Need of Set Operations

1. No duplicates in the result set.
2. All distinct rows.

JOINS vs Set Operations

JOINS	Set Operations
No need of same set of attributes	Set of attributes must be same
Condition needs to met	No condition is required
Distinct columns are created	Distinct rows are created

The various Set Operations are:

1. UNION
2. INTERSECTION
3. MINUS

UNION

The UNION operator is used to combine the result-set of two or more SELECT statements.

1. Every SELECT statement within UNION must have the same number of columns
2. The columns must also have similar data types
3. The columns in every SELECT statement must also be in the same order

```
SELECT column_name(s) FROM table1
UNION
SELECT column_name(s) FROM table2;
```

UNION ALL

The UNION operator selects only distinct values by default. To allow duplicate values, use UNION ALL:

```
SELECT column_name(s) FROM table1
UNION ALL
SELECT column_name(s) FROM table2;
```

The column names in the result-set are usually equal to the column names in the first SELECT statement.

INTERSECT

The INTERSECT operator is used to get the intersection of the result-set of two or more SELECT statements. INTERSECT operation returns only common rows in both queries after sorting them and removing duplicates.

1. Every SELECT statement within INTERSECT must have the same number of columns
2. The columns must also have similar data types
3. The columns in every SELECT statement must also be in the same order

```
SELECT DISTINCT INNER JOIN;
```

MINUS
MINUS operation returns only the rows present in the first table that don’t appear in the second table after sorting them and removing duplicates.

TABLE1 – TABLE2

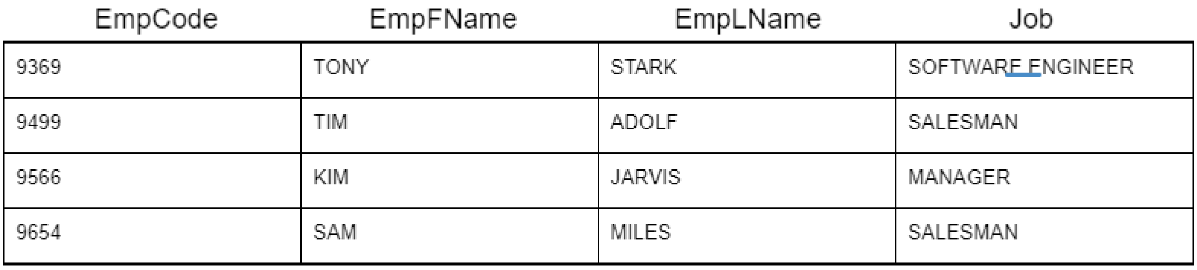
```
SELECT TABLE1.COL1, TABLE1.COL2, TABLE1.COL3 …………
FROM TABLE1
LEFT JOIN TABLE2
ON TABLE1.COL = TABLE2.COL
WHERE TABLE2.COL IS NULL;
```

TABLE2 – TABLE1

```
SELECT TABLE2.COL1, TABLE2.COL2, TABLE2.COL3 …………
FROM TABLE1
RIGHT JOIN TABLE2
ON TABLE1.COL = TABLE2.COL
WHERE TABLE1.COL IS NULL;
```

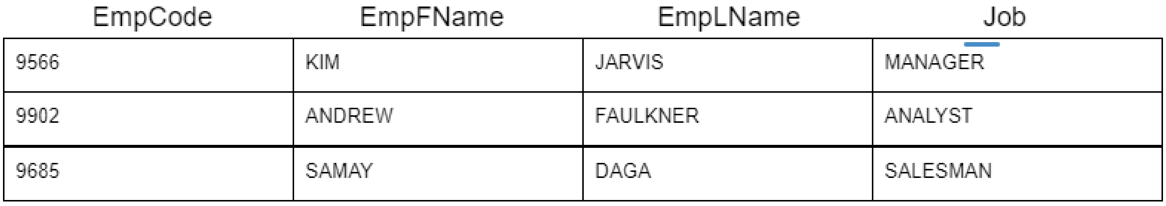
Q80: Using the tables given below, list out all the employees of the company. The data should not contain duplicate rows of employees.

EMPDEPT1



EmpCode	EmpFName	EmpLName	Job
9369	TONY	STARK	SOFTWARE ENGINEER
9499	TIM	ADOLF	SALESMAN
9566	KIM	JARVIS	MANAGER
9654	SAM	MILES	SALESMAN

EMPDEPT2



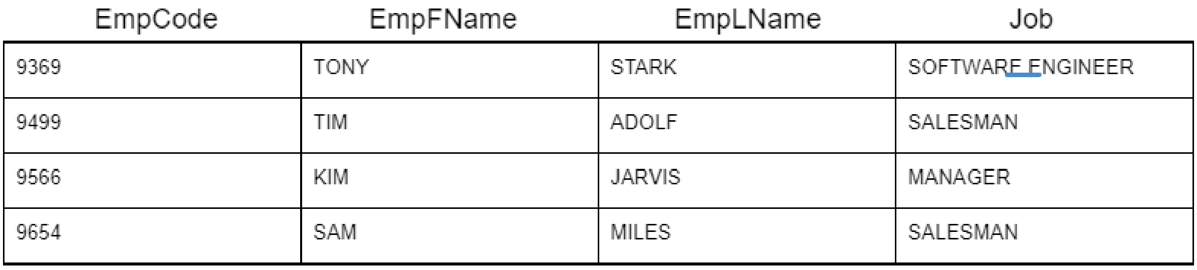
EmpCode	EmpFName	EmpLName	Job
9566	KIM	JARVIS	MANAGER
9902	ANDREW	FAULKNER	ANALYST
9685	SAMAY	DAGA	SALESMAN

