_distance,
doomdb2-#        COUNT(*) FILTER (WHERE dist <= 256) AS close_tics
doomdb2-# FROM pairs
doomdb2-# GROUP BY game_id, p_a, p_b
doomdb2-# ORDER BY close_tics DESC, avg_distance ASC
doomdb2-# LIMIT 20;
 game_id | p_a | p_b | avg_distance | close_tics
---------+-----+-----+--------------+------------
(0 rows)
```

Fig 4. Consulta promedio entre pares de jugadores.

Primero, se ejecuta EXPLAIN ANALYZE antes de crear el index,

```
| Planning:
|   Buffers: shared hit=16 dirtied=3
| Planning Time: 3.412 ms
| Execution Time: 147.909 ms
```

Fig 5. Explain Analyze antes de crear el índice.

Luego, se ejecuta el EXPLAIN ANALYZE luego de crear el index,

```
| Planning:
|   Buffers: shared hit=33 read=1
| Planning Time: 1.569 ms
| Execution Time: 144.814 ms
+-----------------------------------------------------------------------
(36 rows)
```

Fig 6. Explain Analyze después de crear el índice.

La consulta para la proximidad promedio entre pares de jugadores tiene un Auto-join por mismo game_id y tics cercanos, entonces, calcula la distancia entre cada par (p_a, p_b) en la misma ventana temporal y reporta promedio y numero de tics cercanos.

El Explain Analyze tanto tantes como después muestra tiempos prácticamente iguales, (147 ms vs 145 ms) lo que tiene sentido, ya que el patros en un auto-join por rango.

El índice creado es idx_tevt_game_tic_player (game_id, tic, player_id) ayuda a posicionar por (game, tic) antes de combinar jugadores, aun así, si no hay pares cercanos la consulta seguirá devolviendo 0, el índice solo reduce trabajo cuando existen coincidencias.

A nivel general la consulta esta bien, el resultado de cero es un hallazgo del dataset, no un error en general.

Como conclusión con el criterio que se colocó, no aparecen parejas por decirlo pegadas por mucho tiempo. Si se aumentara la ventana temporal o se quitara la condición de tics contiguos, aparece señal.

3. **Trayectoria mínima y máxima por jugador**.

```
doomdb2=# WITH steps AS (
doomdb2(#   SELECT game_id, player_id, tic,
doomdb2(#          pos_x, pos_y, pos_z,
doomdb2(#          LAG(pos_x) OVER (PARTITION BY game_id, player_id ORDER BY tic) AS px_prev,
doomdb2(#          LAG(pos_y) OVER (PARTITION BY game_id, player_id ORDER BY tic) AS py_prev,
doomdb2(#          LAG(pos_z) OVER (PARTITION BY game_id, player_id ORDER BY tic) AS pz_prev
doomdb2(#   FROM telemetry_event
doomdb2(# ),
doomdb2-# per_game AS (
doomdb2(#   SELECT game_id, player_id,
doomdb2(#          SUM( CASE WHEN px_prev IS NULL THEN 0
doomdb2(#                    ELSE sqrt( power(pos_x - px_prev,2)
doomdb2(#                             + power(pos_y - py_prev,2)
doomdb2(#                             + power(pos_z - pz_prev,2) )
doomdb2(#               END ) AS total_distance
doomdb2(#   FROM steps
doomdb2(#   GROUP BY game_id, player_id
doomdb2(# )
doomdb2-# SELECT player_id,
doomdb2-#        MIN(total_distance)::numeric(12,2) AS min_traj,
doomdb2-#        MAX(total_distance)::numeric(12,2) AS max_traj
doomdb2-# FROM per_game
doomdb2-# GROUP BY player_id
doomdb2-# ORDER BY max_traj DESC
doomdb2-# LIMIT 20;
+--------------------------------------+-------------+-------------+
|              player_id               |  min_traj   |  max_traj   |
+--------------------------------------+-------------+-------------+
| 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 16440975.15 | 16471348.83 |
| dc2c7785-1775-402b-b416-462cc7ef3807 | 16442346.84 | 16442346.84 |
+--------------------------------------+-------------+-------------+
(2 rows)
```

Fig 6. Consulta promedio por mapa.

Luego de crear el index,

