

ÍNDICES

RFC1: Para optimizar la consulta y mejorar el rendimiento, se considera la creacion de indices en las columnas utilizadas en las operaciones de JOIN y las condiciones de filtro.

1. Índice en la tabla habitaciones en la columna num_habitacion: como se agrupa por h.num_habitacion y realizando un JOIN con la tabla habitaciones, un indice en esta columna puede ayudar a acelerar la operacion. Índice primario.
2. Índice en la tabla reservas en la columna habitacion: Dado que la tala reservas se une con habitaciones basada en la columna habitacion, crear un indice en esta columna puede mejorar el rendimiento de la operacion JOIN.Índice secundario.
3. Índice en la tabla consumen en la columna cliente: Dado que la tabla consumen se une con la tabla reservas basada en la columna cliente, crear un indice en esta columna puede mejorar el rendimiento.Índice B-tree.
4. Índice en la tabla servicios en la columna tipo: puesto que se realiza un JOIN con la tabla servicios basado en la columna tipo, un indice en esta columna puede ser util.Índice B-tree.

RFC2: La consulta involucra tablas como tipos_servicio, consumen y servicios. Para optimizar esta consulta, determinamos los siguientes índices:

1. Índice en la tabla Tipos_Servicio en la columna id_tipo: Dado que la tabla tipos_servicio es la tabla princpial en esta consulta, un indice en la columna id_tipo es importante para acelerar la búsqueda y JOIN.
Tipo de índice: Índice primario.
2. Índice compuesto en la tabla consumen en las columnas servicio y fecha: Esto mejorara el rendimiento de la condicion C.fecha BETWEEN : fecha_inicio AND fecha_fin .
Tipo de indice: Indice B_tree Compuesto
3. Índice en la tabla servicios en la columna Tipo: Esto facilitara el JOIN con la tabla servicios.
Tipo indice: Indice B-tree.

RFC3: La consulta involucra tablas como habitaciones y reservas. Para optimizar esta consulta, determinamos los siguientes indices:

1. Índice en tabla habitaciones en la columna num_habitacion :Dado que la consulta agrupa por el numero de habitacion, un indice en esta columna mejorara el rendimiento.
Tipo de indice: Indice B-tree
2. Índice en la tabla reservas en la columna habitacion: Este indice se utiliza para realizar la operacion de JOIN.
Tipo de indice: Indice B-tree
3. Índice compuesto en la tabla reservas en las columnas fecha_entrada y fecha_salida_ Dado que hay condiciones de filtro en estas columnas, un indice compuesto puede mejorar el rendimiento de la consulta.
Tipo de indice: Indice B_tree compuesto.

RFC4: La consulta tiene multiples condiciones en la clausula WHERE, algunas de las cuales son condicionales(dependen de si las variables de enlace están establecidas o no). Por lo cual se establecieron los siguientes índices:

1. Índice en la tabla tipos_servicio en la columna id_tipo: Dado que tipos_servicio es la tabla principal en la consulta, un índice en id_tipo es importante.
Tipo de índice: Índice primario
2. Índice en la tabla servicio en la columna tipo: Facilita la union JOIN con la tabla servicios.
Tipo de índice: Índice B-tree.
3. Índice compuesto en la tabla consumen y en la columna servicio y fecha. Esto puede mejorar el rendimiento del BETWEEN. Además, dado que esta tabla se une en función de la variable servicio, un índice en esta columna también es útil.
Tipo de índice: Índice B-tree compuesto
4. Índice en la tabla servicios en la columna Precio: Dado que hay una condición relacionada con el precio, un índice en la columna precio de la tabla puede ayudar a acelerar las búsquedas de precio.
Tipo de índice: Índice B-tree.

