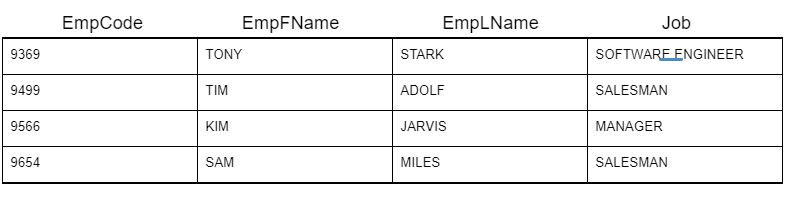
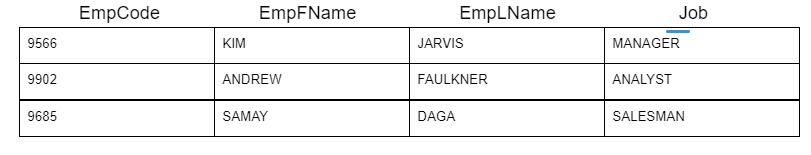
**UNION:**

**SQL Query - 1**

**Send Feedback**

**Problem Statement:**  
Using the tables given below, list out all the employees of the company.

**Information about the table:**  
Table **Empdept1:**:  
  
Table **Empdept2**:  


**Output Table Structure:**  


Note-1: The data should not contain duplicate rows of employees.

Note-2: Write keywords of syntax in uppercase alphabets.

SELECT EmpCode,EmpFName,EmpLName,Job

from Empdept1

UNION

SELECT EmpCode,EmpFName,EmpLName,Job

from Empdept2;

+---------+----------+----------+-------------------+

| EmpCode | EmpFName | EmpLName | Job |

+---------+----------+----------+-------------------+

| 9369 | TONY | STARK | SOFTWARE ENGINEER |

| 9499 | TIM | ADOLF | SALESMAN |

| 9566 | KIM | JARVIS | MANAGER |

| 9654 | SAM | MILES | SALESMAN |

| 9902 | ANDREW | FAULKNER | ANALYST |

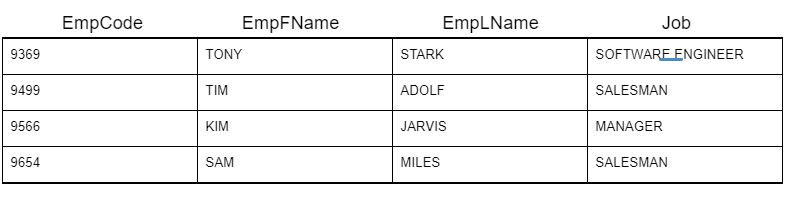
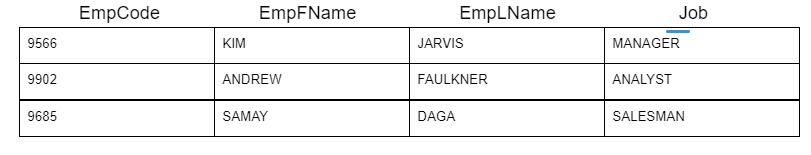
| 9685 | SAMAY | DAGA | SALESMAN |

+---------+----------+----------+-------------------+

**SQL Query - 2**

**Send Feedback**

**Problem Statement:**  
List down employees (all the details) from both the departments who work as Salesman.

**Information about the table:**  
Table **Empdept1:**:  
  
Table **Empdept2**:  


**Output Table Structure:**  


Note-1: The data should contain duplicate rows of employees.

Note-2: Write keywords of syntax in uppercase alphabets.