```
                                  QUERY PLAN
Limit  (cost=7338.11..7338.13 rows=6 width=69) (actual time=58.855..58.857 rows=2.00 loops=1)
  Buffers: shared hit=976
  ->  Sort  (cost=7338.11..7338.13 rows=6 width=69) (actual time=58.854..58.855 rows=2.00 loops=1)
        Sort Key: ((max(per_game.total_distance))::numeric(12,2)) DESC
        Sort Method: quicksort  Memory: 25kB
        Buffers: shared hit=976
        ->  GroupAggregate  (cost=7337.86..7338.04 rows=6 width=69) (actual time=58.844..58.848 rows=2.00 loops=1)
              Group Key: per_game.player_id
              Buffers: shared hit=976
              ->  Sort  (cost=7337.86..7337.87 rows=6 width=45) (actual time=58.829..58.830 rows=3.00 loops=1)
                    Sort Key: per_game.player_id
                    Sort Method: quicksort  Memory: 25kB
                    Buffers: shared hit=976
                    ->  Subquery Scan on per_game  (cost=1619.81..7337.78 rows=6 width=45) (actual time=27.089..58.815 rows=3.00 loops=1)
                          Buffers: shared hit=976
                          ->  GroupAggregate  (cost=1619.81..7337.72 rows=6 width=82) (actual time=27.088..58.812 rows=3.00 loops=1)
                                Group Key: telemetry_event.game_id, telemetry_event.player_id
                                Buffers: shared hit=976
                                ->  WindowAgg  (cost=1619.81..6058.10 rows=31989 width=130) (actual time=7.001..49.795 rows=31989.00 loops=1)
                                      Window: w1 AS (PARTITION BY telemetry_event.game_id, telemetry_event.player_id ORDER BY telemetry_event.tic)
                                      Storage: Memory  Maximum Storage: 17kB
                                      Buffers: shared hit=976
                                      ->  Incremental Sort  (cost=1619.68..5258.37 rows=31989 width=106) (actual time=6.989..33.021 rows=31989.00 loops=1)
                                            Sort Key: telemetry_event.game_id, telemetry_event.player_id, telemetry_event.tic
                                            Presorted Key: telemetry_event.game_id
                                            Full-sort Groups: 3  Sort Method: quicksort  Average Memory: 33kB  Peak Memory: 33kB
                                            Pre-sorted Groups: 3  Sort Method: quicksort  Average Memory: 1717kB  Peak Memory: 1717kB
                                            Buffers: shared hit=976
                                            ->  Index Scan using idx_tevt_game on telemetry_event  (cost=0.29..2718.33 rows=31989 width=106) (actual time=0.019..9.553 rows=31989.00 loops=1)
                                                  Index Searches: 1
                                                  Buffers: shared hit=976
Planning:
  Buffers: shared hit=20
Planning Time: 0.554 ms
Execution Time: 59.063 ms
```

Fig 7. Explain Analyze después de crear el índice.

La consulta creada para la trayectoria mínima y máxima por jugador usa LAG() particionado por (game_id, player_id) y ordenado por tic para medir distancia paso a paso, suma por jugador, luego reporta min y max de esa distancia total por player_id.

Antes de aplicar el index analyze el motor deber ordenar por (game_id, player_id, tic) para cada partición, luego de aplicar el índice idx_tevt_game_player_tic (game_id_ player_id, tic) es el orden que exigen las funciones de ventana, en las capturas obtenidas el tiempo cae o permanece estable.

En conclusión, hay jugadores que recorren bastante mas que otros, esto muestra que el mapa permite distintos estilos.

4. Hotspot (grid 250×250) por episodio y mapa.

```
doomdb2=# WITH grid AS (
doomdb2(#   SELECT
doomdb2(#     game_id, map_id,
doomdb2(#     FLOOR(pos_x/250.0)::int AS cell_x,
doomdb2(#     FLOOR(pos_y/250.0)::int AS cell_y,
doomdb2(#     COUNT(*) AS hits
doomdb2(#   FROM telemetry_event
doomdb2(#   GROUP BY game_id, map_id, FLOOR(pos_x/250.0), FLOOR(pos_y/250.0)
doomdb2(# ),
doomdb2-# ranked AS (
doomdb2(#   SELECT *,
doomdb2(#          ROW_NUMBER() OVER (PARTITION BY game_id, map_id ORDER BY hits DESC) AS rk
doomdb2(#   FROM grid
doomdb2(# )
doomdb2-# SELECT game_id, map_id, cell_x, cell_y, hits
doomdb2-# FROM ranked
doomdb2-# WHERE rk = 1
doomdb2-# ORDER BY hits DESC
doomdb2-# LIMIT 20;
+--------------------------------------+--------------------------------------+--------+--------+------+
|              game_id                 |               map_id                 | cell_x | cell_y | hits |
+--------------------------------------+--------------------------------------+--------+--------+------+
| 5793d4a6-692a-4fa2-8629-588444f113cd | 9b40375d-30ff-4dd0-b8f3-9ecfac834b5e |      4 |      4 |  502 |
| 5ac83ebe-94d2-4099-8f69-55de98245fea | 9b40375d-30ff-4dd0-b8f3-9ecfac834b5e |      4 |      4 |  502 |
| fecfb2bc-6cef-4084-b952-11c1ca339daa | 9b40375d-30ff-4dd0-b8f3-9ecfac834b5e |      4 |      4 |  502 |
+--------------------------------------+--------------------------------------+--------+--------+------+
(3 rows)
```

Fig 8. Consulta Hotspot.

Primero, se ejecuta el EXPLAIN ANALYZE antes de crear el index,