Ans:

```
SELECT * FROM EMPDEPT1
UNION
SELECT * FROM EMPDEPT2;
```

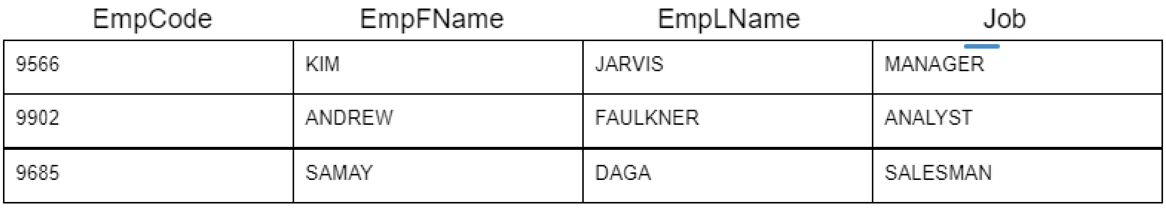
```
+---------+----------+----------+-------------------+
| EmpCode | EmpFName | EmpLName | Job               |
+---------+----------+----------+-------------------+
|    9369 | TONY     | STARK    | SOFTWARE ENGINEER |
|    9499 | TIM      | ADOLF    | SALESMAN          |
|    9566 | KIM      | JARVIS   | MANAGER           |
|    9654 | SAM      | MILES    | SALESMAN          |
|    9902 | ANDREW   | FAULKNER | ANALYST           |
|    9685 | SAMAY    | DAGA     | SALESMAN          |
+---------+----------+----------+-------------------+
Q81: List down employees (all the details) from both the departments who
work as Salesman. The data should contain duplicate rows of employees.
```

EMPDEPT1



EmpCode	EmpFName	EmpLName	Job
9369	TONY	STARK	SOFTWARE ENGINEER
9499	TIM	ADOLF	SALESMAN
9566	KIM	JARVIS	MANAGER
9654	SAM	MILES	SALESMAN

EMPDEPT2



EmpCode	EmpFName	EmpLName	Job
9566	KIM	JARVIS	MANAGER
9902	ANDREW	FAULKNER	ANALYST
9685	SAMAY	DAGA	SALESMAN

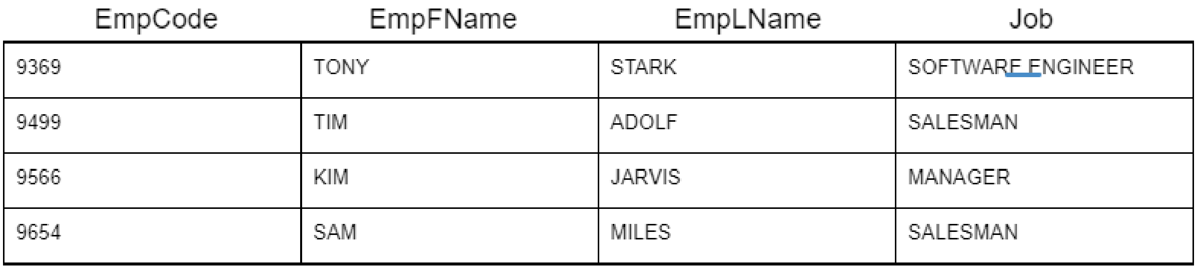
Ans:

```
SELECT * FROM EMPDEPT1 WHERE JOB = 'SALESMAN'
UNION ALL
SELECT * FROM EMPDEPT2 WHERE JOB = 'SALESMAN';
```

```
+---------+----------+----------+----------+
| EmpCode | EmpFName | EmpLName | Job      |
+---------+----------+----------+----------+
|    9499 | TIM      | ADOLF    | SALESMAN |
|    9654 | SAM      | MILES    | SALESMAN |
|    9685 | SAMAY    | DAGA     | SALESMAN |
+---------+----------+----------+----------+
```

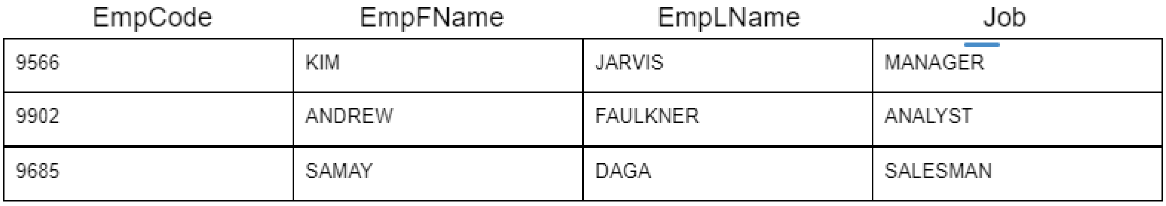
Q82: List out each employee name and employee code from both the departments and order them in ascending order by their code. Duplicates are allowed.

