

emp_id INT

);

University Institute of Engineering Department of Computer Science & Engineering

EXPERIMENT: 3 NAME: SARTHAK UID: 23BCS11978 **BRANCH: BE-CSE** SECTION/GROUP: KRG 2A SEMESTER: 5TH SUBJECT CODE: 23CSP-339 **SUBJECT NAME: ADBMS** 1. Aim Of The Practical: [EASY] You are given with employee relation having only the attributes named as EMP_ID. Your task is to fetch the maximum EMP ID, But Excluding the duplicate values. [EASY] Find id name and description of product which has not been sold for once. [EASY] Find the total quantity sold with respect to each product name [LEETCODE 1890] Write a solution to report the latest login for all users in the year 2020. Do not include the users who did not login in 2020. 2. Tools Used: SQL Server Management Studio 3. Code : ----- EASY-----**CREATE TABLE Employee (**

```
INSERT INTO Employee (emp id)
VALUES (1), (2), (3), (4), (5), (5), (6), (7), (8), (9), (9), (3);
select max(emp id)
from Employee
where emp_id not in
(select emp id
from Employee
group by emp_id
having count(emp id)>1)
----- EASY-----
CREATE TABLE TBL PRODUCTS
ID INT PRIMARY KEY IDENTITY,
[NAME] NVARCHAR(50),
[DESCRIPTION] NVARCHAR(250)
CREATE TABLE TBL_PRODUCTSALES
ID INT PRIMARY KEY IDENTITY,
PRODUCTID INT FOREIGN KEY REFERENCES TBL PRODUCTS (ID),
UNITPRICE INT,
QUALTITYSOLD INT
INSERT INTO TBL PRODUCTS VALUES ('TV', '52 INCH BLACK COLOR LCD TV')
INSERT INTO TBL PRODUCTS VALUES ('LAPTOP', 'VERY THIIN BLACK COLOR ACER LAPTOP')
INSERT INTO TBL PRODUCTS VALUES ('DESKTOP', 'HP HIGH PERFORMANCE DESKTOP')
INSERT INTO TBL PRODUCTSALES VALUES (3,450,5)
INSERT INTO TBL PRODUCTSALES VALUES (2,250,7)
INSERT INTO TBL PRODUCTSALES VALUES (3,450,4)
INSERT INTO TBL PRODUCTSALES VALUES (3,450,9)
SELECT *FROM TBL PRODUCTS
SELECT *FROM TBL PRODUCTSALES
select ID,[NAME],[DESCRIPTION] from TBL PRODUCTS
where ID not in
(select distinct PRODUCTID from TBL PRODUCTSALES)
----- EASY-----
select P.[NAME],
(select SUM(S.QUALTITYSOLD) from TBL PRODUCTSALES S where S.PRODUCTID = P.ID)
as [QTY SOLD]
from TBL PRODUCTS P;
```

----- LEETCODE-----

SELECT user_id, MAX(time_stamp) as 'last_stamp' FROM Logins

WHERE YEAR(time_stamp)='2020' GROUP BY user_id

4. Output:

[EASY]

	emp_id
1	1
2	2
3	3
4	4
5	5
6	5
7	6
8	7
9	8
10	9
11	9
12	3



[EASY]

	ID	NAME	DESCRIPTION
1	1	TV	52 INCH BLACK COLOR LCD TV
2	2	LAPTOP	VERY THIIN BLACK COLOR ACER LAPTOP
3	3	DESKTOP	HP HIGH PERFORMANCE DESKTOP

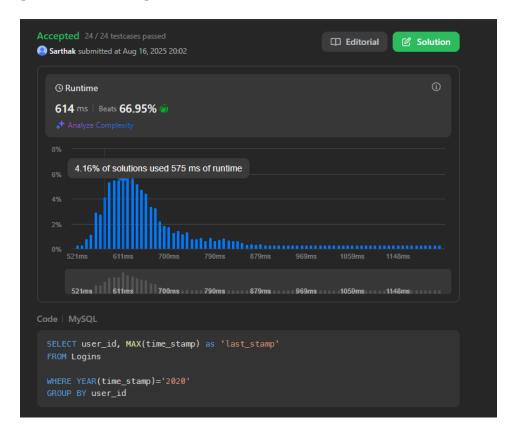
	ID	PRODUCTID	UNITPRICE	QUALTITYSOLD
1	1	3	450	5
2	2	2	250	7
3	3	3	450	4
4	4	3	450	9

	ID	NAME	DESCRIPTION
1	1	TV	52 INCH BLACK COLOR LCD TV

[EASY]

	NAME	QTY_SOLD
1	TV	NULL
2	LAPTOP	7
3	DESKTOP	18

[LEETCODE 1890]



5. Learning Outcomes:

- Understand and implement subqueries to model hierarchical relationships within a single table (e.g., finding employees who report to a specific manager using a subquery).
- Construct relational subqueries to fetch meaningful information such as employee—manager relationships, including handling cases where no manager exists by using correlated subqueries.
- **Design and populate tables** using CREATE TABLE and INSERT INTO statements, and practice writing subqueries for real-world hierarchical and timeseries data scenarios.
- Use subqueries in place of multi-table joins to retrieve and match data across datasets, such as comparing actual vs. requested values (e.g., NPV values for specific years).
- Handle missing data in subqueries using functions like ISNULL() or COALESCE() to substitute default values when queried values are not found.
- Apply conditional subqueries with multiple criteria (e.g., filtering based on both ID and YEAR inside a subquery) to ensure accurate data mapping.
- **Develop problem-solving approaches using subqueries** (both correlated and non-correlated) to derive insights from HR records and financial datasets in enterprise applications.