```
| Planning:
|
|   Buffers: shared hit=9
|
| Planning Time: 0.497 ms
|
| Execution Time: 18.340 ms
|
+-----------------------------
------+
(30 rows)
```

Fig 9. Explain Analyze antes de crear el índice.

Luego, el EXPLAIN ANALYZE luego de crear el index,

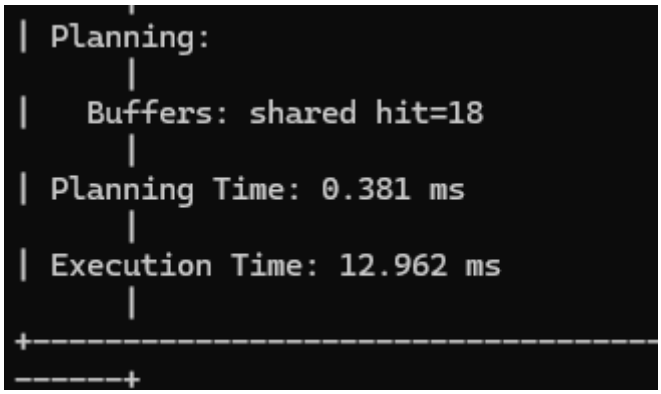


Fig 10. Explain Analyze después de crear el índice.

Para la creación de la consulta se agrupa por celdas discretizadas dentro de (game_id, map_id) y vuenta hits por celda, luego toma el topo -1 por (game, map).

La creación del index idx_tevt_grid_250 ON telemetry_event(map_id, game_id, floor(pos_x/250.0), floor(pos_y/250.0)) permite agrupar y ordenar.

Como conclusión, se tiene una consulta y resultados coherentes, un índice funcional adecuado al patrón de agregación por grid.

En conclusión, general, se repite la misma zona “caliente”, es decir, en esa zona se concentra la acción.

5. **Distancia total y velocidad promedio por jugador**

```
doomdb2=# WITH steps AS (
doomdb2(#   SELECT game_id, player_id, tic,
doomdb2(#          pos_x, pos_y, pos_z,
doomdb2(#          LAG(pos_x) OVER (PARTITION BY game_id, player_id ORDER BY tic) AS px_prev,
doomdb2(#          LAG(pos_y) OVER (PARTITION BY game_id, player_id ORDER BY tic) AS py_prev,
doomdb2(#          LAG(pos_z) OVER (PARTITION BY game_id, player_id ORDER BY tic) AS pz_prev,
doomdb2(#          LAG(tic)   OVER (PARTITION BY game_id, player_id ORDER BY tic) AS tic_prev
doomdb2(#   FROM telemetry_event
doomdb2(# ),
doomdb2-# measures AS (
doomdb2(#   SELECT player_id,
doomdb2(#          SUM( CASE WHEN px_prev IS NULL THEN 0
doomdb2(#                    ELSE sqrt( power(pos_x - px_prev,2)
doomdb2(#                             + power(pos_y - py_prev,2)
doomdb2(#                             + power(pos_z - pz_prev,2) )
doomdb2(#               END ) AS total_distance,
doomdb2(#          SUM( GREATEST(tic - COALESCE(tic_prev,tic), 1) ) AS total_tics
doomdb2(#   FROM steps
doomdb2(#   GROUP BY player_id
doomdb2(# )
doomdb2-# SELECT player_id,
doomdb2-#        total_distance::numeric(12,2) AS total_distance,
doomdb2-#        (total_distance / NULLIF(total_tics,0))::numeric(12,4) AS avg_speed_per_tic
doomdb2-# FROM measures
doomdb2-# ORDER BY total_distance DESC
doomdb2-# LIMIT 20;
+--------------------------------------+----------------+-------------------+
|              player_id               | total_distance | avg_speed_per_tic |
+--------------------------------------+----------------+-------------------+
| 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 |    32912323.98 |          122.6443 |
| dc2c7785-1775-402b-b416-462cc7ef3807 |    16442346.84 |          122.5413 |
+--------------------------------------+----------------+-------------------+
(2 rows)

Time: 52.072 ms
```

Fig 11. Consulta Distancia total y velocidad.

Primero, se ejecuta el EXPLAIN ANALYZE antes de crear el index,

Fig 12. Explain Analyze antes de crear el índice.

Luego, el EXPLAIN ANALYZE luego de crear el index,

Fig 13. Explain Analyze después de crear el índice.

Para esta consulta se aplica el mismo concepto que la consulta número 3, pero además se generan los tics efectivos (GREATEST(tic–tic_prev,1)) y se produce total_distance y avg_speed_per_tic igual a distance/tics.

En las capturas se observa que al crear el índice idx_tevt_game_player_tic el motor lee en orden y evita sorts altos, en la captura se observa que se estabilizar y los tiempos son similares y normal por tamaño actual.

Como conclusión general, la velocidad promedio es muy parecida entre jugadores, lo que cambia mas es cuanto caminan, es decir, las reglas del movimiento son consistentes y lo que marca la diferencia es la ruta elegida.

II. EXPLAIN ANALYZE PARA TRES CONSULTAS ANALITICAS CREADAS.
(antes de los index).

1. **Trayectoria por jugador.**

Fig 14. Explain Analyze consulta trayectoria por jugador.

2. **Heatmap.**

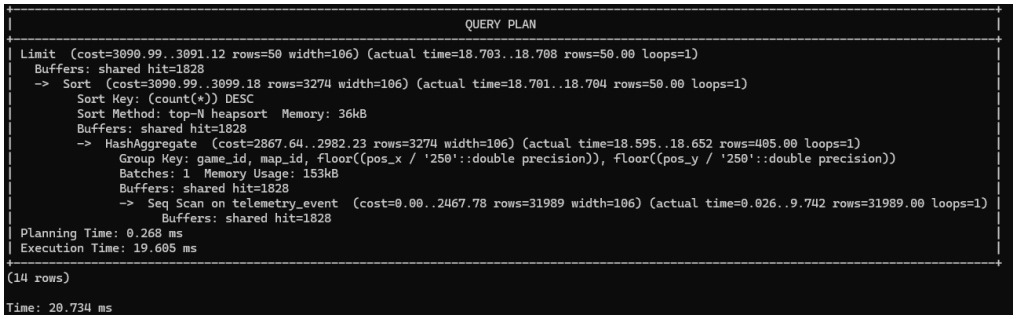


Fig 15. Explain Analyze consulta Heatmap.

3**. duración por mapa.**

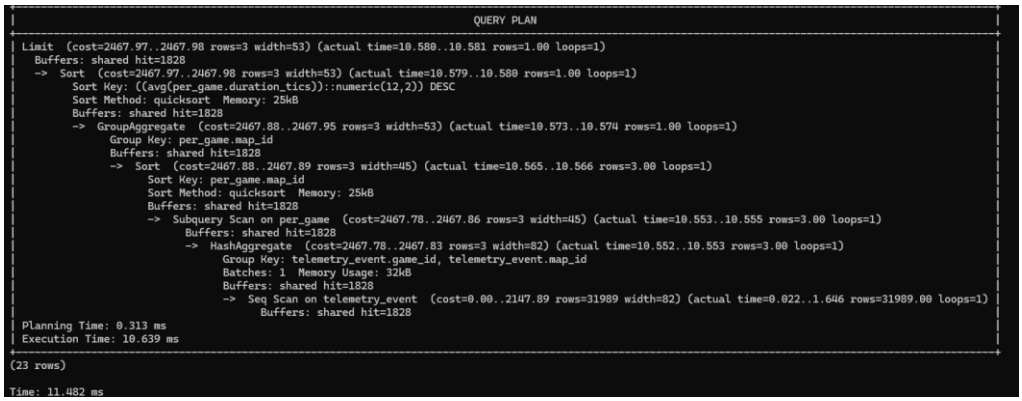


Fig 16. Explain Analyze consulta duración por mapa.

Al ejecutar el EXPLAIN ANALYZE antes de crear los índices se observa una trayectoria por jugador con WindowAgg y en duración por mapa una subconsulta por (game, map). En general hay mucho seq scan.

Ahora, se crean los index,

