SQL HANDSON

WeCP

UST
Trivandrum

PROBLEM 1

You want to find the details of students based on their MARKS.

Write a SQL query to display details of all students who have scored MARKS within the range **400 and 6000** except those whose MARKS are **1200 and 5236**.

Schema

```
CREATE TABLE STUDENTS(
  STUDENT_ID INT PRIMARY KEY,
  STUDENT_NAME VARCHAR(50),
  MARKS INT,
  CITY VARCHAR(50)
)
```

Table structure

STUDENTS

Name	Type	Description
STUDENT_ID	INT	Column denoting STUDENT_ID representing id of the student
STUDENT_NAME	VARCHAR(50)	Column denoting STUDENT_NAME representing name of the student
MARKS	INT	Column denoting MARKS representing marks scored by the student
CITY	VARCHAR(50)	Column denoting CITY representing city

Name	Type	Description
		in which student resides

Sample testcase 1
Input

STUDENTS

STUDENT_ID	STUDENT_NAME	MARKS	CITY
101	Rahul	1000	Pune
102	Shyam	4500	Pune
103	Priya	90	Agra
495	Cheese	5236	Pune
12	Jam	34895	Mumbai
104	Rahull	100	Pune
105	Shyamm	450	Pune
106	Rahulll	1200	Pune
107	Shyammm	4505	Pune

Output

101	Rahul	1000	Pune
102	Shyam	4500	Pune
105	Shyamm	450	Pune
107	Shyammm	4505	Pune

```
SELECT
    STUDENT_ID,
    STUDENT_NAME,
    MARKS,
    CITY
FROM STUDENTS
WHERE (MARKS BETWEEN 400 AND 6000) AND (MARKS!=1200 AND MARKS!=5236)
```

PROBLEM 2

You want to do a product price analysis based on the manufacturing date.

Write a SQL query to display the details of all products for which the **PROD_PRICE** is greater than at least one of the products manufactured on **31st July 2018**.

Schema

```
CREATE TABLE PRODUCT(
  PRO_NO int(10) PRIMARY KEY,
  PROD_NAME VARCHAR(50),
  MANU_DATE DATE,
  PROD_PRICE int(10),
  SALES_ID int(10)
)
```

Table structure

PRODUCT

Name	Type	Description
PRO_NO	int(10)	Column denoting PRO_NO representing product number
PROD_NAME	VARCHAR(50)	Column denoting PROD_NAME representing the name of the product
MANU_DATE	DATE	Column denoting MANU_DATE representing the date on which the product was manufactured
PROD_PRICE	int(10)	Column denoting PROD_PRICE representing the price of the product
SALES_ID	int(10)	Column denoting SALES_ID representing id of a salesman selling the product

Sample testcase 1
Input

PRODUCT

PRO_NO	PROD_NAME	MANU_DATE	PROD_PRICE	SALES_ID
101	Cake	2020-10-11	1000	34

PRO_NO	PROD_NAME	MANU_DATE	PROD_PRICE	SALES_ID
102	Veg Lollipop	2018-07-31	4500	15
103	Biscuit	1990-10-11	90	34
104	Bu	2018-07-31	450	15

Output

101	Cake	2020-10-11	1000	34
102	Veg Lollipop	2018-07-31	4500	15

Sample testcase 2

Input

PRODUCT

PRO_NO	PROD_NAME	MANU_DATE	PROD_PRICE	SALES_ID
34	Milk	1998-10-07	300	23
45	Butter	2020-10-11	56	11
77	Ghee	2018-07-31	78	27

Output

34	Milk	1998-10-07	300	23

```
-- Enter your query here
SELECT * FROM PRODUCT
WHERE PROD_PRICE>
```

```
(SELECT MIN(PROD_PRICE) FROM PRODUCT
 WHERE MANU_DATE='2018-07-31');
```

PROBLEM 3

You want to display product details based on manufacturing company.

Write a SQL query to display the **PROD_NAME**, **PROD_PRICE** and **COM_NAME** for the most expensive product of each company only if they have **COM_ID** which is present in both the given tables.

Schema

```
CREATE TABLE COMPANY(COM_ID int(10) PRIMARY KEY, COM_NAME VARCHAR(50))

CREATE TABLE PRODUCT(
  PROD_ID int(10) PRIMARY KEY,
  PROD_NAME VARCHAR(50),
  PROD_PRICE int(10),
  COM_ID int(10)
)
```

Table structure

COMPANY

Name	Type	Description
COM_ID	int(10)	Column denoting COM_ID representing id of the company.
COM_NAME	VARCHAR(50)	Column denoting COM_NAME representing the name of the company.

PRODUCT

Name	Type	Description
PROD_ID	int(10)	Column denoting PROD_ID representing id of the product.
PROD_NAME	VARCHAR(50)	Column denoting PROD_NAME representing name of the product.
PROD_PRICE	int(10)	Column denoting PROD_PRICE representing price of the product.
COM_ID	int(10)	Column denoting COM_ID representing the id of the company.

