

## Prueba Técnica Data Quality Tester

Tecnologías sugeridas: SQL + Python (pandas o Great Expectations) o framework de preferencia.

### Datos de entrada

Tabla: customers\_raw

customer_id	name	email	country
1	Ana Torres	ana@email.com	Colombia
2	Juan Pérez	NULL	Mexico
3	Laura Gómez	laura_gomez@email.com	NULL
4	Juan Pérez	juanperez@email.com	Mexico
5	NULL	andres@email.com	Chile

Tabla: transactions\_raw

transaction_id	customer_id	amount	date
100	1	200.0	2025-01-01
101	2	150.0	2025-01-02
102	2	150.0	2025-01-02
103	3	NULL	2025-01-03
104	6	300.0	2025-01-04

### ◇ Parte 1: Validaciones SQL

#### 1. Completitud:

Escribe una consulta que detecte registros con valores NULL en campos obligatorios (``name``, ``email``, ``country`` en ``customers_raw``; ``amount`` en ``transactions_raw``).

#### 2. Unicidad:

Escribe una consulta que identifique duplicados de clientes (por ``email``) en ``customers_raw``.

#### 3. Integridad referencial:

Detecta transacciones con ``customer_id`` inexistente en ``customers_raw``.

#### 4. Reconciliación:

Escribe una consulta que compare totales de ``amount`` por país entre tablas ``transactions_raw`` y ``transactions_clean`` (post-transformación).

### ◇ Parte 2: Automatización de Pruebas de Calidad

#### 1. Implementa al menos 3 tests automatizados de calidad de datos, ya sea:

- Con Python (pandas o pytest)
- O con Great Expectations, dbt tests, o el framework que prefieras.

2. Los tests deben validar automáticamente las reglas definidas en la parte 1.
3. Cada test debe fallar si la regla no se cumple y generar un reporte o salida legible.