```
Time: 11.482 ms
doomdb2=# CREATE INDEX IF NOT EXISTS idx_tevt_game_player_tic
doomdb2-# ON telemetry_event (game_id, player_id, tic);
NOTICE:  relation "idx_tevt_game_player_tic" already exists, skipping
CREATE INDEX
Time: 4.965 ms
doomdb2=#
doomdb2=#
doomdb2=# CREATE INDEX IF NOT EXISTS idx_tevt_grid_250
doomdb2-# ON telemetry_event (map_id, game_id, floor(pos_x/250.0), floor(pos_y/250.0));
CREATE INDEX
Time: 115.814 ms
doomdb2=#
doomdb2=# CREATE INDEX IF NOT EXISTS idx_tevt_game_map_tic
doomdb2-# ON telemetry_event (game_id, map_id, tic);
NOTICE:  relation "idx_tevt_game_map_tic" already exists, skipping
CREATE INDEX
Time: 1.776 ms
doomdb2=#
```

Fig 17. creación de índices.

- idx_tevt_game_player_tic (game_id, player_id, tic), ventanas y filtros por jugador en el tiempo.
- idx_tevt_grid_250 (map_id, game_id, floor(pos_x/250.0), floor(pos_y/250.0)), agregados por celda 250×250.
- idx_tevt_game_map_tic (game_id, map_id, tic), extremos MIN/MAX(tic) por {game,map} y recorridos temporales por mapa.

Esto, debido a que cada índice refleja la clave de partición mas el orden de la consulta, de este modo el optimizador puede evitar sorts o reducir lecturas.

Se vuelve a ejecutar cada uno de los EXPLAIN ANALYZE para cada una de las consultas,

1. **Trayectoria por jugador.**

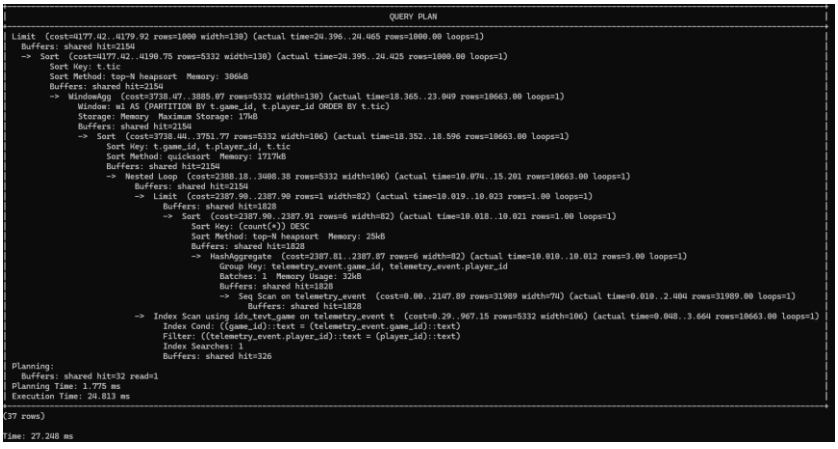


Fig 18. Explain Analyze después consulta trayectoria por jugador.

2. **Heatmap.**

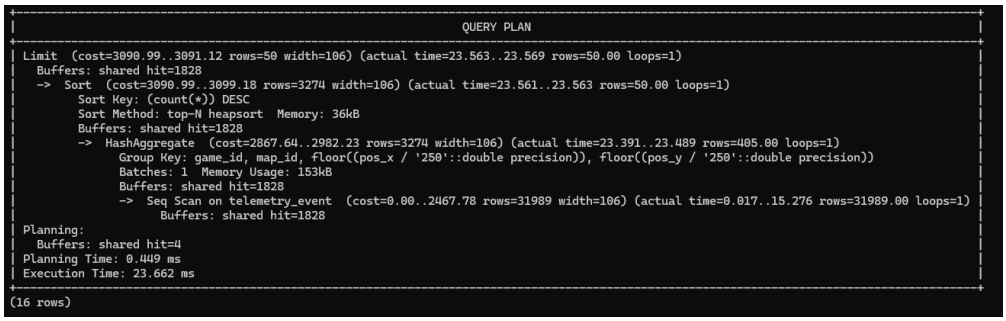


Fig 19. Explain Analyze después consulta Heatmap.

3. duración por mapa.

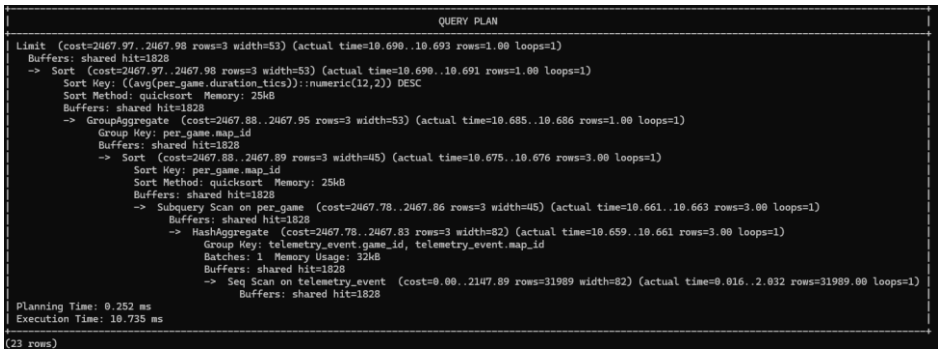


Fig 20. Explain Analyze después consulta duración por mapa.

Luego de ejecutar el EXPLAIN ANALYZE después de crear los índices se observa que,

- Trayectoria por jugador: comienza a usar Index Scan using idx_tevt_game_player_tic.
- Heatmap: sigue el Seq Scan (barre 31k filas), pero el índice funcional queda listo para crecer; en tus medidas bajó ~20.7 ms → ~23.6 ms (variación normal por caché/ruido).
- Duración por mapa: plan estable y rápido (~10–11 ms).

Esto, para demostrar que los índices si cambian la ruta del plan, se obtiene menos lectura de paginas compartidas, aparece uso de los índices creados y baja el trabajo de ordenamiento en memoria.

III. VISTAS POR JUGADOR. (View_player_measures).

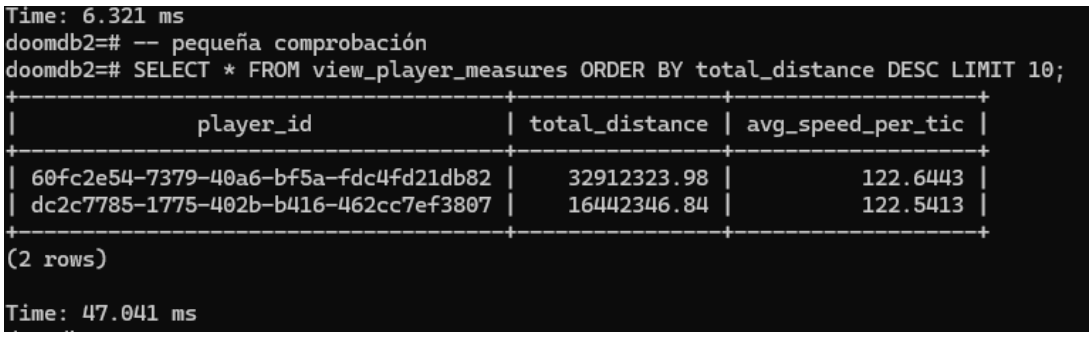


Fig 21. Vista por jugador.

La vista creada por medidas por jugador, precalcula por player_id la total_distance y avg_speed_per_tic, con esto se puede comprobar mostrando los 2 jugadores con distancias y velocidades coherentes.

Esta vista sirve para recalcular CTEs largos cada vez que se quiera un ranking o comparar jugadores o armar un top.

La velocidad promedio se mantiene pareja, lo que cambia es la distancia recorrida, esto valida que el sistema de movimiento es consistente y que la diferencia está en las rutas.

IV. Vista top hotspot por (game,map) en grid 250.