EMPDEPT1



EmpCode	EmpFName	EmpLName	Job
9369	TONY	STARK	SOFTWARE ENGINEER
9499	TIM	ADOLF	SALESMAN
9566	KIM	JARVIS	MANAGER
9654	SAM	MILES	SALESMAN

EMPDEPT2



EmpCode	EmpFName	EmpLName	Job
9566	KIM	JARVIS	MANAGER
9902	ANDREW	FAULKNER	ANALYST
9685	SAMAY	DAGA	SALESMAN

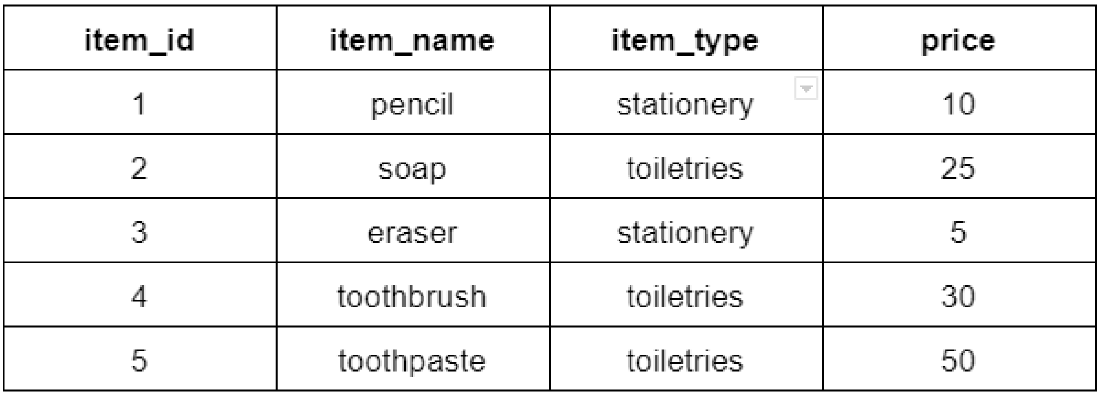
Ans:

```
SELECT EMPFNAME, EMPLNAME, EMPCODE FROM EMPDEPT1
UNION ALL
SELECT EMPFNAME, EMPLNAME, EMPCODE FROM EMPDEPT2
ORDER BY EMPCODE;
```

```
+----------+----------+---------+
| EMPFNAME | EMPLNAME | EMPCODE |
+----------+----------+---------+
| TONY     | STARK    |    9369 |
| TIM      | ADOLF    |    9499 |
| KIM      | JARVIS   |    9566 |
| KIM      | JARVIS   |    9566 |
| SAM      | MILES    |    9654 |
| SAMAY    | DAGA     |    9685 |
| ANDREW   | FAULKNER |    9902 |
+----------+----------+---------+
```

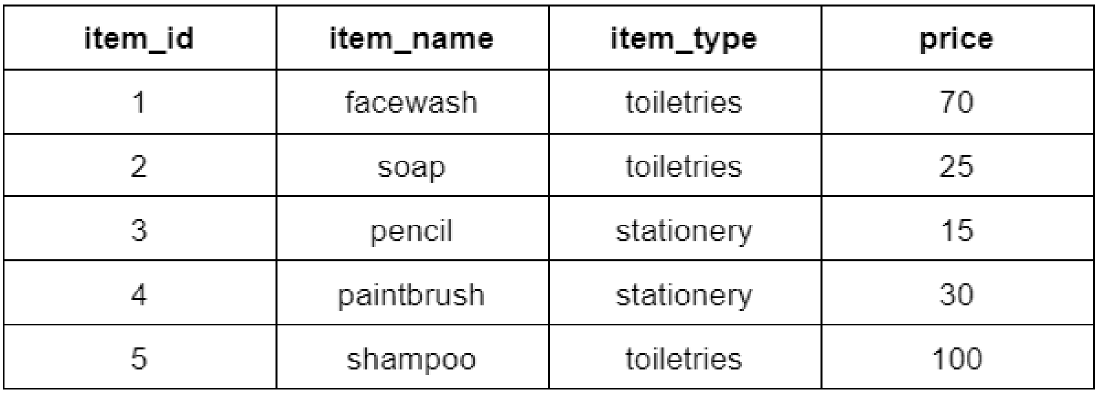
Q83: Write a SQL query to print the item name, item type and price of all the items present in both the shops in descending order of their price.

SHOP_1 TABLE



item_id	item_name	item_type	price
1	pencil	stationery	10
2	soap	toiletries	25
3	eraser	stationery	5
4	toothbrush	toiletries	30
5	toothpaste	toiletries	50

SHOP_2 TABLE



item_id	item_name	item_type	price
1	facewash	toiletries	70
2	soap	toiletries	25
3	pencil	stationery	15
4	paintbrush	stationery	30
5	shampoo	toiletries	100

