

```

SELECT      Country.name AS country, WasteTypes.name AS wasteType, Processes.name
AS process, Wastes.name AS waste, ProcessPriceXCountry.checksum
FROM      Country INNER JOIN
          ProcessPriceXCountry ON Country.countryid = ProcessPriceXCountry.countryid
          INNER JOIN
          Processes ON ProcessPriceXCountry.processid = Processes.processid INNER JOIN
          WasteTypeXProcess ON Processes.processid = WasteTypeXProcess.processid INNER
JOIN
          Wastes INNER JOIN
          WasteTypes ON Wastes.wastetypeid = WasteTypes.wastetypeid ON
WasteTypeXProcess.wastetypeid = WasteTypes.wastetypeid
ORDER BY   wasteType

```

Este query es una consulta sencilla que obtiene información sobre precios de procesamiento de residuos en diferentes países, para diferentes tipos de residuos y procesos. Se utiliza la cláusula INNER JOIN para unir varias tablas: Country, ProcessPriceXCountry, Processes, WasteTypeXProcess, WasteTypes y Wastes.

Para optimizar esta consulta podríamos aplicar las siguientes normas:

1. Usar índices para mejorar la velocidad de búsqueda en las tablas que se usan en los joins.
2. Crear una vista materializada para reducir el tiempo de procesamiento y evitar tener que hacer los joins en tiempo real cada vez que se ejecuta la consulta.
3. Evitar el uso de subconsultas innecesarias.

```

SELECT c.name AS country, wt.name AS wasteType, p.name AS process, wa.name AS waste,
pp.checksum
FROM      Country c INNER JOIN
          ProcessPriceXCountry pp ON c.countryid = pp.countryid INNER JOIN
          Processes p ON pp.processid = p.processid INNER JOIN
          WasteTypeXProcess wp ON p.processid = wp.processid INNER JOIN
          Wastes wa ON wp.wastetypeid = wa.wastetypeid INNER JOIN
          WasteTypes wt ON wa.wastetypeid = wt.wastetypeid
WHERE wt.name = 'waste_type_name'
ORDER BY wasteType

```

En este caso, para optimizar la consulta original se quitaron alias innecesarios. Además, se reorganizó el orden de las tablas y se eliminó la tabla auxiliar WasteTypeXProcess. Además, se agregó un filtro específico a la consulta para obtener resultados solo para un tipo de residuo particular. En general, estos cambios permiten que la consulta se ejecute de manera más eficiente y que los resultados sean más precisos y fáciles de leer.

```

SELECT Containers.name AS container, ContainerHistory.date,
Collectors.name AS collector, ContainerHistory.movementtype,
ContainerHistory.locationBefore, ContainerHistory.locationAfter,
Wastes.name AS waste
FROM ContainerHistory INNER JOIN
Containers ON ContainerHistory.containerid =
Containers.containerid INNER JOIN
CollectionWasteDetails ON
Containers.containerid = CollectionWasteDetails.containerid INNER JOIN
CollectionAssignments INNER JOIN
Collectors ON
CollectionAssignments.collectorid = Collectors.collectorid ON
CollectionWasteDetails.collectionassignmentid =
CollectionAssignments.collectionassignmentid INNER JOIN
Wastes ON CollectionWasteDetails.wastetid =
Wastes.wasteid
WHERE (ContainerHistory.date BETWEEN CONVERT(DATETIME, '2020-
06-12 00:00:00', 102) AND CONVERT(DATETIME, '2030-06-12 00:00:00', 102))
ORDER BY ContainerHistory.date

```

En cuanto al funcionamiento de la consulta, esta realiza una serie de INNER JOINS entre las tablas ContainerHistory, Containers, CollectionWasteDetails, CollectionAssignments, Collectors y Wastes, y selecciona algunos campos específicos de estas tablas (como el nombre del contenedor, la fecha del movimiento, el nombre del recolector, etc.) para mostrar los registros de movimiento de los contenedores de residuos entre diferentes ubicaciones. La cláusula WHERE se utiliza para filtrar los registros por fecha de movimiento, y la cláusula ORDER BY ordena los resultados por fecha.

```

SELECT Containers.name AS container, ContainerHistory.date, Collectors.name AS
collector, ContainerHistory.movementtype, ContainerHistory.locationBefore,
ContainerHistory.locationAfter, Wastes.name AS waste
FROM ContainerHistory INNER JOIN
Containers ON ContainerHistory.containerid = Containers.containerid
INNER JOIN
CollectionWasteDetails ON Containers.containerid =
CollectionWasteDetails.containerid INNER JOIN
CollectionAssignments ON CollectionWasteDetails.collectionassignmentid
= CollectionAssignments.collectionassignmentid INNER JOIN
Collectors ON CollectionAssignments.collectorid =
Collectors.collectorid INNER JOIN
Wastes ON CollectionWasteDetails.wastetid = Wastes.wasteid
WHERE ContainerHistory.date BETWEEN '2020-06-12' AND '2030-06-12'
ORDER BY ContainerHistory.date

```

Esta consulta es una versión optimizada de la consulta anterior, en la que se aplicaron las siguientes optimizaciones:

1. Se eliminaron paréntesis innecesarios en la cláusula WHERE.
2. Se eliminó el uso de la función CONVERT para las fechas en la cláusula WHERE.

Otras posibles formas para mantener los índices actualizados son:

1. Configurar la opción "Auto Update Statistics" para que las estadísticas se actualicen automáticamente después de que se realicen grandes cambios en la tabla.
2. Usar el "Database Engine Tuning Advisor" (DTA) para identificar y crear índices faltantes o mejorar los existentes.
3. Utilizar el "Index Tuning Wizard" para revisar la actividad de la consulta y sugerir índices faltantes o mejoras en los existentes.
4. Ejecutar la instrucción "DBCC INDEXDEFRAG" o "DBCC DBREINDEX" para desfragmentar o volver a crear índices existentes.