

# RemoveAliasRefactoring

Alcorta Lautaro, Otero Bárbara

# RemoveAliasRefactoring

## Definición:

Dada una consulta SQL y un alias a eliminar, busca dicho alias en la consulta, lo elimina y reemplaza las referencias al mismo por el nombre de la tabla o columna correspondiente.

```
SELECT a.id, a.name, m.name  
FROM airplane as a, maker as m  
WHERE a.maker_id = m.id
```

Removemos el  
alias "a"

```
SELECT airplane.id,  
airplane.name, m.name  
FROM airplane, maker as m  
WHERE airplane.maker_id = m.id
```

# RemoveAliasRefactoring

## **Preciendiciones (entrada):**

- Es una consulta de SQLite
- La consulta es válida (Parsea)
- El alias a remover existe en la consulta y está asociado a una tabla o columna.

## **Postciondiciones (salida):**

- Es una consulta de SQLite
  - La consulta es válida (Parsea)
-

# RemoveAliasRefactoring

## Implementación:

- AliasCheckerVisitor
  - Recorre el árbol generado por antlr buscando el alias a remover, si lo encuentra, pone en verdadero la variable booleana "aliasEncontrado" y guarda la referencia a la tabla o columna en "aliasReference".
- RemoveAliasVisitor
  - Recorre el árbol generado por antlr reemplazando el alias a remover por el nombre de tabla o columna guardado en "aliasReference", además elimina la palabra reservada "AS" de la consulta que está antes del alias a eliminar.
- RemoveAliasRefactoring
  - Subclase de Refactoring, define la lógica para chequear post y pre condiciones, hacer las transformaciones y settear el alias a remover.
- RemoveAliasRefactoringTest
  - Colección de test para comprobar que dadas las precondiciones, realiza correctamente las transformaciones.

# RemoveAliasRefactoring

## Conclusiones:

- Se chequea que como precondition parsee y el alias a remover exista y que como postcondición parsee y el alias a remover no exista, esto lo hacemos mediante el AliasCheckerVisitor.
- Es un refactoring porque la consulta antes y después de la transformación tiene el mismo comportamiento (solo cambia la sintaxis).