

Clean Data: Write a query to clean the data and remove the currency sign before the price. Ensure to preserve the values without the currency sign in a new column.

Example Response

Solution 1

– Create a new column 'price_cleaned' without the currency sign

```
ALTER TABLE properties
```

```
ADD COLUMN price_cleaned DECIMAL(10, 2);
```

– Update the 'price_cleaned' column with values without the currency sign

```
UPDATE properties
```

```
SET price_cleaned = REPLACE(price, '$', '');
```

– Check the result

```
SELECT * FROM properties;
```

Top Earning Properties: Write a query to find how much the top 10 properties from the company list are earning.

Example Response

Solution 1

– Find the top 3 earning properties

```
SELECT property_name, SUM(CAST(REPLACE(price, '$', '') AS DECIMAL(10, 2))) AS total_earnings
```

```
FROM properties
```

```
GROUP BY property_name
```

```
ORDER BY total_earnings DESC
```

```
LIMIT 3;
```

Solution 2

```
SELECT property_id, property_name, city, price, company_name
```

```
FROM properties
```

```
ORDER BY CAST(REPLACE(price, '$', '') AS DECIMAL) DESC
```

```
LIMIT 3;
```

Family-Friendly Properties: Write a query to retrieve properties suitable for families with amenities like spacious rooms and a kid-friendly environment.

Example Response

Solution 1

– Retrieve properties suitable for families

```

SELECT property_name, city, company_name
FROM properties
WHERE property_id IN (
    SELECT property_id
    FROM reviews
    WHERE review_text LIKE '%family%' OR review_text LIKE '%kid-friendly%'
);

```

Solution 2

```

SELECT p.property_id, p.property_name, p.city, p.price, p.company_name
FROM properties p
WHERE p.property_id IN (
    SELECT r.property_id
    FROM reviews r
    WHERE LOWER(r.review_text) LIKE '%family%'
    OR LOWER(r.review_text) LIKE '%kid%'
)
ORDER BY p.property_id;

```

Most Expensive Property: Create a stored procedure to fetch the details of the most expensive property in each city.

Example Response

– Retrieve the details of the most expensive property in each city

Solution 1

– Using Stored Procedure

```

CREATE PROCEDURE GetMostExpensivePropertyPerCity()
BEGIN
    SELECT p.*
    FROM properties p
    JOIN (
        SELECT city, MAX(price) AS max_price
        FROM properties
        GROUP BY city
    ) AS max_prices
    ON p.city = max_prices.city AND p.price = max_prices.max_price;
END

```

Solution 2

– Without Stored Procedure

```

SELECT p.property_id, p.property_name, p.city, p.price, p.company_name
FROM properties p
JOIN (
    SELECT city, MAX(CAST(REPLACE(price, '$', '' ) AS DECIMAL(10, 2))) AS max_price
    FROM properties
    GROUP BY city
) max_prices ON p.city = max_prices.city AND CAST(REPLACE(p.price, '$', '' ) AS DECIMAL(10, 2)) =
max_prices.max_price;

```

Lowest Rated Property: Write a query to find the property with the lowest average review score.

Example Response

Solution 1

– Find the property with the lowest average review score

```
SELECT r.property_id, p.property_name, AVG(score) AS avg_score
FROM reviews r
JOIN properties p ON r.property_id = p.property_id
GROUP BY r.property_id
ORDER BY avg_score ASC
LIMIT 1;
```