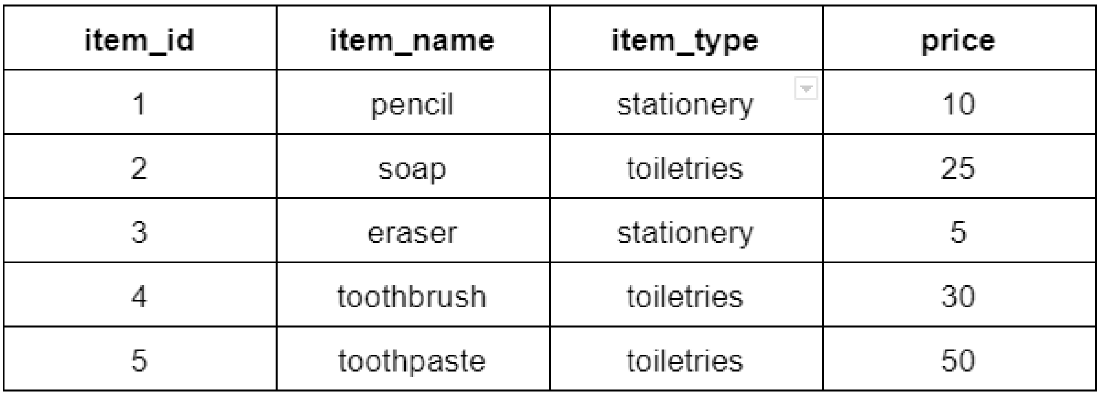
Ans:

```
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 |
+------------+-----------+-------+
```

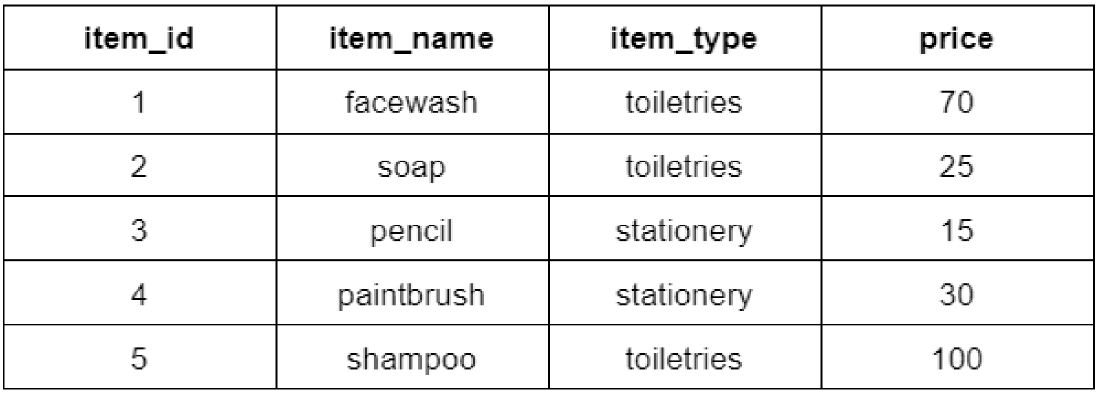
Q84: Write a SQL query to get the item_name, price of items in shop_1 and shop_2 where price is greater than 25.

SHOP_1 TABLE



item_id	item_name	item_type	price
1	pencil	stationery	10
2	soap	toiletries	25
3	eraser	stationery	5
4	toothbrush	toiletries	30
5	toothpaste	toiletries	50

SHOP_2 TABLE



item_id	item_name	item_type	price
1	facewash	toiletries	70
2	soap	toiletries	25
3	pencil	stationery	15
4	paintbrush	stationery	30
5	shampoo	toiletries	100

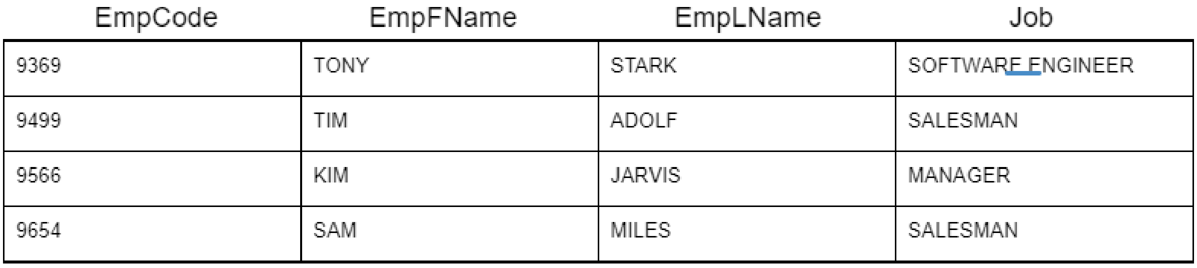
Ans:

```
SELECT ITEM_NAME, PRICE FROM SHOP_1 WHERE PRICE > 25
UNION ALL
SELECT ITEM_NAME, PRICE FROM SHOP_2 WHERE PRICE > 25;
```

```
+------------+------------+-------+
| 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 |
+------------+------------+-------+
```

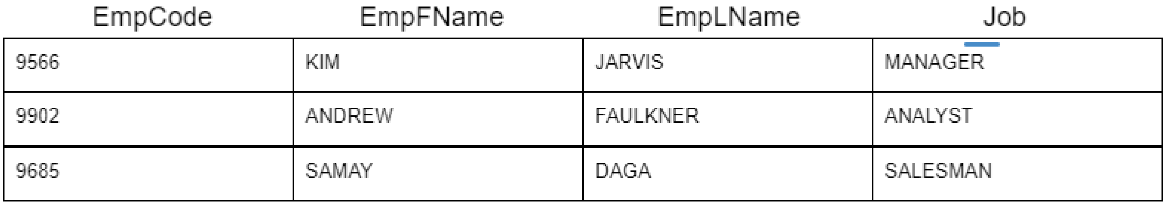
Q85: Find out all the details of employees that work for both the departments. The data should not contain duplicate rows of employees

