Aggregations on `pets` Table

32. **Count the number of rows in the `pets` table:**

```sql
SELECT COUNT(*) AS total_pets
FROM pets;
```

33. **Count the number of female pets:**

```sql
SELECT COUNT(*) AS female_pets
FROM pets
WHERE sex = 'Female';
```

34. **Count the number of female cats:**

```sql
SELECT COUNT(*) AS female_cats
FROM pets
WHERE sex = 'Female' AND species = 'Cat';
```

35. **Calculate the mean age of pets:**

```sql
SELECT AVG(age) AS mean_age
FROM pets;
```

36. **Calculate the mean age of dogs:**

```sql
SELECT AVG(age) AS mean_dog_age
FROM pets
WHERE species = 'Dog';
```

37. **Calculate the mean age of male dogs:**

```sql
SELECT AVG(age) AS mean_male_dog_age
FROM pets
WHERE species = 'Dog' AND sex = 'Male';
```

38. **Get count, mean, min, and max age of pets:**

```sql
SELECT
    COUNT(*) AS total_pets,
    AVG(age) AS mean_age,
    MIN(age) AS min_age,
    MAX(age) AS max_age
FROM pets;
```

39. **Repeat the above with rounded averages and readable labels:**

```sql
SELECT
    COUNT(*) AS "Total Pets",
    ROUND(AVG(age), 1) AS "Average Age",
    MIN(age) AS "Youngest Age",
    MAX(age) AS "Oldest Age"
FROM pets;
```

Null Values in `employees_null` Table

40. **Count rows with missing salaries:**

```sql
SELECT COUNT(*) AS missing_salaries
FROM employees_null
WHERE salary IS NULL;
```

41. **Count salespeople with non-missing salaries:**

```sql
SELECT COUNT(*) AS salespeople_with_salaries
FROM employees_null
WHERE job_title = 'Salesperson' AND salary IS NOT NULL;
```

Aggregations on `employees` Table

42. **Calculate mean salary for employees who joined after 2010:**

```sql
SELECT AVG(salary) AS mean_salary_after_2010
FROM employees
WHERE CAST(SUBSTR(startdate, 1, 4) AS INTEGER) > 2010;
```

43. **Calculate mean salary in Swiss Francs (CHF):**

```sql
SELECT AVG(salary * 0.97) AS mean_salary_chf
FROM employees;
```

44. **Calculate mean salary in USD and CHF:**

```sql
SELECT
    PRINTF('$%,.0f', AVG(salary)) AS "Mean Salary in USD",
    PRINTF('%.0f Fr.', AVG(salary * 0.97)) AS "Mean Salary in CHF"
FROM employees;
```

Grouping and Aggregations

45. **Calculate average age of pets by species:**

```sql
SELECT species, AVG(age) AS avg_age
FROM pets
GROUP BY species;
```

46. **Repeat the above with readable labels:**

```sql
SELECT species AS "Species", AVG(age) AS "Average Age"
FROM pets
GROUP BY species;
```

47. **Get count, mean, min, and max age by species:**

```sql
SELECT
    species,
    COUNT(*) AS total_pets,
    AVG(age) AS avg_age,
    MIN(age) AS min_age,
    MAX(age) AS max_age
```

```
FROM pets
GROUP BY species;
```

48. **Show mean salaries by job title:**

```
SELECT job_title, AVG(salary) AS avg_salary
FROM employees
GROUP BY job_title;
```

49. **Show mean salaries by job title in New Zealand Dollars (NZD):**

```
SELECT
    job_title,
    AVG(salary * 1.65) AS avg_salary_nzd
FROM employees
GROUP BY job_title;
```

50. **Show count, mean, min, and max salaries by job title:**

```
SELECT
    job_title,
    COUNT(*) AS num_employees,
    AVG(salary) AS avg_salary,
    MIN(salary) AS min_salary,
    MAX(salary) AS max_salary
FROM employees
GROUP BY job_title;
```

51. **Show mean salaries by job title, sorted descending:**

```
SELECT
    job_title,
    AVG(salary) AS avg_salary
FROM employees
GROUP BY job_title
ORDER BY avg_salary DESC;
```

Frequent Names

52. **Top 5 most common first names:**

```
SELECT
    firstname,
    COUNT(*) AS name_count
FROM employees
GROUP BY firstname
ORDER BY name_count DESC
LIMIT 5;
```

53. **Show first names with exactly 2 occurrences:**

```
SELECT firstname
FROM employees
GROUP BY firstname
HAVING COUNT(*) = 2;
```

Transactions Table

54. **Preview the `transactions` table:**

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

55. **Top 5 largest orders by number of items:**

```sql
SELECT
    order_id,
    customer_id,
    SUM(quantity) AS total_items
FROM transactions
GROUP BY order_id, customer_id
ORDER BY total_items DESC
LIMIT 5;
```

56. **Total cost of each transaction:**

```sql
SELECT
    order_id,
    SUM(unit_price * quantity) AS total_cost
FROM transactions
GROUP BY order_id;
```

57. **Top 5 transactions by total cost:**

```sql
SELECT
    order_id,
    SUM(unit_price * quantity) AS total_cost
FROM transactions
GROUP BY order_id
ORDER BY total_cost DESC
LIMIT 5;
```

58. **Top 5 customers by total revenue:**

```sql
SELECT
    customer_id,
    SUM(unit_price * quantity) AS total_revenue
FROM transactions
GROUP BY customer_id
ORDER BY total_revenue DESC
LIMIT 5;
```

59. **Top 5 employees by revenue generated:**

```sql
SELECT
    employee_id,
    SUM(unit_price * quantity) AS total_revenue
FROM transactions
GROUP BY employee_id
ORDER BY total_revenue DESC
LIMIT 5;
```

60. **Customer who worked with the most employees:**

```sql
SELECT
    customer_id,
    COUNT(DISTINCT employee_id) AS num_employees
FROM transactions
GROUP BY customer_id
```

```sql
ORDER BY num_employees DESC
LIMIT 1;
```

61. **Customers with total business over $80,000:**

```sql
SELECT
    customer_id,
    SUM(unit_price * quantity) AS total_revenue
FROM transactions
GROUP BY customer_id
HAVING total_revenue > 80000;
```