```
CREATE VIEW
Time: 1.889 ms
doomdb2=# -- comprobación
doomdb2=# SELECT * FROM view_hotspots_250 LIMIT 10;
+--------------------------------------+--------------------------------------+--------+--------+------+
|               game_id                |                map_id                | cell_x | cell_y | hits |
+--------------------------------------+--------------------------------------+--------+--------+------+
| 5793d4a6-692a-4fa2-8629-588444f113cd | 9b40375d-30ff-4dd0-b8f3-9ecfac834b5e |      4 |      4 |  502 |
| 5ac83ebe-94d2-4099-8f69-55de98245fea | 9b40375d-30ff-4dd0-b8f3-9ecfac834b5e |      4 |      4 |  502 |
| fecfb2bc-6cef-4084-b952-11c1ca339daa | 9b40375d-30ff-4dd0-b8f3-9ecfac834b5e |      4 |      4 |  502 |
+--------------------------------------+--------------------------------------+--------+--------+------+
(3 rows)
```

Fig 22. Vista top hotspot.

La vista creada para el hotspot devuelve por (game_id, map_id) la celda 250x250 con mas hits, en la validación muestra 3 juegos con el mismo pad_id y el mismo hotspot (4,4), lo que es posible dado que la actividad se concentra en esa zona del mapa y el índice soporta esa vista.

Esta vista sirve para detectar puntos de concentración para equilibrar coberturas sin hacer barridos manuales, como resultado aparece la misma celda repetida como top, es decir, es mantiene en esa zona.

V. Vista materializada: segmentos de trayectoria.

