Limpieza de datos: acercamiento

1. Identificación y manejo de datos faltantes:

Práctica en SQLite:

```
-- Identificar datos faltantes en la columna "Composer" de la tabla "tracks"
SELECT trackid, name, composer
FROM tracks
WHERE composer IS NULL;
```

2. Detección y corrección de outliers:

```
WITH Quantiles AS (
    SELECT
        UnitPrice,
        NTILE(4) OVER(ORDER BY UnitPrice) AS quartile
    FROM tracks
)

, Q1 AS (
    SELECT
        MAX(UnitPrice) AS first_quartile
    FROM Quantiles
    WHERE quartile = 1
)

, Q3 AS (
    SELECT
        MAX(UnitPrice) AS third_quartile
    FROM Quantiles
    WHERE quartile = 3
)

-- Detección de outliers
SELECT T.trackid, T.name, T.UnitPrice
FROM tracks T, Q1, Q3
WHERE T.UnitPrice < first_quartile - 1.5 * (third_quartile - first_quartile)
   OR T.UnitPrice > third_quartile + 1.5 * (third_quartile - first_quartile);
```

3. Normalización y estandarización de datos:

```
-- Estandarización
SELECT trackid, name,
       (UnitPrice - avg(UnitPrice) OVER()) / stddev(UnitPrice) OVER() as standardized_pric
e
FROM tracks;
```

4. Conversión de Tipos de Datos

```
UPDATE invoices
SET InvoiceDate = DATE(InvoiceDate);
```

5. Eliminación de Filas Duplicadas

Práctica: Eliminar registros duplicados de la tabla `albums`.

```
DELETE FROM albums
WHERE rowid NOT IN (
    SELECT MIN(rowid)
    FROM albums
    GROUP BY Title);
```

6. Corrección de Valores Inconsistentes

Práctica: Supongamos que por algún error, algunos tracks tienen un precio de `-0.99`. Vamos a corregir eso:

```
UPDATE tracks
SET UnitPrice = 0.99
WHERE UnitPrice = -0.99;
```

7. Limpieza de Cadenas de Texto

Práctica: Asegurarse de que todos los nombres de los géneros estén en título (primera letra en mayúscula):

```
UPDATE genres
SET Name = UPPER(substr(Name, 1, 1)) || LOWER(substr(Name, 2));
```

8. Reemplazo de Valores Null con un Valor Predeterminado

Práctica: Suponiendo que algunas canciones no tienen un género asignado (NULL en la columna `GenreId`), podríamos reemplazar esos valores NULL con un valor predeterminado (por ejemplo, un `GenreId` que represente "Desconocido").

```
-- Primero, podríamos asegurarnos de que hay un género llamado 'Desconocido'
INSERT OR IGNORE INTO genres (Name) VALUES ('Desconocido');

-- Luego, actualizaríamos los tracks sin género con este valor predeterminado
UPDATE tracks
SET GenreId = (SELECT GenreId FROM genres WHERE Name = 'Desconocido')
WHERE GenreId IS NULL;
```

▼ Normalización y Estandarización

La normalización y la estandarización son técnicas utilizadas en el preprocesamiento de datos, especialmente en análisis de datos y aprendizaje automático. Ambas ayudan a escalar y transformar los datos, pero de diferentes maneras.

Normalización:

- **Objetivo**: Escalar los datos para que se encuentren en un rango específico, típicamente entre 0 y 1.
- **Fórmula:** $\left(x_{norm} = \frac{x - min(x)}{max(x) - min(x)}\right)$

- **Utilidad**: Es útil cuando queremos que nuestros datos estén en una escala comparable y no queremos que las características (features) con escalas numéricas más grandes dominen el modelo.

Estandarización:

- **Objetivo**: Reescalar los datos para que tengan una media de 0 y una desviación estándar de 1.
- **Fórmula**: $\left(x_{stand} = \frac{x-\mu}{\sigma}\right)$, donde $\left(\mu\right)$ es la media y $\left(\sigma\right)$ es la desviación estándar.
- **Utilidad**: Es útil para algoritmos que asumen que los datos están centrados, como la regresión lineal, SVM o redes neuronales. También es esencial cuando las características tienen diferentes unidades o escalas.

Ejemplo práctico:

Imagina un grupo ha presentado dos exámenes en diferentes asignaturas. El primer examen tenía un total de 50 puntos y el segundo examen tenía un total de 100 puntos. Se reciben las calificaciones y se quiere comparar el desempeño de los alumnos en ambos exámenes.

Problema: Las calificaciones de los dos exámenes están en diferentes escalas. Comparar una calificación de 40/50 en el primer examen con una calificación de 80/100 en el segundo examen directamente no sería justo.

Normalización:

- **Objetivo**: Convertir las calificaciones para que todas estén en una escala de 0 a 1.

  Por ejemplo, una calificación de 40 en el examen de 50 puntos se convertiría en $\frac{40}{50} = 0.8$ y una calificación de 80 en el examen de 100 puntos también sería 0.8. Ahora puedes ver fácilmente que el desempeño en ambos exámenes es el mismo.

Estandarización:

- **Objetivo**: Convertir las calificaciones para que tengan una media de 0 y una desviación estándar de 1. Esto es útil si quieres saber cuánto se desvió cada calificación del promedio.

Supongamos que en el examen de 50 puntos, la media de las calificaciones es 30 y la desviación estándar es 10. Una calificación de 40 tendría una puntuación estandarizada de $\frac{40-30}{10} = 1$, lo que significa que está 1 desviación estándar por encima de la media.