EMPDEPT1



EmpCode	EmpFName	EmpLName	Job
9369	TONY	STARK	SOFTWARE ENGINEER
9499	TIM	ADOLF	SALESMAN
9566	KIM	JARVIS	MANAGER
9654	SAM	MILES	SALESMAN

EMPDEPT2



EmpCode	EmpFName	EmpLName	Job
9566	KIM	JARVIS	MANAGER
9902	ANDREW	FAULKNER	ANALYST
9685	SAMAY	DAGA	SALESMAN

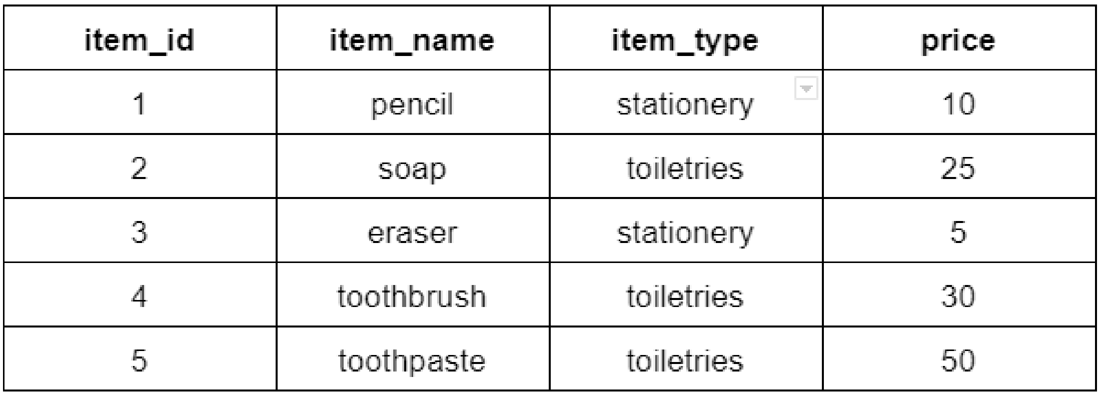
Ans:

```
SELECT DISTINCT EMPDEPT1.EMPCODE, EMPDEPT1.EMPFNAME, EMPDEPT1.EMPLNAME,
EMPDEPT1.JOB
FROM EMPDEPT1
INNER JOIN EMPDEPT2
ON EMPDEPT1.EMPCODE = EMPDEPT2.EMPCODE;
```

```
+---------+----------+----------+---------+
| EMPCODE | EMPFNAME | EMPLNAME | JOB     |
+---------+----------+----------+---------+
|    9566 | KIM      | JARVIS   | MANAGER |
+---------+----------+----------+---------+
```

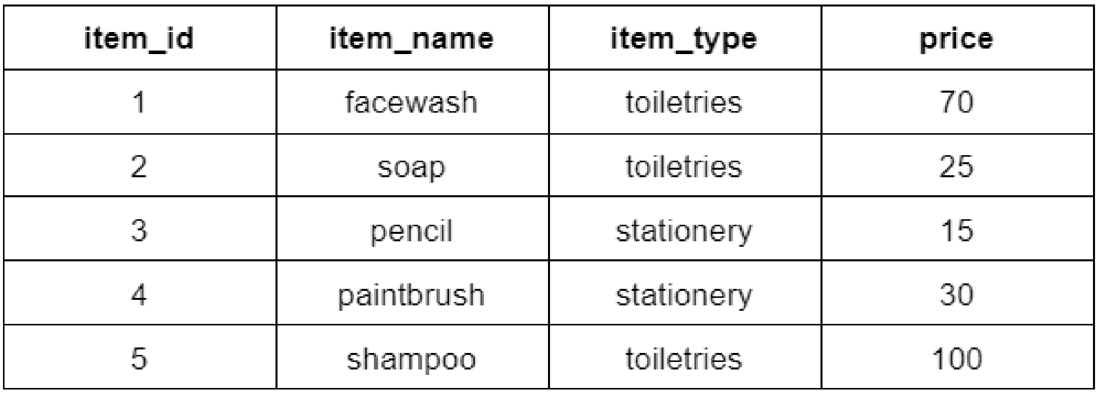
Q86: Write a SQL query to find the item name along with its type of stationery item which is available in both the shops.

