```
-- PART 1 - MISSING VALUES DETECTION
-- ================================================

-- We first connect to the "limpieza_db" database.
psql -U postgres -d limpieza_db

-- A cell with missing data is marked as "NULL"
-- in SQL. So we can perform the detection using
-- WHERE, for both numeric and categorical variables.

-- Let's look at the contents of the "inversionistas" table.
SELECT * FROM inversionistas;

-- The numeric variables are "edad" and "monto," while the
-- only categorical variable is "categoria"
\d inversionistas

-- Example 1: Let's detect missing values in the numeric variables
SELECT *
FROM inversionistas
WHERE edad IS NULL OR monto IS NULL;

-- Example 2: Let's detect missing values in the categorical variables
SELECT *
FROM inversionistas
WHERE categoria IS NULL;

-- Example 3: and we can combine the above into a single query
-- to detect all rows with missing data (whether numeric or categorical)
SELECT *
FROM inversionistas
```

```
WHERE edad IS NULL OR monto IS NULL OR categoria IS NULL;


-- PART 2 - HANDLING MISSING VALUES
-- ================================================

-- Once the missing values have been detected, we can perform their
handling.
-- This handling depends on the type of variable we have (numeric or
-- categorical)
-- and SQL has some basic tools for this handling:


-- Example 1: delete the record (for numeric or categorical variables)
-- In this case, we can simply repeat the previous query, changing
-- "IS NULL" to "IS NOT NULL" and "OR" to "AND" to preserve only records
-- (rows) that are complete.
SELECT *
FROM inversionistas
WHERE edad IS NOT NULL AND monto IS NOT NULL AND categoria IS NOT NULL;


-- Example 2: If the variable is numeric, we can perform imputation by the
-- mean.
SELECT id, nombre, edad,
      -- Edad: impute missing data using the mean or return the existing
      -- record.
      CASE
            WHEN edad IS NULL THEN (SELECT AVG(edad) FROM inversionistas
            WHERE edad IS NOT NULL)
      ELSE edad
END AS edad_imput,
-- Continue with SELECT amount,
-- Amount: impute missing data using the mean or return the existing
-- record.
```

```
CASE
      WHEN monto IS NULL THEN (SELECT AVG(monto) FROM inversionistas WHERE
      monto IS NOT NULL)
      ELSE monto
END AS monto_imput,
FROM inversionistas;
-- And remember that imputation by the mean
-- is sensitive to the presence of outliers. So we should handle the
-- outliers first and then perform the imputation.


-- Example 3: If the variable is categorical, we can impute by the most
-- frequent value.
SELECT id, nombre, categoria,
      -- Category: impute with the most frequent category.
      CASE
      WHEN categoria IS NULL THEN (
            SELECT categoria
            FROM inversionistas
            GROUP BY categoria
            ORDER BY COUNT(*) DESC
            LIMIT 1)
      ELSE categoria
END AS categoria_imput
FROM inversionistas;
```