SELECT

strftime('%Y', date) AS Year,

strftime('%m', date) AS Month,

AVG(open) AS avg\_open,

AVG(high) AS avg\_high,

AVG(low) AS avg\_low,

AVG(close) AS avg\_close,

SUM(volume) AS total\_volume

FROM yum

GROUP BY Year, Month

ORDER BY Year ASC, Month ASC;

CREATE VIEW yum\_by\_month AS

SELECT

strftime('%Y', date) AS Year,

strftime('%m', date) AS Month,

AVG(open) AS avg\_open,

AVG(high) AS avg\_high,

AVG(low) AS avg\_low,

AVG(close) AS avg\_close,

SUM(volume) AS total\_volume

FROM yum

GROUP BY Year, Month

ORDER BY Year ASC, Month ASC;

CREATE VIEW trans\_by\_month AS

SELECT

strftime('%Y', date) AS Year,

strftime('%m', date) AS Month,

SUM(sales) AS total\_sales

FROM transactions

GROUP BY Year, Month;

CREATE VIEW trans\_by\_employee AS

SELECT

employee\_id,

SUM(sales) AS total\_sales

FROM transactions

GROUP BY employee\_id;

WITH pet\_initials AS (

SELECT LOWER(SUBSTR(name, 1, 1)) AS first\_initial

FROM pets

)

SELECT first\_initial, COUNT(\*) AS count

FROM pet\_initials

GROUP BY first\_initial

ORDER BY count DESC

LIMIT 1;

WITH employee\_info AS (

SELECT

firstname || ' ' || lastname AS name,

job,

salary,

strftime('%Y', hire\_date) AS year

FROM employees

)

SELECT

name || ' started in ' || year || ' and makes $' ||

printf("%.0f", salary) || ' working in ' ||

CASE

WHEN job = 'IT' THEN job

ELSE LOWER(job)

END || '.'

FROM employee\_info;

WITH company\_types AS (

SELECT

CASE

WHEN INSTR(company\_name, 'LLC') > 0 THEN 'LLC'

WHEN INSTR(company\_name, 'Inc') > 0 THEN 'Inc'

WHEN INSTR(company\_name, 'Ltd') > 0 THEN 'Ltd'

WHEN INSTR(company\_name, 'PLC') > 0 THEN 'PLC'

ELSE 'Other'

END AS company\_type,

sales

FROM transactions

)

SELECT

company\_type,

SUM(sales) AS total\_revenue,

COUNT(\*) AS transaction\_count

FROM company\_types

GROUP BY company\_type;

SELECT

employees.firstname, employees.lastname, transactions.sales

FROM employees

JOIN transactions ON employees.employee\_id = transactions.employee\_id;

SELECT

employees.firstname, employees.lastname, SUM(transactions.sales) AS total\_sales

FROM employees

JOIN transactions ON employees.employee\_id = transactions.employee\_id

GROUP BY employees.employee\_id

ORDER BY total\_sales DESC

LIMIT 1;

SELECT

employees.firstname, employees.lastname, trans\_by\_employee.total\_sales

FROM employees

JOIN trans\_by\_employee ON employees.employee\_id = trans\_by\_employee.employee\_id

ORDER BY trans\_by\_employee.total\_sales DESC

LIMIT 1;

WITH employee\_sales AS (

SELECT

employee\_id,

SUM(sales) AS total\_sales

FROM transactions

GROUP BY employee\_id

)

SELECT

employees.firstname, employees.lastname, employee\_sales.total\_sales

FROM employees

JOIN employee\_sales ON employees.employee\_id = employee\_sales.employee\_id

ORDER BY employee\_sales.total\_sales DESC

LIMIT 1;

SELECT

employees.firstname, employees.lastname, SUM(transactions.sales) AS total\_sales, employees.salary

FROM employees

JOIN transactions ON employees.employee\_id = transactions.employee\_id

GROUP BY employees.employee\_id

HAVING total\_sales > 1.5 \* employees.salary;

SELECT

transactions.transaction\_id, transactions.date, transactions.sales, employees.firstname, employees.lastname

FROM transactions

JOIN employees ON transactions.employee\_id = employees.employee\_id

WHERE transactions.date < employees.hire\_date;

SELECT

strftime('%Y', transactions.date) AS year,

strftime('%m', transactions.date) AS month,

SUM(transactions.sales) AS company\_revenue,

SUM(yum.volume) AS yum\_trade\_volume

FROM transactions

JOIN yum ON strftime('%Y-%m', transactions.date) = strftime('%Y-%m', yum.date)

GROUP BY year, month;

SELECT

strftime('%Y', transactions.date) AS year,

strftime('%m', transactions.date) AS month,

SUM(transactions.sales) AS company\_revenue,

SUM(yum.volume) AS yum\_trade\_volume,

MIN(yum.low) AS lowest\_price,

MAX(yum.high) AS highest\_price

FROM transactions

JOIN yum ON strftime('%Y-%m', transactions.date) = strftime('%Y-%m', yum.date)

GROUP BY year, month;