SHOP_1 TABLE



item_id	item_name	item_type	price
1	pencil	stationery	10
2	soap	toiletries	25
3	eraser	stationery	5
4	toothbrush	toiletries	30
5	toothpaste	toiletries	50

SHOP_2 TABLE



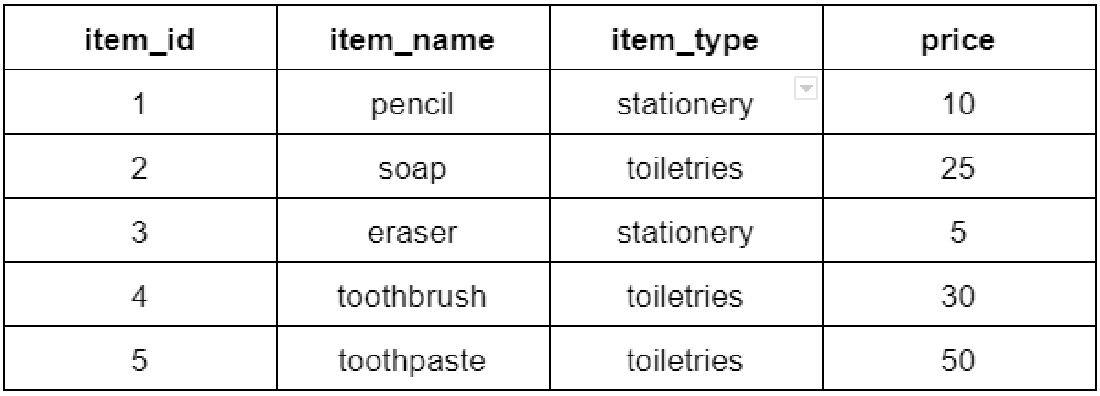
item_id	item_name	item_type	price
1	facewash	toiletries	70
2	soap	toiletries	25
3	pencil	stationery	15
4	paintbrush	stationery	30
5	shampoo	toiletries	100

Ans:

```
SELECT DISTINCT SHOP_1.ITEM_NAME, SHOP_1.ITEM_TYPE
FROM SHOP_1
INNER JOIN SHOP_2
ON SHOP_1.ITEM_NAME = SHOP_2.ITEM_NAME
WHERE SHOP_1.ITEM_TYPE = 'stationery';
```

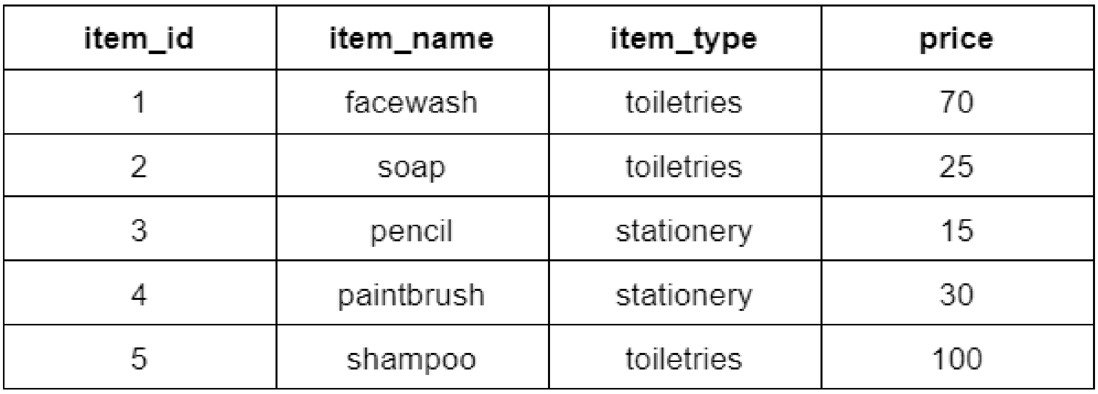
Q87: Write a SQL query to find the name and price of items whose price is greater than 20 and available in both the shops.

SHOP_1 TABLE



item_id	item_name	item_type	price
1	pencil	stationery	10
2	soap	toiletries	25
3	eraser	stationery	5
4	toothbrush	toiletries	30
5	toothpaste	toiletries	50

SHOP_2 TABLE



item_id	item_name	item_type	price
1	facewash	toiletries	70
2	soap	toiletries	25
3	pencil	stationery	15
4	paintbrush	stationery	30
5	shampoo	toiletries	100

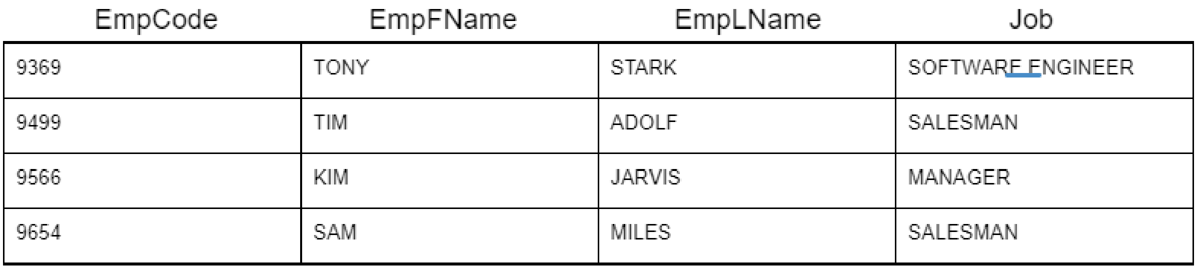
Ans:

```
SELECT DISTINCT SHOP_1.ITEM_NAME, SHOP_1.PRICE
FROM SHOP_1
INNER JOIN SHOP_2
ON SHOP_1.ITEM_NAME = SHOP_2.ITEM_NAME
WHERE SHOP_1.PRICE > 20;
```

```
+-----------+-------+
| ITEM_NAME | PRICE |
+-----------+-------+
| soap      |    25 |
+-----------+-------+
```