SELECT EmpCode,EmpFName,EmpLName,Job

from Empdept1

Where Job = 'SALESMAN'

UNION

SELECT EmpCode,EmpFName,EmpLName,Job

from Empdept2

Where Job = 'SALESMAN';

+---------+----------+----------+----------+

| EmpCode | EmpFName | EmpLName | Job |

+---------+----------+----------+----------+

| 9499 | TIM | ADOLF | SALESMAN |

| 9654 | SAM | MILES | SALESMAN |

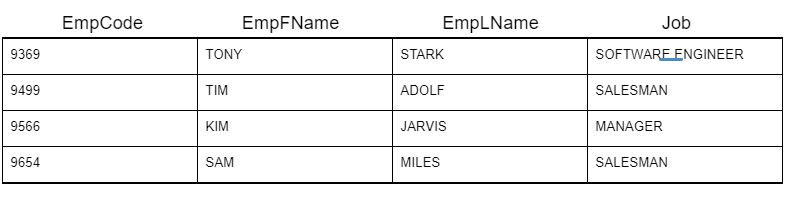
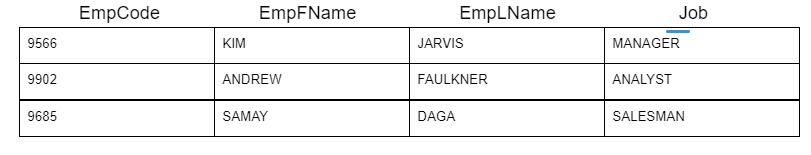
| 9685 | SAMAY | DAGA | SALESMAN |

+---------+----------+----------+----------+

**SQL Query - 3**

**Send Feedback**

**Problem Statement:**  
List out each employee name and employee code from both the departments and order them in ascending order by their code.

**Information about the table:**  
Table **Empdept1:**:  
  
Table **Empdept2**:  


**Output Table Structure:**  


Note-1: Duplicates are allowed.

Note-2: Write keywords of syntax in uppercase alphabets.

+----------+----------+---------+

| EmpFName | EmpLName | EmpCode |

+----------+----------+---------+

| TONY | STARK | 9369 |

| TIM | ADOLF | 9499 |

| KIM | JARVIS | 9566 |

| KIM | JARVIS | 9566 |

| SAM | MILES | 9654 |

| SAMAY | DAGA | 9685 |

| ANDREW | FAULKNER | 9902 |

+----------+----------+---------+

SELECT EmpFName,EmpLName,EmpCode

from

Empdept1

UNION ALL

SELECT EmpFName,EmpLName,EmpCode

from

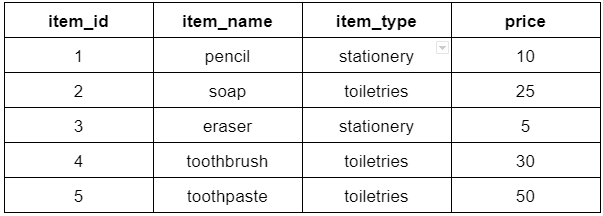
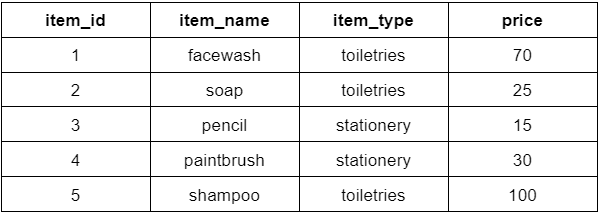
Empdept2

order by EmpCode;

**SQL query - 4**

**Send Feedback**

**Problem Statement:**  
Write a SQL query to print the item name, item type and price of all the items present in both the shops in descending order of their price.

**Information about the table:**  
Table **shop\_1**:  
  
Table **shop\_2**:  


**Output Table Structure:**  


Note: Write keywords of syntax in uppercase alphabets.

