```
-- PART 1: INITIAL DATABASE SCAN
-- ================================================

-- Connect to the database
psql -U postgres -d nba_db

-- Check data types
\d players

-- Check the number of records
SELECT COUNT(*) FROM players;

-- Initial look at the table
SELECT * FROM players;

-- Perform missing data management (if applicable)
SELECT * FROM players
WHERE player_name IS NULL OR
salary IS NULL OR
position IS NULL OR
age IS NULL OR
games_played IS NULL OR
field_goals_pctg IS NULL OR
three_pt_pctg IS NULL OR
two_pt_pctg IS NULL OR
free_throws_pctg IS NULL OR
points_per_game IS NULL;

-- In total we have 34 records with incomplete data
-- We will delete these records and store the result in a
-- new table (players_cleaned) inside the database
```

```
-- CTE to preserve full rows
WITH datos_completos AS (
      SELECT * FROM players
      WHERE player_name IS NOT NULL AND
            salary IS NOT NULL AND
            position IS NOT NULL AND
            age IS NOT NULL AND
            games_played IS NOT NULL AND
            field_goals_pctg IS NOT NULL AND
            three_pt_pctg IS NOT NULL AND
            two_pt_pctg IS NOT NULL AND
            free_throws_pctg IS NOT NULL AND
            points_per_game IS NOT NULL
)
-- And store the result in the new table "players_cleaned"
SELECT * INTO players_cleaned FROM datos_completos;

-- Verify that we have a new table
\d

-- And display the number of records in the original table and the new
table
-- (467 vs. 433)
SELECT COUNT(*) FROM players;
SELECT COUNT(*) FROM players_cleaned;

-- PART 2: UNIVARIATE ANALYSIS OF NUMERICAL VARIABLES
-- ====================================================

-- Measures of central tendency and dispersion: mean, median, standard
deviation
```

```
-- and interquartile range of numerical variables


-- Let's look at these measures for salaries, for example.
WITH medidas_salarios AS (
      SELECT
            AVG(salary) AS mu,
            STDDEV(salary) AS s,
            PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY salary) AS mediana,
            PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY salary) -
            PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY salary) AS iqr
      FROM players_cleaned
)
SELECT * FROM medidas_salarios;


-- It's easier to interpret the above if the salaries are represented in
millions of dollars.


-- Let's start by creating the "salary_m" column.
ALTER TABLE players_cleaned
ADD COLUMN salary_m NUMERIC;


-- Update the column with the calculated values.
UPDATE players_cleaned
SET salary_m = ROUND(salary / 1000000, 2);


-- And verify that the table has been updated correctly.
SELECT salary, salary_M FROM players_cleaned ORDER BY salary_m DESC;


-- And now let's look at the salary measures of interest and add
-- rounding to the mean and standard deviation
WITH medidas_salarios AS (
      SELECT
```

```
            ROUND(AVG(salary_M),1) AS mu,
            ROUND(STDDEV(salary_M),1) AS s,
            PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY salary_M) AS
            mediana,
            PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY salary_M) -
            PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY salary_M) AS iqr
      FROM players_cleaned
)
SELECT * FROM medidas_salarios;


-- And let's note some observations:
-- There are large differences between mean vs. median and standard
-- deviation vs. IQR, which indicates the presence of "outliers": salaries
-- that are either very low or very high relative to the normal range.


-- Let's look at the salary distribution in a little more detail. Let's
create a CTE that calculates the minimum, Q1, Q2, Q3, and maximum salaries.
WITH rangos_salarios AS (
      SELECT
            MIN(salary_m) as min,
            PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY salary_m) AS q1,
            PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY salary_m) AS q2,
            PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY salary_m) AS q3,
            MAX(salary_m) as max
      FROM players_cleaned
)
SELECT * FROM rangos_salarios;


-- Observations:
-- - 75% of players have salaries ranging from $0.01 million to $11.22
--   million
-- - The remaining 25% have salaries ranging from $11.22 million to $48.07
--   million
-- - There is indeed a skew in the salary distribution
```

```sql
-- Let's now perform an analysis on the ages
WITH rangos_edades AS (
      SELECT
            MIN(age) as min,
            PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY age) AS q1,
            PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY age) AS q2,
            PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY age) AS q3,
            MAX(age) as max
      FROM players_cleaned
)
SELECT * FROM rangos_edades;

-- Observations:
-- - 75% of the players are between 19 and 29 years old, and
--   the remaining 25% are between 29 and 42 years old.

-- PART 3: UNIVARIATE ANALYSIS OF CATEGORICAL VARIABLES
-- ====================================================

-- Let's analyze the categorical variable "position." Let's first look at
-- the different positions

SELECT DISTINCT(position) FROM players_cleaned;

-- Notes:
-- We have 9 different player positions:
-- - PG: point guard
-- - PF: power forward
-- - SF-SG: small forward - shooting guard
-- - PG-SG: point guard - shooting guard
```

```
-- - C: center

-- - SG: shooting guard

-- - SG-PG: shooting guard - point guard

-- - SF: small forward

-- - SF-PF: small forward - power forward


-- And let's see, for example, how many players there are at each position
-- in the database.

SELECT position, COUNT(*) as nro_jugadores

FROM players_cleaned

GROUP BY position

ORDER BY nro_jugadores DESC;
```