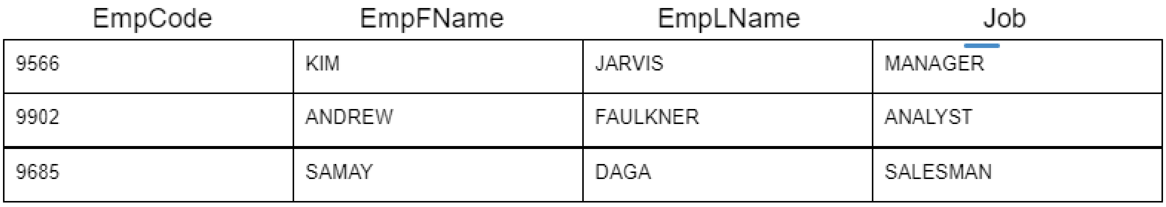
Q88: List down all the details of employees working in dept1 but not in Dept2.

EMPDEPT1



EmpCode	EmpFName	EmpLName	Job
9369	TONY	STARK	SOFTWARE ENGINEER
9499	TIM	ADOLF	SALESMAN
9566	KIM	JARVIS	MANAGER
9654	SAM	MILES	SALESMAN

EMPDEPT2



EmpCode	EmpFName	EmpLName	Job
9566	KIM	JARVIS	MANAGER
9902	ANDREW	FAULKNER	ANALYST
9685	SAMAY	DAGA	SALESMAN

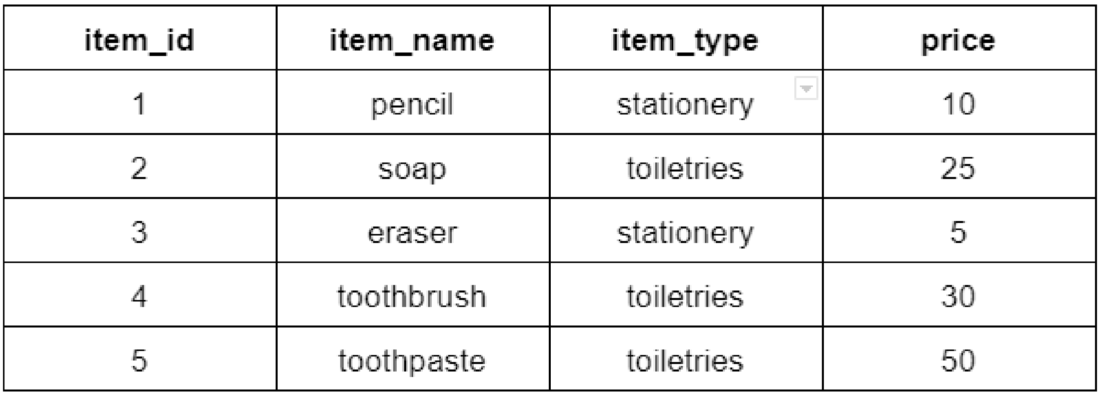
Ans:

```
SELECT EMPDEPT1.EMPCODE, EMPDEPT1.EMPFNAME, EMPDEPT1.EMPLNAME,
EMPDEPT1.JOB
FROM EMPDEPT1
LEFT JOIN EMPDEPT2
ON EMPDEPT1.EMPCODE = EMPDEPT2.EMPCODE
WHERE EMPDEPT2.EMPCODE IS NULL;
```

```
+---------+----------+----------+-------------------+
| EMPCODE | EMPFNAME | EMPLNAME | JOB               |
+---------+----------+----------+-------------------+
|    9369 | TONY     | STARK    | SOFTWARE ENGINEER |
|    9499 | TIM      | ADOLF    | SALESMAN          |
|    9654 | SAM      | MILES    | SALESMAN          |
+---------+----------+----------+-------------------+
```

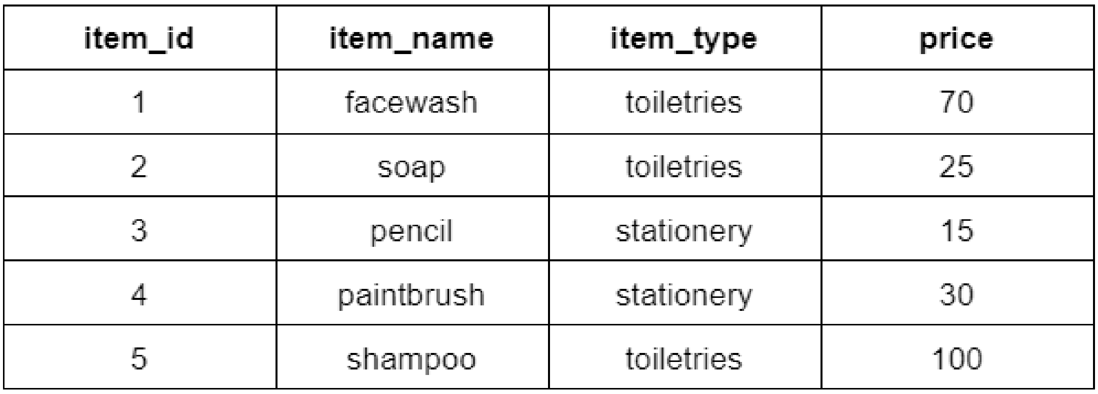
Q89: Write a SQL query to print the item name, item type of only the items which are available in shop 1 but not in shop 2 in the ascending order of item name.

