Lab 5
Database Systems

General Instructions:

- **All questions are compulsory.**
- **Each question carries 10 marks each.**
- **Copy the query and the answer in a text file before evaluation.**
- **All the best!**

Q.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.

Q.2) Print as the following condition satisfies for the time that passed between two payments for the same payment number.

- If the time between payments is within 1 year: Flag should be assigned 'FIRST YEAR'.
- If the time between payments is between 1^{st} and 2^{nd} year, then Flag should be assigned 'SECONDYEAR'.
- If time between payments is between 2^{nd} and 3^{rd} year, then flag should be assigned “THIRD YEAR”.
- Otherwise assign flag = “Defaulter”.

You must display the Payment Number, first date of payment, second date of payment, date difference and Flag.
You may use the following hint.

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

Q.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, If it is >=7000 and <8000, then rank is 3 Otherwise rank is 4.

Note: Create a table ACCOUNT_COPY by the following commands. Do not try fancy things on your own as it might change the database and cost you marks for the upcoming labs.

```
DROP TABLE IF EXISTS `ACCOUNT_COPY`;

CREATE TABLE `ACCOUNT_COPY` (
  `ACC_NO` varchar(20) NOT NULL DEFAULT '',
  `BALANCE` decimal(10,0) DEFAULT NULL,
  `BR_NAME` varchar(20) NOT NULL,
  PRIMARY KEY (`ACC_NO`)) ENGINE=InnoDB DEFAULT
CHARSET=latin1;

LOCK TABLES `ACCOUNT_COPY` WRITE;

INSERT INTO `ACCOUNT_COPY` VALUES ('20012023094',
4000,'zone1'),('20012023095',4000,'zone1'),('20012023096',
5000,'zone1'),('20012023097',10000,'zone2'),('20012023098',
10000,'zone4'),('20012023046',6000,'zone6'),('20012033046',
6000,'zone3'),('20112033046',7000,'zone9'),('20112023046',
8000,'zone8'),('20012023047',9000,'zone7');
/*!40000 ALTER TABLE `ACCOUNT_COPY` ENABLE KEYS */;
UNLOCK TABLES;
```

Q.4) Run the following queries one by one.

```
DROP TABLE IF EXISTS ORDERS;

 CREATE TABLE ORDERS (ORD_ID INT, AMOUNT FLOAT,
ORD_DATE DATE, CUSTOMERID INT, SALESMANID INT,
PRIMARY KEY(ORD_ID));
```

```
INSERT INTO ORDERS VALUES (1, 270.65,'2012-09-10', 3001,
5005),(2, 150.5,'2012-10-05',3005,5002) ,(3,65.26,'2012-10-05',
3002,5001) ,(4,110.5,'2012-08-17',3009,5003) ,(5,948.5,'2012-09-10',
3005,5002) ,(6,2400.6,'2012-07-27',3007,5001) ,(7,5760,'2012-09-10',
3002,5001) ,(8,1983.43,'2012-10-10',3004,5006) ,
(9,2480.4,'2012-10-10',3009,5003),(10,250.45,'      2012-06-27',
3008,5002),(11,75.29,'2012-08-17', 3003, 5007),
(12,3045.6,'2012-04-25',3002,5001);

DROP TABLE IF EXISTS SALESMAN;

CREATE TABLE SALESMAN(SALESMANID INT, NAME
VARCHAR(10), PRIMARY KEY(SALESMANID));

 INSERT INTO SALESMAN VALUES (5001, 'JAMES'),(5002,
'NAIL'),(5005, 'PIT'),(5006, 'LYON'),(5003, 'LAUSON'),(5007,
'PAUL');
```

QUESTION – part a) You have to create a view called GOODSALESMAN that finds the salesman who has the customer with the highest order of a day. Display salesmanid, customerid and amount. Part b) You have to create a view GREATSALESMAN that finds the salesman who has the customer with the highest order of a day at least 3 times overall. Display salesmanid.