keys_to_database

```
-- Count the number of null values in the ticker column
SELECT count(*) - COUNT(ticker) AS missing
 FROM fortune500;
```

Explanation:

- This SQL query calculates the number of null values in the ticker column of the fortune500 table. COUNT(*) counts all rows, while COUNT(ticker) counts only rows where the ticker column is not NULL. Subtracting the latter from the former gives the count of rows with NULL values in the ticker column. The result is aliased as missing.

```
SELECT company.name
-- Table(s) to select from
FROM company
INNER JOIN fortune500
ON fortune500.ticker = company.ticker;
```

Explanation:

- This SQL query retrieves the names of companies from a company table that are also listed in a fortune500 table. It does this using an INNER JOIN, which only returns rows where the ticker symbol (presumably a unique company identifier) matches in both tables. The result will only include companies present in both datasets.

```
-- Count the number of tags with each type
SELECT type, COUNT(type) AS count
 FROM tag_type
 -- To get the count for each type, what do you need to do?
 GROUP BY type
 -- Order the results with the most common tag types listed first
 ORDER BY count DESC; --Corrected the ORDER BY clause
```

Explanation:

- This SQL query counts the occurrences of each unique type in the tag_type table. It uses GROUP BY to group rows with the same type together, and COUNT(type) to count the number of rows in each group. The result is then ordered in descending order based on the count, showing the most frequent tag types first. The original ORDER BY type DESC was incorrect; it sorted alphabetically by type, not by frequency. The corrected version uses ORDER BY count DESC.

```sql
-- Select the 3 columns desired
SELECT company.name, tag_type.tag, tag_type.type
 FROM company
    -- Join to the tag_company table
    INNER JOIN tag_company
    ON company.id = tag_company.company_id
    -- Join to the tag_type table
    INNER JOIN tag_type
    ON tag_company.tag= tag_type.tag
 -- Filter to most common type
 WHERE type='cloud';
```

Explanation:

- This SQL query retrieves the name of companies associated with the ‘cloud’ tag. It performs joins across three tables (company, tag_company, and tag_type) to link company names with their respective tags and tag types. The WHERE clause filters the results to show only companies tagged with type = 'cloud'.

```sql
-- Use coalesce
SELECT coalesce(industry, sector, 'Unknown') AS industry2,
    -- Don't forget to count!
    count(*)
 FROM fortune500
-- Group by what? (What are you counting by?)
 GROUP BY industry2
-- Order results to see most common first
 ORDER BY count DESC
-- Limit results to get just the one value you want
 LIMIT 1;
```

Explanation:

- This SQL query finds the most frequent industry in the fortune500 table. If industry is null, it uses sector instead; otherwise, it uses ‘Unknown’ if both are null. It then groups the results by this combined industry field (industry2), counts the occurrences of each, orders the results by count in descending order, and finally limits the output to only show the single most frequent industry.