**FIFO ANALYTICS**

**Khái niệm :**

**First In First Out**

**Inventory Management**

**Inventory Analysis**

**Inventory Age**

**Practice :** [**Link**](https://www.notion.so/Inventory-Report-SQL-Test-aca2d92feb1f4a81af123cdc31bb332e)

# "Inventory Report" SQL Test

# Overview

In this code test you are expected to create 3 inventory reports using SQL query (MySQL v8.0).

Database schema and sample data is provided below.

The entire test should take around 1 - 2 hours to complete.

# Database Schema & Test Data

The database contains 3 tables, which contains stock movement records and customer data.

### Table 1: Movement Table

The most important table. It contains the stock movement history of all warehouses, data including the item SKU, quantity changed, closing inventory balance and date.

Every movement record should link to a document.

### Table 2: Document Table

There are two types of document, purchase and sales\_order.

Some document might connected to a customer, but it is optional.

### Table 3: Customer Table

Include customer ID and email address.

## Schema SQL and Sample Data

The following SQL should create all the tables needed and populate it with sample data. You can copy and paste them to <https://db-fiddle.com/> to start the test.

**Please do not modify the provided sample data in your final result.**

CREATE TABLE movement (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

document\_id INT,

warehouse VARCHAR(100),

sku VARCHAR(100),

quantity INT,

balance INT,

created\_at DATETIME

);

CREATE TABLE document (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

customer\_id INT,

type TEXT

);

CREATE TABLE customer (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

contact VARCHAR(100)

);

INSERT INTO movement (document\_id, sku, warehouse, quantity, balance, created\_at)

VALUES

(1, 'iPhone', 'HK', 10, 10, '2021-1-1'),

(1, 'iPod', 'HK', 3, 3, '2021-1-1'),

(2, 'iPod', 'HK', -1, 2, '2021-1-2'),

(3, 'iPod', 'HK', -2, 0, '2021-1-3'),

(4, 'iPod', 'HK', 5, 5, '2021-2-1'),

(5, 'iMac', 'US', 5, 5, '2021-2-1'),

(5, 'iPhone', 'US', 2, 2, '2021-2-1'),

(6, 'iMac', 'HK', 5, 5, '2021-2-2'),

(7, 'iPod', 'HK', -4, 1, '2021-2-8'),

(8, 'iMac', 'HK', -1, 4, '2021-2-9'),

(9, 'iPhone', 'US', -1, 1, '2021-2-17'),

(10, 'iMac', 'HK', 1, 5, '2021-3-2'),

(11, 'iMac', 'HK', -1, 4, '2021-3-8'),

(11, 'iPod', 'HK', -1, 0, '2021-3-8'),

(12, 'iMac', 'US', 5, 10, '2021-3-10');

INSERT INTO document (id, customer\_id, type)

VALUES

(1, NULL, "purchase"),

(2, 1, 'sales\_order'),

(3, 2, 'sales\_order'),

(4, NULL, 'purchase'),

(5, NULL, 'purchase'),

(6, NULL, 'purchase'),

(7, 3, 'sales\_order'),

(8, 1, 'sales\_order'),

(9, 4, 'sales\_order'),

(10, NULL, 'purchase'),

(11, NULL, 'sales\_order'),

(12, NULL, 'purchase'),

(13, NULL, 'purchase');

INSERT INTO customer (id, contact)

VALUES

(1, 'boris0407@gmail.com'),

(2, 'candywong@gmail.com'),

(3, 'flora2002@gmail.com'),

(4, 'glory@gmail.com'),

(5, 'himsonfong@gmail.com');

## Sample Data Preview

Below is what the data looks like when they are combined into one big table.

warehouse | created\_at | sku | quantity | balance | document.type | customer.contact |

----------|---------------------|--------|----------|---------|---------------|---------------------|

HK | 2021-01-01 00:00:00 | iPhone | 10 | 10 | purchase | NULL |

HK | 2021-01-01 00:00:00 | iPod | 3 | 3 | purchase | NULL |

HK | 2021-01-02 00:00:00 | iPod | -1 | 2 | sales\_order | boris0407@gmail.com |

HK | 2021-01-03 00:00:00 | iPod | -2 | 0 | sales\_order | candywong@gmail.com |

HK | 2021-02-01 00:00:00 | iPod | 5 | 5 | purchase | NULL |

US | 2021-02-01 00:00:00 | iMac | 5 | 5 | purchase | NULL |

US | 2021-02-01 00:00:00 | iPhone | 2 | 2 | purchase | NULL |

HK | 2021-02-02 00:00:00 | iMac | 5 | 5 | purchase | NULL |

HK | 2021-02-08 00:00:00 | iPod | -4 | 1 | sales\_order | flora2002@gmail.com |

HK | 2021-02-09 00:00:00 | iMac | -1 | 4 | sales\_order | boris0407@gmail.com |

US | 2021-02-17 00:00:00 | iPhone | -1 | 1 | sales\_order | glory@gmail.com |

HK | 2021-03-02 00:00:00 | iMac | 1 | 5 | purchase | NULL |

HK | 2021-03-08 00:00:00 | iMac | -1 | 4 | sales\_order | NULL |

HK | 2021-03-08 00:00:00 | iPod | -1 | 0 | sales\_order | NULL |

US | 2021-03-10 00:00:00 | iMac | 5 | 10 | purchase | NULL |

# Questions

## Question 1: Customer Leaderboard

Rank customers by quantity they purchased.