Sample testcase 1
Input

COMPANY

COM_ID	COM_NAME
34	Rahul
23	Divya
15	Kiya

PRODUCT

PROD_ID	PROD_NAME	PROD_PRICE	COM_ID
101	Cake	1	37
102	Veg Lollipop	14	15
105	Dummy	23	15
103	Biscuit	9	34

Output

Dummy	23	Kiya
Biscuit	9	Rahul

```
-- Enter your query here
SELECT P.PROD_NAME,P.PROD_PRICE,C.COM_NAME
FROM PRODUCT P
JOIN COMPANY C
ON P.COM_ID=C.COM_ID
WHERE P.PROD_PRICE=
(SELECT MAX(PROD_PRICE) FROM PRODUCT P1
WHERE P1.COM_ID=P.COM_ID);

-- SELECT P.PROD_NAME,P.PROD_PRICE,C.COM_NAME
-- FROM PRODUCT P
-- INNER JOIN COMPANY C
-- ON P.COM_ID=C.COM_ID
-- WHERE P.PROD_PRICE=(
--     SELECT MAX(PROD_PRICE)
--     FROM PRODUCT P1
--     WHERE P1.COM_ID=P.COM_ID
-- )
```

PROBLEM 4

You want to do a comparative analysis of employee salaries across all departments.

Write a SQL query to display the **EMP_NAME**, **EMP_SALARY** and **DEPT_ID** of those employees whose EMP_SALARY is greater than or equal to the EMP_SALARY of the employee whose EMP_NO is **equal to 103**.

Schema

```
CREATE TABLE EMPLOYEE(
  EMP_NO int(10) PRIMARY KEY,
  EMP_NAME VARCHAR(50),
  HIRE_DATE DATE,
  EMP_SALARY int(10),
  DEPT_ID int(10)
)
```

Table structure

EMPLOYEE

Name	Type	Description
EMP_NO	int(10)	Column denoting EMP_NO representing Employee number
EMP_NAME	VARCHAR(50)	Column denoting EMP_NAME representing name of employee
HIRE_DATE	DATE	Column denoting HIRE_DATE representing date on which employee is hired
EMP_SALARY	int(10)	Column denoting EMP_SALARY representing salary of the employee

Name	Type	Description
DEPT_ID	int(10)	Column denoting DEPT_ID representing id of the department where the employee works

Sample testcase 1
Input

EMPLOYEE

EMP_NO	EMP_NAME	HIRE_DATE	EMP_SALARY	DEPT_ID
101	Ram	2020-10-11	1000	34
102	John	2020-11-11	4500	15
103	Vipul	1990-10-11	90	34

Output

Ram	1000	34
John	4500	15
Vipul	90	34

```
-- Enter your query here
SELECT EMP_NAME,EMP_SALARY,DEPT_ID
FROM EMPLOYEE
```

```
WHERE EMP_SALARY>=(
    SELECT EMP_SALARY
    FROM EMPLOYEE
    WHERE EMP_NO=103
)
```

PROBLEM 5

You want to filter product details based on modified PROD_PRICE.

Write a MySQL query to display PRO_NO, PROD_NAME, MANU_DATE and modified PROD_PRICE of the products whose modified PROD_PRICE lies between 1000 and 7000.

Note: The modified PROD_PRICE is calculated as follows:

1. If manufacturing day is Wednesday, Modified PROD_PRICE is PROD_PRICE increased by 100.
2. If manufacturing day is Sunday, Modified PROD_PRICE is PROD_PRICE increased by 50.
3. If manufacturing day is Saturday, Modified PROD_PRICE is PROD_PRICE increased by 700.
4. Else Modified PROD_PRICE is PROD_PRICE increased by 500.

Schema
PRODUCT,

```
CREATE TEMPORARY TABLE ` PRODUCT ` (
  ` PRO_NO ` int NOT NULL,
  ` PROD_NAME ` varchar(50) DEFAULT NULL,
  ` MANU_DATE ` date DEFAULT NULL,
  ` PROD_PRICE ` int DEFAULT NULL,
  ` SALES_ID ` int DEFAULT NULL,
  PRIMARY KEY (` PRO_NO `)
) ENGINE = InnoDB DEFAULT CHARSET = latin1
```

Table structure
PRODUCT

Name	Type	Description
PRO_NO	int	Column denoting PRO_NO representing product number

Name	Type	Description
PROD_NAME	varchar(50)	Column denoting PROD_NAME representing product name
MANU_DATE	date	Column denoting MANU_DATE representing manufacturing date
PROD_PRICE	int	Column denoting PROD_PRICE representing price of product
SALES_ID	int	Column denoting SALES_ID representing is of salesman selling the product

Sample testcase 1
Input

PRODUCT

PRO_NO	PROD_NAME	MANU_DATE	PROD_PRICE	SALES_ID
101	Cake	2020-10-11	1000	34
102	Veg Lollipop	2020-11-11	4500	15
103	Biscuit	1990-10-11	90	34

Output

101	Cake	2020-10-11	1050
102	Veg Lollipop	2020-11-11	4600

```
SELECT
PRO_NO,PROD_NAME,MANU_DATE,
CASE
WHEN DAYOFWEEK(MANU_DATE)=4
THEN PROD_PRICE+100
WHEN DAYOFWEEK(MANU_DATE)=1
THEN PROD_PRICE+50
WHEN DAYOFWEEK(MANU_DATE)=7
THEN PROD_PRICE+700
ELSE PROD_PRICE+500
END AS MODIFIED_PROD_PRICE
FROM PRODUCT
HAVING MODIFIED_PROD_PRICE
BETWEEN 1000 AND 7000;
```