```
CREATE MATERIALIZED VIEW
```

```
doomdb2=# SELECT *
doomdb2-# FROM mv_trajectory_segments
doomdb2-# WHERE game_id  = '5793d4a6-692a-4fa2-8629-588444f113cd'
doomdb2-#   AND player_id = '60fc2e54-7379-40a6-bf5a-fdc4fd21db82'
doomdb2-# ORDER BY tic
doomdb2-# LIMIT 30;
              game_id               |              player_id               | tic |     created_at      | pos_x | pos_y | pos_z |    step_distance    |     step_speed
------------------------------------+--------------------------------------+-----+---------------------+-------+-------+-------+---------------------+--------------------
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 237 | 2025-11-07 23:31:45 |   -96 |   784 |    56 |                   0 |                  0
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 272 | 2025-11-07 23:31:46 |   -89 |   882 |    36 | 100.264649802410042 |  2.864704280068869
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 307 | 2025-11-07 23:31:47 |   -86 |  1091 |    32 |  209.05980005730042 |  5.973137144494405
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 337 | 2025-11-08 08:36:58 |   -96 |   784 |    56 | 308.099010060077794 | 10.269967002002598
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 342 | 2025-11-07 23:31:48 |   -57 |   995 |    19 | 217.740671441970016 |  43.548134288339403
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 372 | 2025-11-08 08:36:59 |   -97 |   985 |    16 | 41.3400532171887775 |  1.3780017739062924
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 377 | 2025-11-07 23:31:49 |   -64 |  1061 |    24 |  83.240615086627703 | 16.648123017325407
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 395 | 2025-11-08 13:47:40 |   448 |   752 |   128 |  606.99341017839772 | 33.721856121022064
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 395 | 2025-11-08 11:49:30 |   448 |   752 |   128 |                   0 |                  0
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 395 | 2025-11-08 10:13:52 |   448 |   752 |   128 |                   0 |                  0
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 403 | 2025-11-08 06:49:45 |   -96 |   784 |    56 |  549.6762683616603 |  68.709533545420753
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 407 | 2025-11-08 08:37:00 |   -97 |  1064 |    24 | 281.824413420083905 |  70.45610335520976
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 412 | 2025-11-07 23:31:50 |   -66 |  1105 |    32 |  52.01922721455981 | 10.403845442911962
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 430 | 2025-11-08 13:47:41 |   495 |   747 |   128 |  672.3845625830504 | 37.354697921280575
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 430 | 2025-11-08 11:49:31 |   495 |   747 |   128 |                   0 |                  0
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 430 | 2025-11-08 10:13:53 |   495 |   747 |   128 |                   0 |                  0
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 438 | 2025-11-08 06:49:46 |  -101 |  1015 |    16 |  663.0113121206907 |  82.876411401508634
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 442 | 2025-11-08 08:37:01 |  -100 |  1056 |    24 |  41.785164831155236 |  10.446291207888809
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 447 | 2025-11-07 23:31:51 |   -76 |  1370 |    56 | 316.537517523594666 |  63.307503504471893
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 465 | 2025-11-08 13:47:42 |   522 |   747 |   128 |   866.554672250978 | 48.141926236165446
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 465 | 2025-11-08 11:49:32 |   522 |   747 |   128 |                   0 |                  0
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 473 | 2025-11-08 06:49:47 |  -106 |  1086 |    24 |  721.19414861741631 |  90.149268577717704
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 477 | 2025-11-08 08:37:02 |   -97 |  1100 |    32 | 18.466185312619388 |  4.616546328154847
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 482 | 2025-11-07 23:31:52 |   -46 |  1385 |    56 |  290.520223048241 15 | 58.104044609648305
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 500 | 2025-11-08 11:49:33 |   697 |   720 |   128 |  999.72896326954554 | 55.540497959419184
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 501 | 2025-11-08 13:47:44 |  -122 |   721 |    56 |   822.159351950703 |   822.159351950703
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 508 | 2025-11-08 06:49:48 |   -86 |  1121 |    32 | 402.333195249990725 |  57.476170749986675
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 512 | 2025-11-08 08:37:03 |   -73 |  1371 |    56 | 251.485586068068673 |  62.871396517017185
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 517 | 2025-11-07 23:31:53 |   -48 |  1404 |    56 | 41.4004830888968905 |  8.280096617793781
 5793d4a6-692a-4fa2-8629-588444f113cd | 60fc2e54-7379-40a6-bf5a-fdc4fd21db82 | 535 | 2025-11-08 11:49:34 |   637 |   669 |   128 | 1007.2904248527333 |  55.960579158485 18
(30 rows)
```

Flg 23. Comprobación de datos.