Include the customer's email address (shown as "guest" if not provided) and quantity they purchased in the report.

### Example Result

contact | sold |

--------------------|------|

flora2002@gmail.com | 4 |

boris0407@gmail.com | 2 |

...

## Question 2: Inventory Snapshot

Write a SQL query to return HK warehouse's stock of any given time.

Define a variable for the time so user can change it easily. For example: SET @date = '2021-4-1 00:00:00'.

### Example Result

For a query of the latest quantity in a HK warehouse, your SQL query should return the following rows.

At date '2021-4-1 00:00:00' , HK warehouse should contain the following items:

warehouse | sku | balance |

----------|--------|---------|

HK | iMac | 4 |

HK | iPhone | 10 |

HK | iPod | 0 |

At a different date '2021-2-1 00:00:00' , HK warehouse should contain the following items:

warehouse | sku | balance |

----------|--------|---------|

HK | iPhone | 10 |

HK | iPod | 0 |

## Question 3: Age of Inventory (Advanced level)

Show the age of the available stocks of a given time, and group the quantity by age "0-30 days", "31-60 days", "61 - 90 days" and "90 days+".

* Age of an stock means the number of day after it enters the warehouse.
* Stocks in a warehouse comes and goes. When deducting stock please follow the "First In First Out" rule, meaning oldest stock will be deducted first.

### Example Result

At 2021-4-1 00:00:00 in HK warehouse, the report should shows there a total 4 iMac in the warehouse, where 1 of them purchased recently ("Age 0-30" has quantity 1)

SQL SERVER CODE

CREATE TABLE movement (  
 id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,  
 document\_id INT,  
 warehouse VARCHAR(100),  
 sku VARCHAR(100),  
 quantity INT,  
 balance INT,  
 created\_at DATETIME  
);  
  
CREATE TABLE document (  
 id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,  
 customer\_id INT,  
 type TEXT  
);  
  
CREATE TABLE customer (  
 id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,  
 contact VARCHAR(100)  
);  
  
INSERT INTO movement (document\_id, sku, warehouse, quantity, balance, created\_at)  
VALUES  
 (1, 'iPhone', 'HK', 10, 10, '2021-1-1'),  
 (1, 'iPod', 'HK', 3, 3, '2021-1-1'),  
 (2, 'iPod', 'HK', -1, 2, '2021-1-2'),  
 (3, 'iPod', 'HK', -2, 0, '2021-1-3'),  
 (4, 'iPod', 'HK', 5, 5, '2021-2-1'),  
 (5, 'iMac', 'US', 5, 5, '2021-2-1'),  
 (5, 'iPhone', 'US', 2, 2, '2021-2-1'),  
 (6, 'iMac', 'HK', 5, 5, '2021-2-2'),  
 (7, 'iPod', 'HK', -4, 1, '2021-2-8'),  
 (8, 'iMac', 'HK', -1, 4, '2021-2-9'),  
 (9, 'iPhone', 'US', -1, 1, '2021-2-17'),  
 (10, 'iMac', 'HK', 1, 5, '2021-3-2'),  
 (11, 'iMac', 'HK', -1, 4, '2021-3-8'),  
 (11, 'iPod', 'HK', -1, 0, '2021-3-8'),  
 (12, 'iMac', 'US', 5, 10, '2021-3-10');  
  
INSERT INTO document (id, customer\_id, type)  
VALUES  
 (1, NULL, 'purchase'),  
 (2, 1, 'sales\_order'),  
 (3, 2, 'sales\_order'),  
 (4, NULL, 'purchase'),  
 (5, NULL, 'purchase'),  
 (6, NULL, 'purchase'),  
 (7, 3, 'sales\_order'),  
 (8, 1, 'sales\_order'),  
 (9, 4, 'sales\_order'),  
 (10, NULL, 'purchase'),  
 (11, NULL, 'sales\_order'),  
 (12, NULL, 'purchase'),  
 (13, NULL, 'purchase');  
  
INSERT INTO customer (id, contact)  
VALUES  
 (1, 'boris0407@gmail.com'),  
 (2, 'candywong@gmail.com'),  
 (3, 'flora2002@gmail.com'),  
 (4, 'glory@gmail.com'),  
 (5, 'himsonfong@gmail.com');  
  
SET IDENTITY\_INSERT customer off  
  
SET IDENTITY\_INSERT document off

Solutions

[DB Fiddle - SQL Database Playground (db-fiddle.com)](https://www.db-fiddle.com/f/5JcnDxzVhyLeKaFi1QWoME/11)