RFC5: La consulta involucra las tablas consumen y tipos_servicio, puedes optimizar el rendimiento mediante la creación de los siguientes índices ;:

1. Índice en la tabla consumen en la columna cliente: Como se filtra por la columna cliente, un índice en esta columna ayudara a acelerar la búsqueda de registros relacionados con un cliente específico. Índice primario
2. Índice en la tabla consumen en la columna fecha: Dado que también se filtra por la columna fecha. Índice primario
3. Índice en la tabla consumen en la columna servicio: Dado que la tabla tipos_servicio se une en función de C.servicio, un índice en esta columna también puede mejorar el rendimiento. Índice B_tree

RFC6.1: La consulta genera un conjunto de fechas secuenciales dentro del rango de fechas de reservas y luego cuenta cuantas reservas hay para cada fecha en ese rango.

Para optimizar esta consulta creamos los siguientes índices:

1. Índice en la tabla reservas en la columna fecha_entrada y fecha_salida: Dado que la consulta involucra condiciones de rango en estas columnas, un índice compuesto en ambas columnas ayudara a mejorar el rendimiento. Índice compuesto
2. Índice en fechas sobre la columna fecha reserva. Índice B-tree

RFC6.2: Para optimizar la consulta y mejorar el rendimiento, creamos el siguiente índice:

1. Índice en la tabla consumen en la columna fecha: Como se esta agrupando y ordenando por esta columna, entonces crear un índice en fecha ayudara a acelerar la búsqueda y el proceso de ordenación. Índice B-tree

RFC6.3:

1. Índice compuesto en la tabla reservas, en las columnas fecha_entrada, fecha_salida y fecha_salida, fecha_entrada: Dado que se busca contar las reservas para cada fecha en orden ascendente, puedes crear un índice en la fecha_entrada, fecha_salida y la diferencia entre fecha_salida y fecha entrada. Esto facilitará la búsqueda y el cálculo de las fechas. Índice compuesto
- 2.

RFC7: La consulta realiza una serie de calculos y operaciones utilizando varias tablas. Para optimizar esta consulta y mejorar el rendimiento es importante tener en cuenta las condiciones de filtro.

1. Índice en la tabla reservas en las columnas fecha_entrada y fecha_salida: Dado que se realizan búsquedas basadas en las fechas de entrada y salida en la tabla reservas, un índice compuesto en estas dos columnas puede mejorar el rendimiento.
2. Índice en la tabla usuarios en la columna num_documento: Puesto que se filtra por el número de documento de los usuarios, un índice en esta columna de la tabla usuarios es beneficioso.
3. Índice en la tabla consumen en la columna cliente: Dado Que la tabla consumen se une con la tabla usuarios basada en la columna cliente, crear un índice en esta columna puede mejorar el rendimiento.
4. Índice en la tabla servicios en la columna tipo: Como se realiza una unión con la tabla servicios basada en la columna tipo; un índice en esta columna tambien puede ser util.

RFC8: La consulta calcula el numero de semanas en las que el numero de servicios consumidos fue menor que 3. Para optimizar esta consulta y mejorar el rendimiento, se considera crear índices en las columnas utilizadas en las operaciones de JOIN y las condiciones de filtro. Dado que en esta consulta no se filtra directamente por fechas o servicios, no es necesario crear índices en esas columnas especificas. Sim embargo, se pueden crear índices en las columnas utilizadas para JOIN y agrupación.

1. Índice en la tabla Tipos servicios en la columna id tipo: Como se realiza un Join con la tabla tipos servicio, se crea un índice en la columna id_tipo para acelerar la operacion de JOIN. Índice B-tree
2. Índice en la tabla consumen de las columnas fecha y servicio, que ayudara a mejorar significativamente el rendimiento en la consulta. Índice compuesto B-tree

RFC9: Para optimizar la consulta y mejorar el rendimiento, es importante tener en cuenta las condiciones de filtro y las tablas involucradas:

1. Índice en la tabla consumen en la columna servicio: Dado que la consulta filtra por C.servicio, un índice en esta columna puede mejorar el rendimiento de la búsqueda. Índice hash
2. Índice en la tabla usuarios en la columna tipo: Cómo se filtra por u.tipo, un índice en esta columna ayudara a acelerar la busqueda de usuarios con el tipo especificado.Índice hash
3. Índice en la tabla consumen en la columna cliente: Puesto que se realiza un JOIN con la tabla consumen basado en la columna cliente, crear un índice en esta columna mejorara el rendimiento de la operación de JOIN. Índice B-tree

4. Índice en la tabla consumen en la columna fecha: Dado que también se filtra por la columna fecha. Índice B-tree

RFC10: La consulta se centra en encontrar usuarios que no han consumido un servicio específico en un rango de fechas dado. Para optimizar esta consulta y mejorar el rendimiento, se crean los siguientes índices:

1. Índice en la tabla usuarios en la columna tipo: Dado que se filtra por u.tipo, un índice en esta columna ayudará a acelerar la búsqueda de usuarios con el tipo especificado. Índice hash
2. Índice en la tabla tipos de usuario en la columna id_tipo: como se realiza un JOIN con la tabla tipos_usuario, se crea un índice en la columna id_tipo para acelerar la operación de JOIN. Índice B-tree
3. Índice compuesto en la tabla consumen en las columnas servicio y fecha: Dado que la subconsulta busca consumos en función del servicio y la fecha, un índice compuesto en estas dos columnas puede mejorar el rendimiento de la subconsulta. Índice B-tree

RFC11:

RFC 11.1: En esta consulta..

1. Índices en las columnas "fecha_salida", "fecha_entrada" y "habitacion" de la tabla reservas. Estas consultas se utilizan para calcular la ocupación de las habitaciones. B-tree
2. Índices en las columnas fecha y servicio de la tabla consumen. Estas columnas se utilizan para determinar los tipos de servicios más populares en cada semana. B-tree

RFC 11.2:

1. Índice en la columna fecha de la tabla consumen: Esta columna se utiliza en la cláusula de unión y en la cláusula WHERE para filtrar las fechas del consumo. Índice secundario.
2. Índice en la columna id_tipo de la tabla tipos_servicio: La columna id_tipo se utiliza en la cláusula de unión para relacionar los servicios consumidos con sus tipos. Índice primario

RFC 11.3:

1. Índice en la columna habitacion de la tabla reservas que se utiliza en la unión con la tabla habitaciones. Índice secundario
2. Índice en la columna fecha reserva de la tabla fechas que se utiliza en la unión con la tabla reservas. Índice B-tree
3. Índice en las columnas inicio_semana y fin_semana de la tabla semanas que se utilizan en la unión con la tabla fechas y semanas

RFC 11.4:

1. Índice en la columna habitacion de la tabla reservas que se utiliza en la unión con la tabla habitaciones. Índice secundario

2. Índice en la columna fecha entrada de la tabla reservas que también se utiliza en la unión con la tabla fechas. Índice B-tree

RFC12: La consulta realiza demasiadas operaciones, joins y cálculos. Para optimizar esta consulta y mejorar el rendimiento se realizan los siguientes índices:

1. Índices compuestos en las columnas de fecha: La consulta realiza múltiples comparaciones de fechas en varias tablas, por lo que sería beneficioso crear índices compuestos en las columnas de fecha, como R.fecha_entrada, R.fecha_salida, C.fecha y ReS.dia_reserva. Índice compuesto secundario
2. Índice en la tabla servicios en la columna precio: Como se filtra por el precio de los servicios, un índice en la columna precio de la tabla servicios puede ser útil. Índice secundario
3. Índice en la tabla reservas servicio en la columna hora inicio y hora fin: Dado que se realizan cálculos relacionados con la hora, la creación de un índice en estas columnas puede mejorar el rendimiento. Índice compuesto
4. Índice en la tabla usuarios en la columna tipo: puesto que se filtra por u.tipo, un índice en esta columna de la tabla usuarios es beneficioso. Índice primario.