SELECT item\_name,item\_type,price

from

shop\_1

UNION ALL

SELECT item\_name,item\_type,price

from

shop\_2

order by price desc;

+------------+------------+-------+

| item\_name | item\_type | price |

+------------+------------+-------+

| shampoo | toiletries | 100 |

| facewash | toiletries | 70 |

| toothpaste | toiletries | 50 |

| toothbrush | toiletries | 30 |

| paintbrush | stationery | 30 |

| soap | toiletries | 25 |

| soap | toiletries | 25 |

| pencil | stationery | 15 |

| pencil | stationery | 10 |

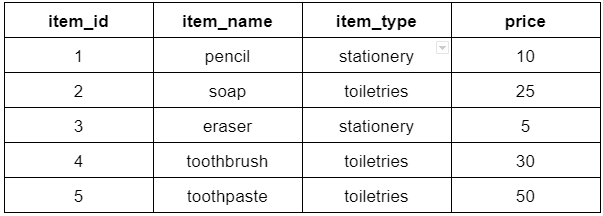
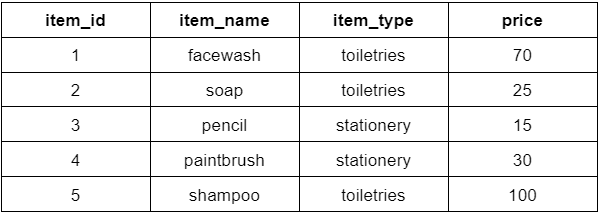
| eraser | stationery | 5 |

+------------+------------+-------+

**SQL query - 5**

**Send Feedback**

**Problem Statement:**  
Write a SQL query to get the item\_name, price of items in shop\_1 and shop\_2 where price is greater than 25.

**Information about the table:**  
Table **shop\_1**:  
  
Table **shop\_2**:  


**Output Table Structure:**  


Note: Write keywords of syntax in uppercase alphabets.

+------------+-------+

| item\_name | price |

+------------+-------+

| pencil | 10 |

| soap | 25 |

| eraser | 5 |

| toothbrush | 30 |

| toothpaste | 50 |

| facewash | 70 |

| paintbrush | 30 |

| shampoo | 100 |

+------------+-------+

SELECT item\_name,price

from

shop\_1

UNION ALL

SELECT item\_name,price

from

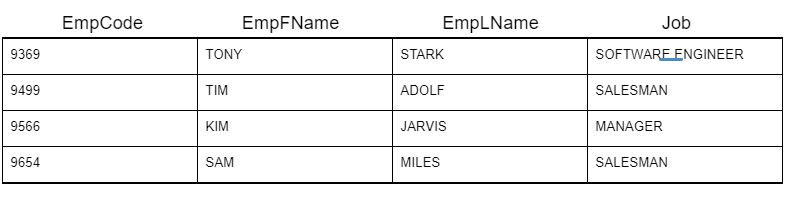
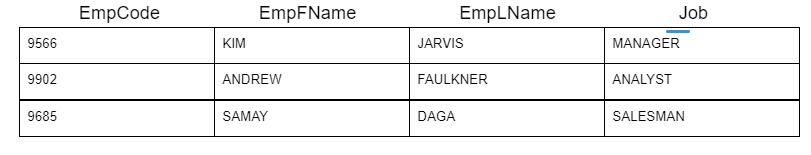
shop\_2

where price>25;

**SQL query - 6**

**Send Feedback**

**Problem Statement:**  
Find out all the details of employees that work for both the departments.

**Information about the table:**  
Table **Empdept1:**:  
  
Table **Empdept2**:  


**Output Table Structure:**  


Note-1: The data should not contain duplicate rows of employees.

Note-2: Write keywords of syntax in uppercase alphabets.

SELECT DISTINCT e1.EmpCode,e1.EmpFName,e1.EmpLName,e1.Job

from

Empdept1 e1

inner join

Empdept2 e2

on e1.EmpCode = e2.EmpCode;

