Basic Queries

1. **Query all data in the `pets` table:**

   ```sql
   SELECT * FROM pets;
   ```

2. **Query the first 5 rows of the `pets` table:**

   ```sql
   SELECT * FROM pets
   LIMIT 5;
   ```

3. **Query names and ages of pets:**

   ```sql
   SELECT name, age
   FROM pets;
   ```

4. **Query pets sorted by age (youngest to oldest):**

   ```sql
   SELECT *
   FROM pets
   ORDER BY age ASC;
   ```

5. **Query pets sorted alphabetically by name:**

   ```sql
   SELECT *
   FROM pets
   ORDER BY name ASC;
   ```

6. **Query all male pets:**

   ```sql
   SELECT *
   FROM pets
   WHERE sex = 'Male';
   ```

7. **Query all cats:**

   ```sql
   SELECT *
   FROM pets
   WHERE species = 'Cat';
   ```

8. **Query pets at least 5 years old:**

   ```sql
   SELECT *
   FROM pets
   WHERE age >= 5;
   ```

9. **Query male dogs (excluding sex and species columns):**

   ```sql
   SELECT name, age
   FROM pets
   WHERE sex = 'Male' AND species = 'Dog';
   ```

10. **Query names of dogs younger than 5 years old:**

    ```sql
    SELECT name
    FROM pets
    ```

```
WHERE species = 'Dog' AND age < 5;
```

11. **Query pets that are male dogs or female cats:**

```
SELECT *
FROM pets
WHERE (sex = 'Male' AND species = 'Dog') OR (sex = 'Female' AND species = 'Cat');
```

12. **Query the 5 oldest pets:**

```
SELECT *
FROM pets
ORDER BY age DESC
LIMIT 5;
```

13. **Query names and ages of female cats, sorted by age descending:**

```
SELECT name, age
FROM pets
WHERE sex = 'Female' AND species = 'Cat'
ORDER BY age DESC;
```

14. **Query pets whose names start with 'P':**

```
SELECT *
FROM pets
WHERE name LIKE 'P%';
```

Working with `employees_null`

15. **Query employees with missing salary:**

```
SELECT *
FROM employees_null
WHERE salary IS NULL;
```

16. **Query employees with salary below $35,000 or missing:**

```
SELECT *
FROM employees_null
WHERE salary < 35000 OR salary IS NULL;
```

17. **Query employees with missing job titles:**

```
SELECT *
FROM employees_null
WHERE job_title IS NULL;
```

Working with `employees`

18. **Query the newest and most senior employees:**

```
SELECT *
FROM employees
```

```
ORDER BY startdate DESC
LIMIT 1;

SELECT *
FROM employees
ORDER BY startdate ASC
LIMIT 1;
```

19. **Query employees named Thomas:**

```
SELECT *
FROM employees
WHERE firstname = 'Thomas';
```

20. **Query employees named Thomas or Shannon:**

```
SELECT *
FROM employees
WHERE firstname IN ('Thomas', 'Shannon');
```

21. **Query employees named Robert, Lisa, or names starting with J, excluding sales:**

```
SELECT *
FROM employees
WHERE (firstname IN ('Robert', 'Lisa') OR firstname LIKE 'J%')
AND job_title != 'Sales';
```

Column Operations

22. **Query the top 5 rows of `employees`:**

```
SELECT *
FROM employees
LIMIT 5;
```

23. **Query `employees` with salary converted to Euros:**

```
SELECT firstname, lastname, salary / 1.1 AS salary_eur
FROM employees;
```

24. **Query with `salary_eur` renamed as `salary_eu`:**

```
SELECT firstname, lastname, salary / 1.1 AS salary_eu
FROM employees;
```

25. **Combine `firstname` and `lastname` into `fullname` and round salary:**

```
SELECT
    firstname || ', ' || lastname AS fullname,
    ROUND(salary / 1.1, 2) AS salary_eu
FROM employees;
```

26. **Replace `startdate` with `startyear` using `substr()`:**

```
SELECT
    firstname || ', ' || lastname AS fullname,
    ROUND(salary / 1.1, 2) AS salary_eu,
    SUBSTR(startdate, 1, 4) AS startyear
FROM employees;
```

27. **Use `strftime()` instead of `substr()` for `startyear`:**

```
SELECT
    firstname || ', ' || lastname AS fullname,
    ROUND(salary / 1.1, 2) AS salary_eu,
    STRFTIME('%Y', startdate) AS startyear
FROM employees;
```

28. **Format salary as USD with dollar sign and comma separators:**

```
SELECT
    firstname || ', ' || lastname AS fullname,
    STRFTIME('%Y', startdate) AS startyear,
    PRINTF('$%,.0f', salary) AS salary
FROM employees;
```

29. **Add a `bonus` column for salespeople eligible last year:**

```
SELECT
    firstname,
    lastname,
    CASE
        WHEN job_title = 'Salesperson' THEN 'yes'
        ELSE 'no'
    END AS bonus
FROM employees;
```

30. **Add a `bonus` column for salespeople with salary >= $100,000:**

```
SELECT
    firstname,
    lastname,
    CASE
        WHEN job_title = 'Salesperson' AND salary >= 100000 THEN 'yes'
        ELSE 'no'
    END AS bonus
FROM employees;
```

31. **Create a `target_comp` column for next year's bonus structure:**

```
SELECT
    firstname,
    lastname,
    PRINTF('$%,.0f', salary) AS salary,
    CASE
        WHEN job_title = 'Salesperson' AND salary > 100000 THEN PRINTF('$%,.0f', salary * 1.1)
        WHEN job_title = 'Salesperson' AND salary <= 100000 THEN PRINTF('$%,.0f', salary * 1.05)
        WHEN job_title = 'Administrator' THEN PRINTF('$%,.0f', salary * 1.05)
        ELSE PRINTF('$%,.0f', salary)
    END AS target_comp
FROM employees;
```