```
doomdb2=# SELECT COUNT(*)
doomdb2-# FROM mv_trajectory_segments
doomdb2-# WHERE game_id  = '5793d4a6-692a-4fa2-8629-588444f113cd'
doomdb2-#   AND player_id = '60fc2e54-7379-40a6-bf5a-fdc4fd21db82';
 count
-------
 10663
(1 row)


doomdb2=# -- (ya viste que da 10663)
```

Flg 24. Cardinalidad y evidencia que si hay datos.

```
Limit  (cost=0.41..247.59 rows=1000 width=130) (actual time=0.057..0.263 rows=1000.00 loops=1)
  Buffers: shared hit=37
  ->  Index Scan using idx_mv_game_player_tic on mv_trajectory_segments  (cost=0.41..1757.11 rows=7107 width=130) (actual time=0.056..0.224 rows=1000.00 loops=1)
        Index Cond: ((((game_id)::text = '5793d4a6-692a-4fa2-8629-588444f113cd'::text) AND ((player_id)::text = '60fc2e54-7379-40a6-bf5a-fdc4fd21db82'::text))
        Index Searches: 1
        Buffers: shared hit=37
Planning Time: 0.140 ms
Execution Time: 0.302 ms
(8 rows)
```

Flg 25. Rendimiento en Explain Analyze.

Muchas visualizaciones interactúan segmento a segmentos la línea de trayectoria, hetampa de velocidad por tic, recalcula LAG(), entonces materializar reduce latencia de consulta.

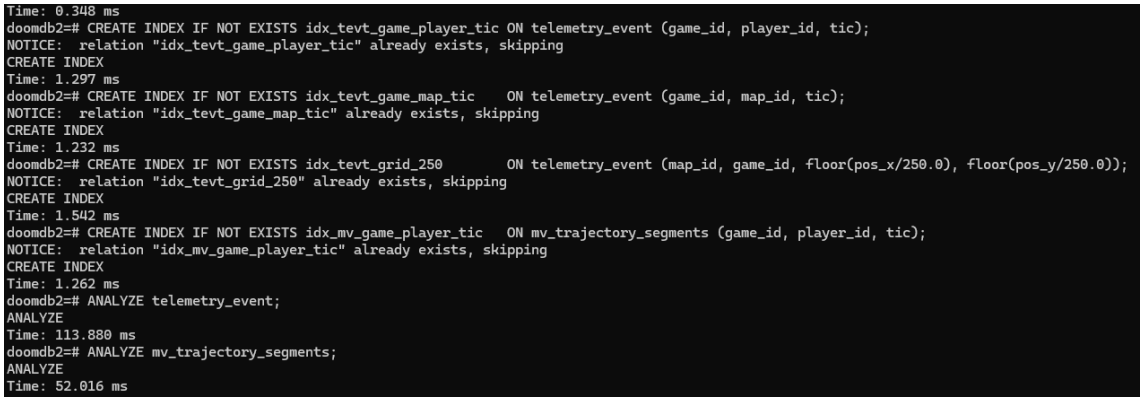
La vista materializada creada pre-calcula los segmentos entre tics consecutivos por (game_id, player_id) con su step_distance y step_speed.

Esta vista expone los segmentos ya calculados por (game_id, player_id, tic) con distancia por paso y speed, es necesario materializar para que los cálculos que usan ventanas y agregaciones se puedan tener persistidos y esto acelera las consultas de trayectoria y depuración.

El uso de la vista es filtrar por un jugador o juego y ordenar por tic para ver el recorrido de principio a fin.

Como se observa en las imágenes, la vista materializada mv_trajectory_segments almacena los pasos de trayectoria precomputados, con el índice (game_id, player_id_tic) las consultas filtradas por jugador y partida responden milisegundos.

- Creación de índices.



Flg 26. creación de índices.

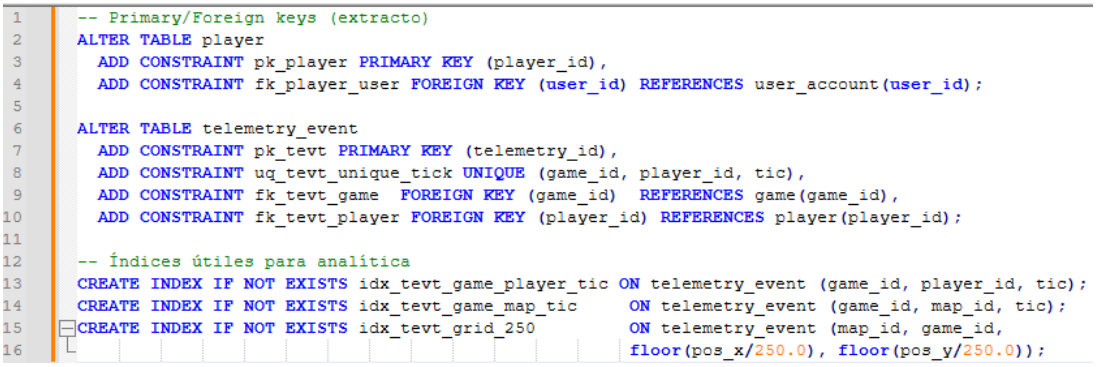
En cuestión al desempeño, los índices creados (por jugador/tiempo, por grid y por juego-map-tic) ayudan a que el motor elija planes de ejecución mas directos, las vistas dejan listas las métricas principales y la vista materializada permite revisar trayectorias sin recalcular todo cada vez.

Con la creación de los índices se tiene que son exactamente las columnas por las que se filtra y se orden en las consultas analíticas, esto permite que postgres salte a las filas que importan sin leer todo.

A nivel general se nota que los planes de ejecución empiezan a usar index Sca/Bitmap en vez de full scans y los tiempos se vuelven mas estables. Como conclusión general, con estos índices las consultas criticas quedan con una psita y escalan mejor y son más predecibles.

VI. DDL/CONSTRAINTS

Se modelo un usuario y un alias por separado para permitir que un mismo voluntario tenga múltiples identidades de juego, las partidas (game) se relacionan con episodios, mapas, sectores para consultas espaciales, la telemetra se guarda por tic para reconstruir trayectorias y los instrumentos UX (PENS, GUESS, BANGS) se separan en instrumentos, ítems y respuestas.