+---------+----------+----------+---------+

| EmpCode | EmpFName | EmpLName | Job |

+---------+----------+----------+---------+

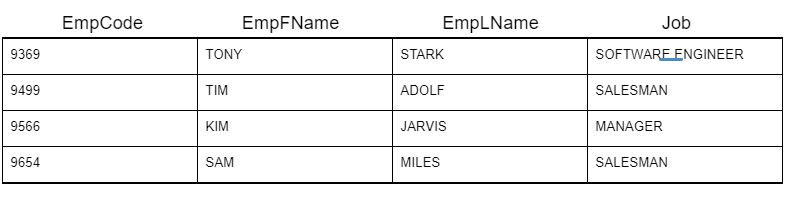
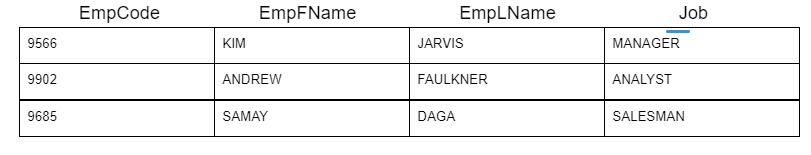
| 9566 | KIM | JARVIS | MANAGER |

+---------+----------+----------+---------+

**SQL query - 9**

**Send Feedback**

**Problem Statement:**  
List down all the details of employees working in dept1 but not in Dept2.

**Information about the table:**  
Table **Empdept1:**:  
  
Table **Empdept2**:  


**Output Table Structure:**  


Note-1: Write keywords of syntax in uppercase alphabets.

Note-2: Use employee code to link the two tables.

Select e1.EmpCode,e1.EmpFName,e1.EmpLName,e1.Job

from

Empdept1 e1

left join

Empdept2 e2

on e1.EmpCode = e2.EmpCode

where e2.EmpCode is null;

+---------+----------+----------+-------------------+

| EmpCode | EmpFName | EmpLName | Job |

+---------+----------+----------+-------------------+

| 9369 | TONY | STARK | SOFTWARE ENGINEER |

| 9499 | TIM | ADOLF | SALESMAN |

| 9566 | KIM | JARVIS | MANAGER |

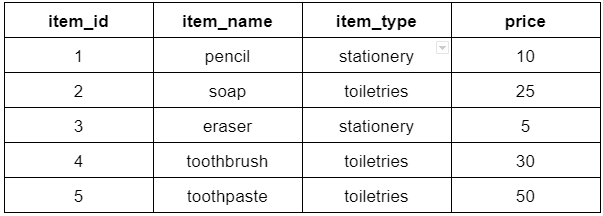
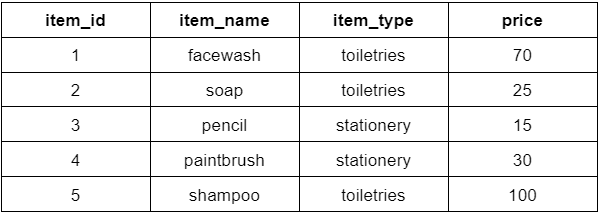
| 9654 | SAM | MILES | SALESMAN |

+---------+----------+----------+-------------------+

**SQL query - 7**

**Send Feedback**

**Problem Statement:**  
Write a SQL query to find the item name along with its type of stationery item which is available in both the shops.

**Information about the table:**  
Table **shop\_1**:  
  
Table **shop\_2**:  


**Output Table Structure:**  


Note: Write keywords of syntax in uppercase alphabets.

Select DISTINCT s1.item\_name,s2.item\_type

from

shop\_1 s1

inner join

shop\_2 s2

on s1.item\_name = s2.item\_name

where s1.item\_type = 'stationery';

+-----------+------------+

| item\_name | item\_type |

+-----------+------------+

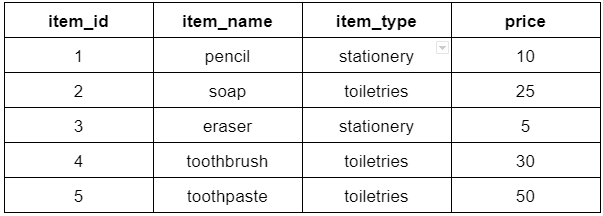
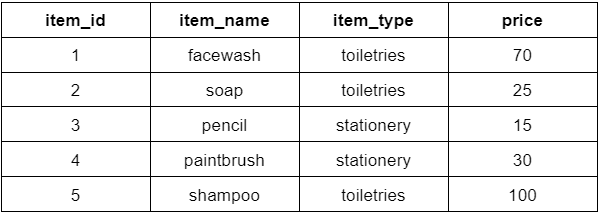
| pencil | stationery |

+-----------+------------+

**SQL query - 8**

**Send Feedback**

**Problem Statement:**  
Write a SQL query to find the name and price of items whose price is greater than 20 and available in both the shops.

