

BITS Pilani, K.K. Birla Goa Campus
DBS Lab 05 (Solution) Full Mark: 15

1. Display the total amount of payment made in every month. Your results should display month in word (Eg. JANUARY, FEBRUARY ...) and the total amount paid in that month. [2]

Solution:

```
mysql> SELECT SUM(AMOUNT),CASE WHEN DATE LIKE '%-01-%' THEN 'JANUARY' WHEN DATE LIKE '%-02-%' THEN 'FEBRUARY' WHEN DATE LIKE '%-03-%' THEN 'MARCH' WHEN DATE LIKE '%-04-%' THEN 'APRIL' WHEN DATE LIKE '%-05-%' THEN 'MAY' WHEN DATE LIKE '%-06-%' THEN 'JUNE' WHEN DATE LIKE '%-07-%' THEN 'JULY' WHEN DATE LIKE '%-08-%' THEN 'AUGUST' WHEN DATE LIKE '%-10-%' THEN 'OCTOBER' WHEN DATE LIKE '%-11-%' THEN 'NOVEMBER' WHEN DATE LIKE '%-12-%' THEN 'DECEMBER' ELSE 'INVALID' END AS MONTH from PAYMENT GROUP BY MONTH;
```

```
+-----+-----+
| SUM(AMOUNT) | MONTH |
+-----+-----+
|      8000   | APRIL |
|    2080428  | JANUARY |
|     10000   | MARCH |
|    1053714  | OCTOBER |
+-----+-----+
4 rows in set, 11 warnings (0.06 sec)
```

2. Print YES if female customer is a depositor, otherwise print NO. (Your result should print NAME of the female customer and the remark as YES or NO). [2]

Solution:

```
mysql> SELECT NAME, IF(C_ID IN (SELECT DISTINCT C.C_ID FROM (SELECT CUST_ID FROM DEPOSITOR) D JOIN CUSTOMER C ON D.CUST_ID=C.C_ID), 'YES', 'NO') REMARK FROM CUSTOMER WHERE GENDER='F';
```

```
+-----+-----+
| NAME      | REMARK |
+-----+-----+
| haritha   | YES    |
| himani    | YES    |
| saritha   | YES    |
| sachitra  | YES    |
| yogitha   | YES    |
| radhika   | NO     |
+-----+-----+
6 rows in set (0.00 sec)
```

3. Create a view which contains Branch Name, the Total amount in that branch and its rank based on the total amount. (For eg. If the total amount in the branch is >= 10000 then its Rank is 1. If it is >=8000 and <10000, then rank is 2. Otherwise rank is 3. [2]

Solution:

```
mysql> CREATE VIEW ZONE_RANK_INFO AS select BR_NAME,sum(BALANCE) as sum,CASE WHEN sum(BALANCE)>=10000 THEN 1 WHEN sum(BALANCE)<10000 AND sum(BALANCE)>=8000 THEN 2 ELSE 3 END AS RANK_ZONE from ACCOUNT group by BR_NAME order by sum(BALANCE) DESC;
mysql> SELECT * FROM ZONE_RANK_INFO;
```

BR_NAME	sum	RANK_ZONE
zone1	13000	1
zone4	10000	1
zone2	10000	1
zone7	9000	2
zone8	8000	2
zone9	7000	3
zone6	6000	3
zone3	6000	3

8 rows in set (0.00 sec)

4. Print as the following condition satisfies for the time that passed between two payments for the same payment number. [3]

If the time is within 1 year: It should print 'FIRST YEAR', if it is within 1-2 year then it should print 'SECOND YEAR', if it is within 2-3 year then it should print 'THIRD YEAR', otherwise print 'DEFAULTER'.

Hint: **DATEDIFF(date1, date2)**–gives difference between the dates in days.

Solution:

```
mysql> SELECT S1.P_NO,S1.DATE,S2.DATE,DATEDIFF(S2.DATE,S1.DATE) AS DDIFF, CASE WHEN
DATEDIFF(S2.DATE,S1.DATE)/365<1 THEN 'FIRST YEAR' WHEN DATEDIFF(S2.DATE,S1.DATE)/365<2
AND DATEDIFF(S2.DATE,S1.DATE)/365>1 THEN 'SECOND YEAR' WHEN
DATEDIFF(S2.DATE,S1.DATE)/365<3 AND DATEDIFF(S2.DATE,S1.DATE)/365>2 THEN 'THIRD YEAR'
ELSE 'DEFAULTER' END AS FLAG FROM PAYMENT AS S1,PAYMENT AS S2 WHERE S1.P_NO=S2.P_NO
AND S2.DATE>S1.DATE ORDER BY FLAG;
```