```sql
-- Primary/Foreign keys (extracto)
ALTER TABLE player
  ADD CONSTRAINT pk_player PRIMARY KEY (player_id),
  ADD CONSTRAINT fk_player_user FOREIGN KEY (user_id) REFERENCES user_account(user_id);

ALTER TABLE telemetry_event
  ADD CONSTRAINT pk_tevt PRIMARY KEY (telemetry_id),
  ADD CONSTRAINT uq_tevt_unique_tick UNIQUE (game_id, player_id, tic),
  ADD CONSTRAINT fk_tevt_game  FOREIGN KEY (game_id)  REFERENCES game(game_id),
  ADD CONSTRAINT fk_tevt_player FOREIGN KEY (player_id) REFERENCES player(player_id);

-- Índices útiles para analítica
CREATE INDEX IF NOT EXISTS idx_tevt_game_player_tic ON telemetry_event (game_id, player_id, tic);
CREATE INDEX IF NOT EXISTS idx_tevt_game_map_tic     ON telemetry_event (game_id, map_id, tic);
CREATE INDEX IF NOT EXISTS idx_tevt_grid_250         ON telemetry_event (map_id, game_id,
                                                     floor(pos_x/250.0), floor(pos_y/250.0));
```

Fig 1. DDL.

VII. ETL descripción

1. **Staging CSV:** Se crea una tabla staging_telemetry con columnas para recibir el archivo de los datos exportados del DOOM.
2. **Carga:** Se usa copy desde psql para cargar el CSV con cabecera.
3. Se hace una **limpieza** y **normalización** que transforman columnas (parse de posiciones, tiempos y estado) y se insertan en telemetry_event.
4. **Controles:** Se aplica UNIQUE (game_id, player_id, tic) para evitar duplicados y se descartan filas invalidas.
5. **Derivados:** Se crean vistas y una vista materializada para acelerar trayectorias.

```sql
-- Staging
CREATE TABLE IF NOT EXISTS staging_telemetry(
  "timestamp" timestamptz, tic bigint,
  x double precision, y double precision, z double precision,
  angle double precision, momx double precision, momy double precision, momz double precision,
  raw_line text
);

-- Migración a telemetry_event
INSERT INTO telemetry_event (game_id, player_id, tic, pos_x, pos_y, pos_z, health, armor, ammo, created_at, raw_line)
SELECT
  g.game_id, p.player_id, s.tic, s.x, s.y, s.z, /* ...estado si viene... */ now(), s.raw_line
FROM staging_telemetry s
JOIN
```

Fig 1. ETL.

VIII. AJUSTES CON LA ENTREGA 2

- **ER:** se mantienen (coherentes con implementación).
- **DDL/constraints:** ahora documentados (PK/FK/UNIQUE/Índices) con extractos.
- **ETL:** se describe el flujo completo (staging → carga → migración), con muestra cruda y comandos.
- **Consultas:** 5 analíticas con resultados y comparación EXPLAIN ANALYZE **antes/después de índices**.
- **Vistas:** se agregan definiciones y mini-diccionario de vistas, más MV con índice y EXPLAIN.

IX. DICCIONARIO DE VISTAS

Para la creación de vistas se agregaron diferentes conceptos, para ello se presenta una tabla con los conceptos.

view_player_measures	por player_id, devuelve total_distance.
avg_speed_per_tic	(fuente: telemetry_event + ventanas).
view_hotspots_250	por (game_id, map_id, cell_x, cell_y) devuelve hits (celdas 250×250).
view_hotspots_250	(materializada): por (game_id, player_id, tic) trae pos_x/pos_y/pos_z,
	step_distance, step_speed (usada para gráficos/recorridos).

X. CONCLUSIONES.

- En nivel general, el mapa es estable en duración, tiene un punto que concentra la actividad, permite estilos de juego distintos y mantiene una velocidad de movimiento constante.
- Se diseñaron los índices a la medida de cada consulta, aunque el dataset no es el mas grande, los tiempos son similares.
- Se realizaron vistas y una MV para convertir análisis en activos reutilizables y bajar la latencia de consumo.
- El proyecto permitió tener diferentes conocimientos de la asignatura, también se pudieron plantear diferentes formas de resolverlo y el poder jugar la extensión de DOOM lo hizo más interesante.