

Using UNION or IN Instead of OR

Agnelli, Nicolas Matias
Gonzalez Agustina Belén



Using UNION or IN Instead of OR

Definición:

Al gestionar bases de datos, resulta crucial optimizar las consultas SQL con el fin de mejorar el rendimiento y disminuir los tiempos de respuesta. Evitar el uso del operador OR en las cláusulas WHERE es una práctica recomendada, dado que puede inducir a la base de datos a realizar un escaneo exhaustivo de la tabla para encontrar las coincidencias. Para potenciar la eficiencia al filtrar filas, se aconseja emplear UNION o IN en lugar de OR, ya que estos pueden ser sutilmente optimizados por el motor de la base de datos.

```
SELECT nombre, apellido  
FROM empleados  
WHERE estado_civil = 'Soltero'  
OR estado_civil = 'Casado'
```



```
SELECT nombre, apellido  
FROM empleados  
WHERE estado_civil IN('Soltero','Casado')
```

Using UNION or IN Instead of OR

Pre-condición:

- Debe contener al menos una cláusula OR
- No contendrá ninguna clausula AND
- Todos los campos de la condición tienen que ser iguales
- El único tipo de comparación será igualdad
- Debe ser sintácticamente correcto

Post-condición:

- No debe contener ninguna cláusula OR
- No debe contener ninguna cláusula AND
- Debe contener la cláusula IN

Using UNION or IN Instead of OR

Implementacion:

- **PreconditionVisitor**

- Reimplementa visitParse(...)
- Reimplementa visitSelect_core(...)
- Reimplementa defaultResult(...)

- **TransformVisitor**

- Reimplementa visitParse(...)
- Reimplementa visitSelect_core(...)
- Reimplementa defaultResult(...)
- Reimplementa visitTable_or_subquery(..)

- **PostconditionsVisitor**

- Reimplementa visitParse(...)
- Reimplementa defaultResult(...)

- **ReplaceOrWithInRefactoring**

- Subclase de Refactoring

- **ReplaceOrWithInRefactoringTest**

- Implementa @test TestRefactorCorrecto()
- Implementa @test TestRefactorFallidoDistintosCampos()
- Implementa @test TestRefactorFallidoFaltaOr()
- Implementa @test TestRefactorFallidoSintacticamente()
- Implementa @test TestRefactorFallidoComparacionIncorrectaWhere()
- Implementa @test TestRefactorFallidoTieneAnd()
- Implementa @test TestRefactorFallidoConsultaInsert()
- Implementa @test TestRefactorFallidoConsultaCreate()

Using UNION or IN Instead of OR

Conclusión:

- PreconditionVisitor para validar las precondiciones re implementa métodos como visitExpr y visitSelect_Core donde se evalúan todas las precondiciones.
- PostConditionVisitor para evaluar que se encuentre la palabra IN luego de la transformación utilizando el método visitExpr para poder evaluar la expresión transformada.
- TransformVisitor que se encarga de unificar en una cadena cada parte de la consulta y dentro de la cláusula WHERE la transformación del OR en la palabra clave IN con su respectiva estructura.

Si las postcondiciones resultan exitosas y no alteran el funcionamiento de la consulta, podemos concluir que el refactoring se ha realizado correctamente.