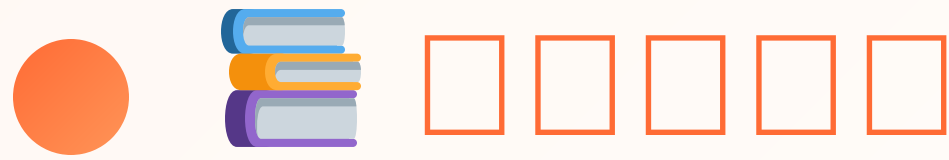


2. SELECT 문

SELECT





1.

SQL - 2

2. SELECT语句

```
-- 基本语法
SELECT column1, column2 FROM 'table';

-- 从CSV文件中读取数据
SELECT customer_id, product_id, quantity FROM 'data/sales.csv';

-- 使用AS关键字为列命名
SELECT customer_name AS name FROM 'data/customers.csv';
```

3.

,

AS OK

4. DuckDB□□□□□□□□

□□□□□□□□□□□□□□

□□□□□□□□□□□□

□□□□□□□□□□□□□□



1 sales.csv ID

```
SELECT product_id, quantity FROM 'data/sales.csv';
```

Ctrl/Cmd + Enter

2

15

customer_id, order_date

练习 2 products.csv

```
SELECT product_name, price FROM 'data/products.csv';
```



练习 2

练习 2

练习 2

练习 2

3 customers.csv

```
SELECT customer_name AS FROM 'data/customers.csv';
```

□ □ □ □ □ □

```
SELECT
  customer_id AS 顧客ID,
  customer_name AS 顧客名,
  email AS 顧客Eメール
FROM 'data/customers.csv';
```



□ □ □ □ □

□ □ □ □ □ □ □ □ □ □

□ □ □ □ □ □ □ □ □ □ □ □ □ □

□ □ □ □ □ □ □ □ □ □ □ □ □



1. □□□□□□□□□□

```
-- □□□□□□□□□□  
SELECT order_date, customer_id, quantity  
FROM 'data/sales.csv'  
LIMIT 5;
```

2. □□□□2□□□□□□□□

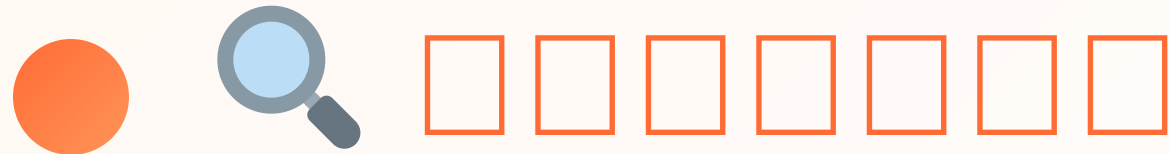
```
SELECT price, product_name, price  
FROM 'data/products.csv';
```

→ price□2□□□□□□□□□□

3. 3월 4일 판매한 상품

```
SELECT
  product_name AS 상품명,
  price AS 가격,
  category AS 카테고리
FROM 'data/products.csv'
ORDER BY price DESC;
```

ORDER BY 4월 4일 판매한 상품



1

```
--   
SELECT  
  order_date AS   
  product_id AS   
  quantity AS   
FROM 'data/sales.csv'  
LIMIT 10;
```

2

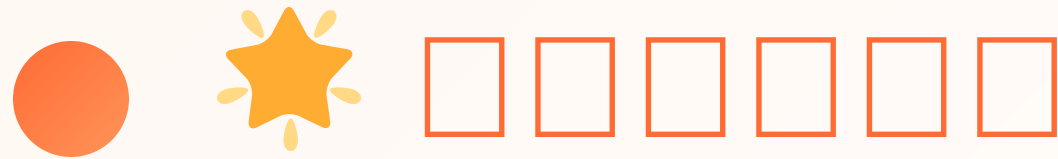
```
-- 查詢客戶資料
SELECT
  customer_name AS 客戶名稱,
  email AS 電子郵件
FROM 'data/customers.csv';
```




客戶資料

客戶資料可以匯出Excel表格

客戶資料可以匯出CSV檔案



```
--   
SELECT customername FROM 'data/customers.csv';
```

 column "customername" does not exist
→ `customer_name`

□ □ □ □ □ □ □ □ □ □ □ □ □ □

```
SELECT customer_id, phone FROM 'data/customers.csv';
```

 phone□□□□□□□□□□□□□□□□



SELECT col1, col2 FROM ...



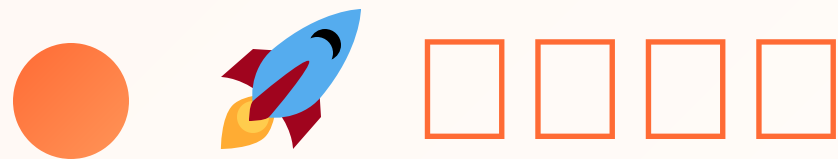
AS



□ □ □ □ □ □ □ □

```
-- 1번 쿼리 실행
SELECT customer_id, customer_name
FROM 'data/customers.csv';

-- 2번 쿼리 실행
SELECT
    customer_id AS ID,
    customer_name AS NAME
FROM 'data/customers.csv';
```



3 WHERE
[Empty boxes for text input]



1

```
-- 1. customers.csv ID
-- 
-- 2. products.csv
-- 
-- 3. sales.csv
-- 
```

2

```
-- 
```

```
SELECT
```

```
customer_id AS __, -- 
```

```
customer_name AS __, -- 
```

```
address AS __ -- 
```

```
FROM 'data/customers.csv';
```

□□2□□□□

```
-- □□□□□□□□□□  
SELECT  
    product_name AS □□□,  
    ___ AS □□□□,    -- price□□□□□□□□  
    ___ AS □□□□    -- category□□□□□□□□  
FROM 'data/products.csv';
```


3

```
-- 1. 테이블의 ID를  
--  
  
-- 2. 테이블의  
--
```

□□4□□□□□□□□□□□□

□□□□□□□□□□□□□□□□

```
-- □□□1□□□□□□  
SELECT customer_id customer_name FROM 'data/customers.csv';  
  
-- □□□2□□□□□□□□  
SELECT id, name FROM 'data/customers.csv';  
  
-- □□□3□AS□□□□□□□□  
SELECT customer_name □□□ AS FROM 'data/customers.csv';
```



```
-- sales.csv
-- 2024 1
--
```



□ □ □ □ □ □ □ □ □ □ □ □ □ □ □

```
-- sales.csv□products.csv□□□□□□□□□□JOIN□□□□□□  
-- □□□□□□□□□□□□□□□□□□□□□□□□□□□□  
-- □□□□□□□□□□□□□□□□
```

● ? FAQ

Q:

A: DuckDB `customer_id` `CUSTOMER_ID`

Q: AS

A: AS

Q:

A: CSV