SHOP_1 TABLE



item_id	item_name	item_type	price
1	pencil	stationery	10
2	soap	toiletries	25
3	eraser	stationery	5
4	toothbrush	toiletries	30
5	toothpaste	toiletries	50

SHOP_2 TABLE



item_id	item_name	item_type	price
1	facewash	toiletries	70
2	soap	toiletries	25
3	pencil	stationery	15
4	paintbrush	stationery	30
5	shampoo	toiletries	100

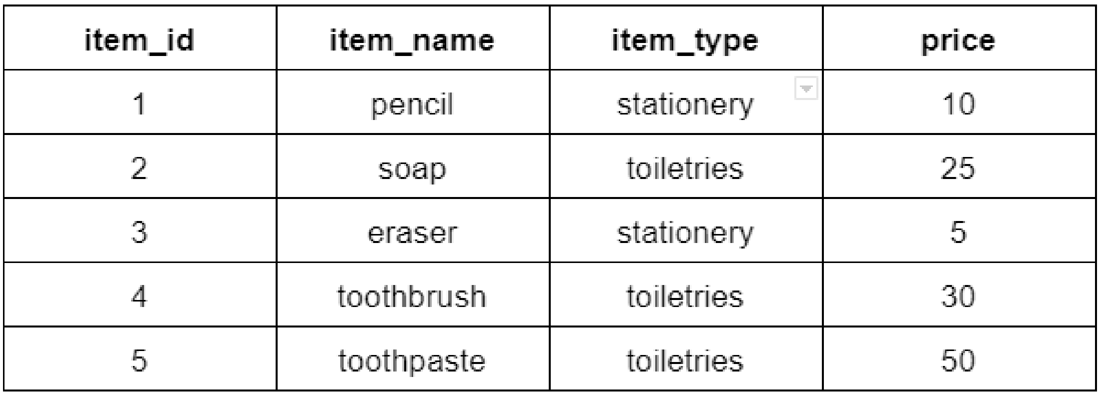
Ans:

```
SELECT SHOP_1.ITEM_NAME, SHOP_1.ITEM_TYPE
FROM SHOP_1
LEFT JOIN SHOP_2
ON SHOP_1.ITEM_NAME = SHOP_2.ITEM_NAME
WHERE SHOP_2.ITEM_NAME IS NULL
ORDER BY SHOP_1.ITEM_NAME;
```

```
+------------+------------+
| ITEM_NAME  | ITEM_TYPE  |
+------------+------------+
| eraser     | stationery |
| toothbrush | toiletries |
| toothpaste | toiletries |
+------------+------------+
```

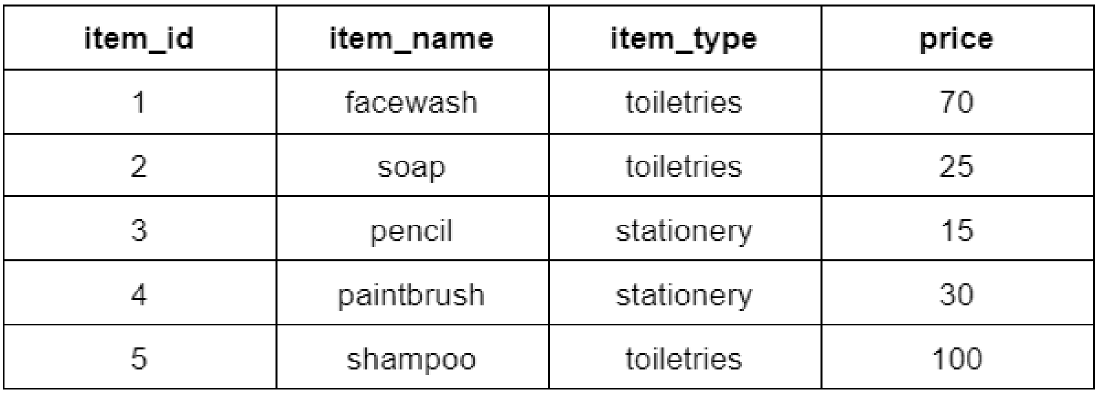
Q90: Write a SQL query to print the item name, price of only the items which are available in shop 2 but not in shop 1 whose price is greater than 50.

SHOP_1 TABLE



item_id	item_name	item_type	price
1	pencil	stationery	10
2	soap	toiletries	25
3	eraser	stationery	5
4	toothbrush	toiletries	30
5	toothpaste	toiletries	50

SHOP_2 TABLE



item_id	item_name	item_type	price
1	facewash	toiletries	70
2	soap	toiletries	25
3	pencil	stationery	15
4	paintbrush	stationery	30
5	shampoo	toiletries	100

Ans:

```
SELECT SHOP_2.ITEM_NAME, SHOP_2.PRICE
FROM SHOP_1
RIGHT JOIN SHOP_2
ON SHOP_1.ITEM_NAME = SHOP_2.ITEM_NAME
WHERE SHOP_1.ITEM_NAME IS NULL AND SHOP_2.PRICE > 50;
```

```
+-----------+-------+
| ITEM_NAME | PRICE |
+-----------+-------+
| facewash  |    70 |
| shampoo   |   100 |
+-----------+-------+
```