LAB ASSIGNMENT 11

Exercise 1(1 mark)

Import the file **sample_database.sql** (1 mark). Double check if you have the database with tables as listed below:

Hint: in this exercise, you may use just phpMyAdmin to work with.

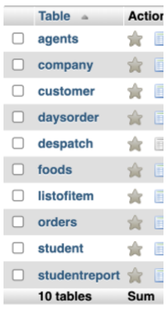


Table	Action
agents	
company	
customer	
daysorder	
despatch	
foods	
listofitem	
orders	
student	
studentreport	
10 tables	Sum

Figure 1 **List of expected table after import**

Exercise 2 (3 marks)

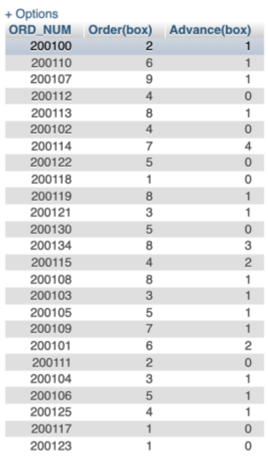
For the company to ship products we will pack them in to boxes. Each box has a capacity of 500 units. Create SQL User Defined Function **Box(amount, box_size)** to get total boxes to be used for each order.

Run the select command given below:

```
SELECT ORD_NUM, Box(ORD_AMOUNT, 500) as `Order(box)`,Box(ADVANCE_AMOUNT, 500) as `Advance(box)` FROM `orders`
```

The output should resemble Figure 2.
Save the code to create function into **ex2_box.sql** (3 marks)

Hint: use export menu in phpMyAdmin and copy just the function definition part.



+ Options

ORD_NUM	Order(box)	Advance(box)
200100	2	1
200110	6	1
200107	9	1
200112	4	0
200113	8	1
200102	4	0
200114	7	4
200122	5	0
200118	1	0
200119	8	1
200121	3	1
200130	5	0
200134	8	3
200115	4	2
200108	8	1
200103	3	1
200105	5	1
200109	7	1
200101	6	2
200111	2	0
200104	3	1
200106	5	1
200125	4	1
200117	1	0
200123	1	0

Figure 2 **List of expected boxes**

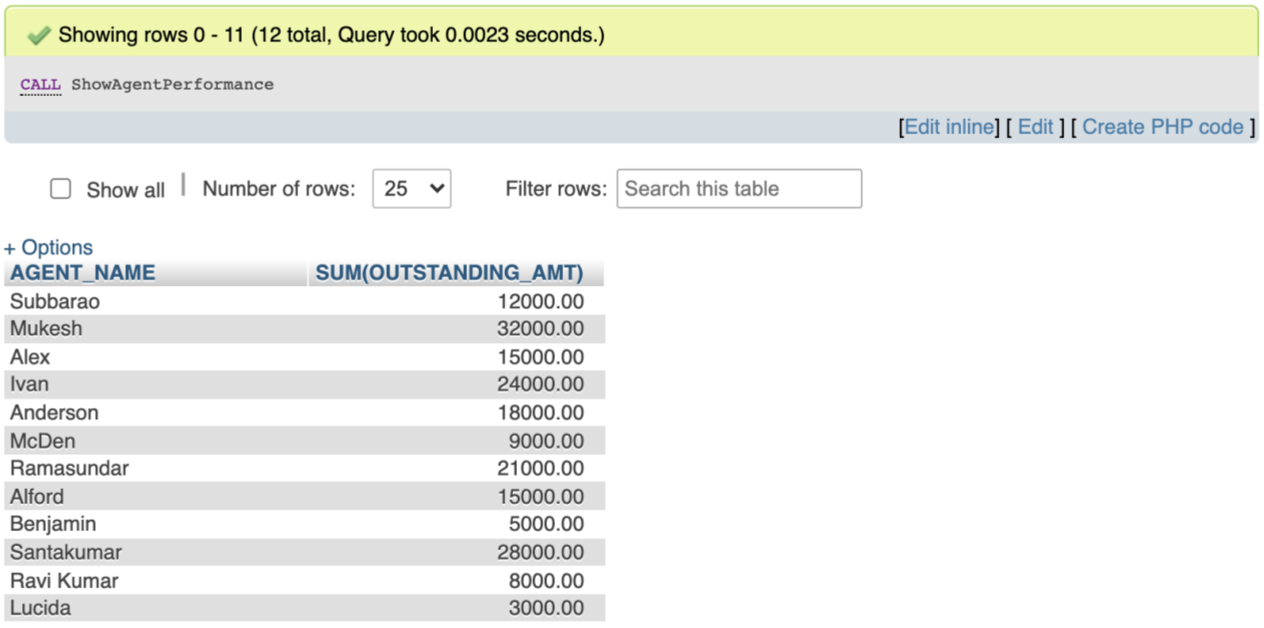
Exercise 3 (3 marks)

