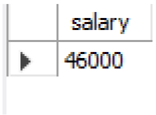
KARTHIKEYAN

1. Find the nth maximum salary from the employee table using correlated subquery.

```
SELECT DISTINCT salary
FROM employee_tbl e1
WHERE 0 = (
  SELECT COUNT(DISTINCT salary)
  FROM employee_tbl e2
  WHERE e2.salary > e1.salary
);
```

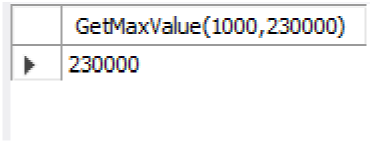


```
salary
46000
```

2.Create a function which takes 2 numbers as input and return the maximum value.

```
delimiter //
create function GetMaxValue(a int ,b int)
returns int
DETERMINISTIC
BEGIN
if (a>b) then
return a;
else
return b;
end if;
end //
delimiter ;
```

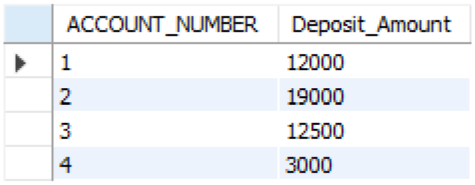
```
select GetMaxValue(1000,230000);
```



```
GetMaxValue(1000,230000)
230000
```

3.Write a query to display account number and total amount deposited by each account holder (Including the opening balance). Give the total amount deposited an alias name of Deposit_Amount. Display the records in sorted order based on account number.- Use the tables created in the previous handson.

```
SELECT
   A.ACCOUNT_NUMBER,
   (IFNULL(A.OPENING_BALANCE, 0) +
    IFNULL(SUM(
      CASE
         WHEN TD.TRANSACTION_TYPE = 'Deposit' THEN TD.TRANSACTION_AMOUNT
         ELSE 0
      END
   ), 0)) AS Deposit_Amount
FROM
   Account A
LEFT JOIN
   Transaction_Details TD
ON
   A.ACCOUNT_NUMBER = TD.ACCOUNT_NUMBER
GROUP BY
   A.ACCOUNT_NUMBER, A.OPENING_BALANCE
ORDER BY
   A.ACCOUNT_NUMBER;
```

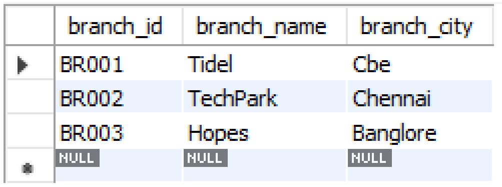


ACCOUNT_NUMBER	Deposit_Amount
1	12000
2	19000
3	12500
4	3000

4.Create table branch_master with columns
   branch_id VARCHAR(6) -primary key
   branch_name VARCHAR(30)
   branch_city VARCHAR(30)
and Insert values into branch_master .

```
create table branch_master
(
         branch_id VARCHAR(6) primary key ,
   branch_name VARCHAR(30),
   branch_city VARCHAR(30)
   );
```

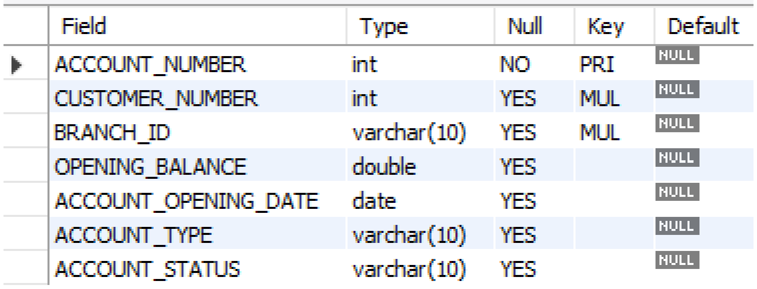
```
insert into branch_master values
  ('BR001','Tidel','Cbe'),
   ('BR002','TechPark','Chennai'),
    ('BR003','Hopes','Banglore')
  ;
```



branch_id	branch_name	branch_city
BR001	Tidel	Cbe
BR002	TechPark	Chennai
BR003	Hopes	Banglore
NULL	NULL	NULL

5. Add column branch_id in accounts_master and refer as foreign key to branch_id of branch_master.

```
ALTER TABLE Account
ADD CONSTRAINT FK_Account_Branch
FOREIGN KEY (BRANCH_ID) REFERENCES Branch_Master(BRANCH_ID);
```



Field	Type	Null	Key	Default
ACCOUNT_NUMBER	int	NO	PRI	NULL
CUSTOMER_NUMBER	int	YES	MUL	NULL
BRANCH_ID	varchar(10)	YES	MUL	NULL
OPENING_BALANCE	double	YES		NULL
ACCOUNT_OPENING_DATE	date	YES		NULL
ACCOUNT_TYPE	varchar(10)	YES		NULL
ACCOUNT_STATUS	varchar(10)	YES		NULL