

Punto 15

Plan 1

La consulta cuenta la cantidad de pacientes que atendio el 'Dr. Sosa', en los hospitales con menos de 35 habitaciones en 'Vicente Lopez'

Sin indice

- 0.61 SEC
- La consulta sin indices trae toda las filas de la tabla paciente ~200000, y las escanea a todas para encontrar las que coinciden.
- Utiliza la clave primaria para buscar ya que no posee indice.

Con indice

- 0.11 SEC
- La consulta con indice trae solo ~15000 filas de la tabla atencioninternacion, aprovechando el indice idx_dr donde se lo filtra con un valor constante, en este caso 'Dr. Sosa'.
- En la tabla hospitales, se filtra por valores constantes aprovechando los indices idx_canthab e idx_ciu_hosp.

Conclusión

Se nota una mejor performance en la consulta con indices, ya que se maneja una cantidad total de filas mucho menor a las que se maneja en la consulta sin indices, también se aprovechan mejor los valores constantes en las clausulas del WHERE que son utilizados para buscar en los indices.

Plan 2

La consulta muestra todas las internaciones que atendio el 'Dr. Dominguez' en los hospitales con más de 42 habitaciones en 'Vicente Lopez' entre las fechas '2006-01-01' y '2006-12-31'.

Sin indice

- 0.50 SEC
- Se ejecuta sin orden, analizando toda la tabla de pacientes, luego la tabla de internaciones es filtrada por el dniPaciente, seguido de la tabla de hospitales filtrada por codHospital y por ultimo la tabla de atencioninternacion en la cual se comparan las columnas dniPaciente, fechaInicioInternacion,

Con indice

- 0.03 SEC
- Se ejecuta ordenada de forma que se filtran primero los hospitales de 'Vicente Lopez' y luego las atenciones del 'Dr. Dominguez', luego los pacientes que tienen internaciones atendidas por el doctor entre las fechas establecidas.

Conclusión

Al igual que el analisis del Plan 1, en el Plan 2 encontramos una gran mejora al ejecutar la consulta aprovechando los indices, el total de filas procesadas para encontrar el resultado es muy significativo (~200000 sin indices, ~30000 con indices), en esta consulta también son aprovechadas las referencias constantes con las que se comparan los valores de los indices.

Plan 3

La consulta muestra lo mismo que la consulta del plan 2, solamente que utiliza una tabla ya filtrada por ciudadhospital='Vicente Lopez' y cantidadhabitaciones >42.

Sin indice

- 0.22 SEC
- Al tener una sub consulta en el FROM, hace que se recorra la totalidad de la tabla hospital, no utilizando así ningún tipo de clave, la consulta escanea ~700000 filas para poder encontrar todos los resultados.

Con indice

- 0.03 SEC
- El orden de ejecución cambia, aprovechando el índice idx_dr solo se examinan ~15000 filas de la tabla atencioninternacion, la tabla hospital sigue siendo

procesada en ultimo lugar, pero en este caso utilizando los indices idx_canthab e idx_ciu_hosp examinando así solo 2 filas para encontrar los resultados.

Conclusión

Teniendo el mismo objetivo las consultas del Plan 2 y 3, se puede ver una gran baja de performance utilizando un sub consulta en el FROM y no utilizando indices el total de filas analizadas por el Plan 3 sin indices es mas del triple de filas que se tuvieron que analizar en el Plan 2 sin indices. Con respecto a la consulta con idices el grado de performance no se ve afectado por que tanto en el Plan 2 como en el Plan 3 se manejan la misma cantidad de filas para encontrar los resultados.

Punto 16

Plan A

La consulta cuenta la cantidad de pacientes que atendio el 'Dr. Sosa', en los hospitales con menos de 35 habitaciones en 'Vicente Lopez'

Sin indice

- (811 Rows) 11.24 SEC
- No se utiliza ningún tipo de clave para buscar los datos, se tiene que escanear todas las tuplas de la tabla internacion ~1350000, es el peor caso posible en performance.

Con indice

- (811 Rows) 0.27 SEC
- A primera vista se evidencia el reducido tiempo de ejecución con respecto a la consulta sin indices, para escanear se realiza un merge de indices (idx_dr,idx_ciu_hosp), logrando así tener que utilizar solo ~6000 tuplas de la tabla internacion.

Plan B

La consulta muestra todas las internaciones que atendio el 'Dr. Dominguez' en en los hospitales con más de 42 habitaciones en 'Vicente Lopez' entre las fechas '2006-01-

01' y '2006-12-31'.

Sin indice

- (72 Rows) 11.59 SEC
- Al igual que en la consulta del PlanA sin indices, no se utiliza ninguna clave para realizar el join, y se tiene que escanear la totalidad de las tuplas en la tabla internacion.

Con indice

- (72 Rows) 0.28 SEC
- Como era de esperar, el uso de indices permite reducir el tiempo que lleva la ejecución de la consulta, y tambien la cantidad de tuplas escaneadas (~6000), en este caso tambien se realiza un merge de indices (idx_dr,idx_ciu_hosp).

Plan C

La consulta muestra todas las internaciones que atendio el 'Dr. Dominguez' en en los hospitales de 'Vicente Lopez' entre las fechas '2006-01-01' y '2006-12-31'.

Sin indice

- (1445 Rows) 11.59 SEC
- Al igual que en la consulta del PlanA y PlanB sin indices, no se utiliza ninguna clave para realizar el join, y se tiene que escanear la totalidad de las tuplas en la tabla internacion.

Con indice

- (1445 Rows) 0.63 SEC
- En esta consulta solo se utiliza el indice idx_dr, haciendo procesar solo ~26500 tuplas de la tabla internacion, el tiempo de ejecución es notoriamente menor en comparacion a el de la consulta sin indices.

Conclusión

En los tres planes los puntos más claros en los que se diferencian las consultas que no aprovechan los indices con las que si lo hacen, son el tiempo en el que son ejecutadas y la cantidad de tuplas que tienen que ser escaneadas para poder encontrar las coincidencias. En las consultas sin indices los tiempos de ejecución rondan los 12 segundos aproximadamente, contrastando así con las consultas que

utilizan indices en donde los tiempos son de entre 0.20 y 0.65 segundos, el otro gran problema de las consultas sin indices es que tienen que escanear la totalidad de tuplas de la tabla internacion ~1350000, en cambio las que utilizan indices solo tienen que escanear entre ~6000 y ~26500 tuplas como maximo.

Quedando así demostrado que en una base de datos desnormalizada la utilizacion de indices produce un gran incremento de performance en la ejecucion de consultas.