Create a Store Procedure **ShowAgentPerformance** to show the performance of agents.

Run the select command given below:

```
CALL ShowAgentPerformance
```

The output should resemble Figure 3.
Save the code to create the store procedure into **ex3_agent_performance.sql** (3 marks)



Showing rows 0 - 11 (12 total, Query took 0.0023 seconds.)

```
CALL ShowAgentPerformance
```

[Edit inline] [Edit] [Create PHP code]
Show all
Number of rows: 25
Filter rows: Search this table
+ Options

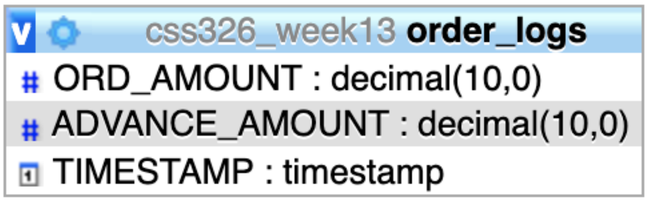
AGENT_NAME	SUM(OUTSTANDING_AMT)
Subbarao	12000.00
Mukesh	32000.00
Alex	15000.00
Ivan	24000.00
Anderson	18000.00
McDen	9000.00
Ramasundar	21000.00
Alford	15000.00
Benjamin	5000.00
Santakumar	28000.00
Ravi Kumar	8000.00
Lucida	3000.00

Figure 3 expected agent performance

Exercise 4 (3 marks)

We want to make sure that no body temper with our **orders** table, so we decided to create the audit table and track all the change of data of the table.

1. Create a table **order_logs** with the following schema (1 mark):



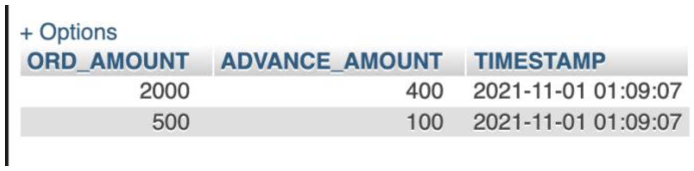
css326_week13 **order_logs**
ORD_AMOUNT : decimal(10,0)
ADVANCE_AMOUNT : decimal(10,0)
TIMESTAMP : timestamp

2. Create a trigger **order_changed** on **orders** updated by recording old ORD_AMOUNT and old ADVANCE_AMOUNT together with timestamp (2 marks)

The expected result when agent **A007** try to update the order into his favor with the following SQL

```
UPDATE orders SET ORD_AMOUNT=10000,ADVANCE_AMOUNT=5000 WHERE AGENT_CODE = 'A007'
```

the trigger should be trigged and save the amount before change into **order_logs** as shown in Figure 4



+ Options

ORD_AMOUNT	ADVANCE_AMOUNT	TIMESTAMP
2000	400	2021-11-01 01:09:07
500	100	2021-11-01 01:09:07

Figure 4 **expected result of monthly sale**

Save a script of (1) & (2) into file **ex4_audit_trigger.sql**

(Hint: You need to find out how to insert the current timestamp in Google or previous lecture)