P_NO	DATE	DATE	DDIFF	FLAG
p2	2000-01-09	2011-10-09	4291	DEFAULTER
p2	2000-01-09	2012-03-11	4445	DEFAULTER
p1	2011-01-09	2011-10-08	272	FIRST YEAR
p1	2011-10-08	2011-10-11	3	FIRST YEAR
p2	2011-10-09	2012-03-11	154	FIRST YEAR
p1	2011-03-11	2011-10-08	211	FIRST YEAR
p1	2011-03-11	2011-10-11	214	FIRST YEAR
p1	2011-01-09	2011-03-11	61	FIRST YEAR
p1	2011-10-08	2012-04-11	186	FIRST YEAR
p1	2012-04-11	2012-10-25	197	FIRST YEAR
p1	2011-01-09	2011-10-11	275	FIRST YEAR
p1	2011-10-11	2012-04-11	183	FIRST YEAR
p1	2011-10-08	2012-10-25	383	SECOND YEAR
p1	2011-03-11	2012-04-11	397	SECOND YEAR
p1	2011-10-11	2012-10-25	380	SECOND YEAR
p1	2011-01-09	2012-04-11	458	SECOND YEAR
p1	2011-03-11	2012-10-25	594	SECOND YEAR
p1	2011-01-09	2012-10-25	655	SECOND YEAR

18 rows in set (0.00 sec)

5. Update the customer street information in a new table called CUSTOMER_COPY to 'NH17B' for the customers who have made payments of more than 10000 for ALL of their loans. [3]

Instruction:

First create a CUSTOMER_COPY table using the following query:

```
mysql> CREATE TABLE CUSTOMER_COPY LIKE CUSTOMER;
```

Copy all data from CUSTOMER table to CUSTOMER_COPY:

```
mysql> INSERT INTO CUSTOMER_COPY SELECT * FROM CUSTOMER;
```

Now write the update query as:

```
mysql> UPDATE CUSTOMER_COPY SET STREET='NH17B' WHERE NAME IN (YOUR QUERY FOR THE PROBLEM);
```

After the above query display the contents of CUSTOMER_COPY table.

Solution:

```
mysql> UPDATE CUSTOMER_COPY SET STREET='NH17B' WHERE NAME IN (SELECT DISTINCT C.NAME FROM CUSTOMER C JOIN (SELECT SUM(P.AMOUNT) AS SUM,B.CUST_ID AS C_ID FROM BORROWER B JOIN PAYMENT P ON P.L_NO = B.LOAN_NO GROUP BY B.CUST_ID HAVING SUM >= 10000) R ON R.C_ID = C.C_ID);
```

Now CUSTOMER_COPY contains:

```
mysql> SELECT * FROM CUSTOMER_COPY;
```

C_ID	NAME	STREET	CITY	GENDER
c1	haritha	NH17B	machlipatnam	f
c2	himani	NH17B	machlipatnam	f
c3	shankar	kadamba	eluru	m
c4	saritha	kadamba	eluru	f
c5	sachitra	saraswathi	nandigama	f
c6	yogitha	kaveri	hyderabad	f
c7	radhika	kaveri	hyderabad	f
c8	ramu	azad	Vijayawada	m

6.) Find the total outstanding loan amount for each customer. If the total outstanding amount is even display 'EVEN' else display 'ODD'. [3]

Format: (5 columns: Cust_id, amount_borrowed, amount_paid, outstanding_amount, remark).

Hint: IFNULL function lets you specify a value if the expression evaluates to NULL.

Usage: IFNULL(variable, 0)

Solution:

```
mysql> SELECT T1.CUST_ID,T1.AMOUNT_BORROWED,T2.AMOUNT_PAID,(AMOUNT_BORROWED-AMOUNT_PAID) AS OUTSTANDING_AMOUNT,IF((AMOUNT_BORROWED-AMOUNT_PAID)%2=0,'EVEN','ODD') AS RESULT FROM (SELECT B.CUST_ID,SUM(AMOUNT) AS AMOUNT_BORROWED FROM BORROWER B JOIN LOAN L ON L.LN_NO=B.LOAN_NO GROUP BY CUST_ID) T1 NATURAL JOIN (SELECT B.CUST_ID,IFNULL(SUM(T1.AMOUNT_PAID),0) AS AMOUNT_PAID FROM BORROWER B LEFT JOIN (SELECT L_NO,SUM(AMOUNT) AS AMOUNT_PAID FROM PAYMENT GROUP BY L_NO) AS T1 ON T1.L_NO=B.LOAN_NO GROUP BY B.CUST_ID) T2;
```

CUST_ID	AMOUNT_BORROWED	AMOUNT_PAID	OUTSTANDING_AMOUNT	RESULT
c1	7000000	2088428	4911572	EVEN
c2	2500000	1048714	1451286	EVEN
c3	4000000	5000	3995000	EVEN
c4	4000000	5000	3995000	EVEN
c5	22000000	0	22000000	EVEN
c6	500000	0	500000	EVEN
c7	500000	0	500000	EVEN
c8	1000000	0	1000000	EVEN

8 rows in set (0.00 sec)

```
mysql> select 'Next Thursday (April 21) is the Lab Test. [10% of Total Mark in DBS]'
Lab_Test_Notice from dual;
```

Lab_Test_Notice
Next Thursday (April 21) is the Lab Test. [10% of Total Mark in DBS]

1 row in set (0.00 sec)