**Information about the table:**  
Table **shop\_1**:  
  
Table **shop\_2**:  


**Output Table Structure:**  


Note: Write keywords of syntax in uppercase alphabets.

Select DISTINCT s1.item\_name,s1.price

from

shop\_1 s1

inner join

shop\_2 s2

on s1.item\_name = s2.item\_name

where s1.price > 20;

+-----------+-------+

| item\_name | price |

+-----------+-------+

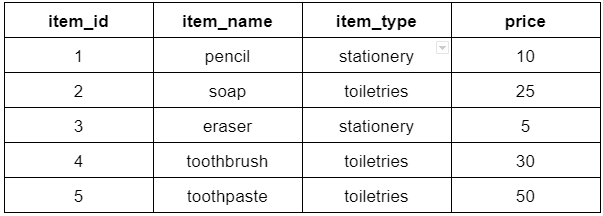
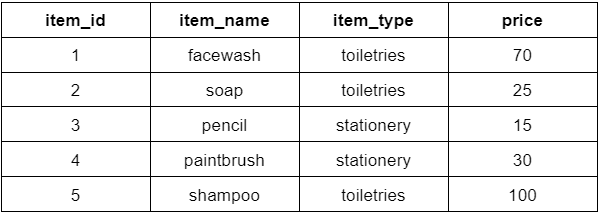
| soap | 25 |

+-----------+-------+

**SQL query - 10**

**Send Feedback**

**Problem Statement:**  
Write a SQL query to print the item name, item type of only the items which are available in shop 1 but not in shop 2 in the ascending order of item name.

**Information about the table:**  
Table **shop\_1**:  
  
Table **shop\_2**:  


**Output Table Structure:**  


Note: Write keywords of syntax in uppercase alphabets.

+------------+------------+

| item\_name | item\_type |

+------------+------------+

| eraser | stationery |

| toothbrush | toiletries |

| toothpaste | toiletries |

+------------+------------+

Select s1.item\_name,s1.item\_type

from

shop\_1 s1

left join

shop\_2 s2

on s1.item\_name = s2.item\_name

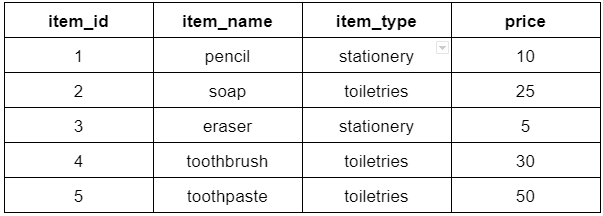
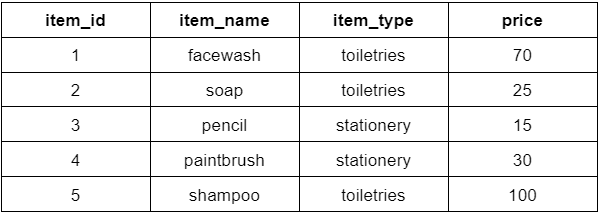
where s2.item\_name is null

order by s1.item\_name;

**SQL query - 11**

**Send Feedback**

**Problem Statement:**  
Write a SQL query to print the item name, price of only the items which are available in shop 2 but not in shop 1 whose price is greater than 50.

**Information about the table:**  
Table **shop\_1**:  
  
Table **shop\_2**:  


**Output Table Structure:**  


Note: Write keywords of syntax in uppercase alphabets.

Select s2.item\_name,s2.price

from

shop\_2 s2

left join

shop\_1 s1

on s1.item\_name = s2.item\_name

where s1.item\_name is null

and s2.price > 50;

+-----------+-------+

| item\_name | price |

+-----------+-------+

| facewash | 70 |

| shampoo | 100 |

+-----------+-------+