SQL - product, depot, stock

Write the following queries in SQL.

NOTE: I am using prod_id instead of #prods and depo_id instead of #deps, because naming a variable starting with # in mySQL would seemingly be ignored.

1. What are the #prods whose name begins with a 'p' and are less than $300.00?

```
SELECT Product. prod_id
FROM Product
Where Product.pname LIKE 'p%' AND Product.price<300;
```

2. Names of the products stocked in "d2".

(a) without in/not in

```
SELECT pname
FROM Product
INNER JOIN Stock ON product.prod_id=stock.prod_id
WHERE depo_id='d2';
```

(b) with in/not in

```
SELECT product.pname
From Product
WHERE product.prod_id IN
(SELECT Stock.prod_id
FROM Stock
WHERE depo_id='d2');
```

3. #prod and names of the products that are out of stock.

(a) without in/not in

```
SELECT product.prod_id, product.pname
FROM Product
INNER JOIN Stock ON product.prod_id=stock.prod_id
WHERE quantity<0;
```

(b) with in/not in

```
SELECT product.prod_id, product.pname
From Product
WHERE product.prod_id IN
(SELECT Stock.prod_id
FROM Stock
WHERE quantity<0);
```

4. Addresses of the depots where the product "p1" is stocked.

(a) without exists/not exists and without in/not in

```
SELECT Depot.addr
From Depot
INNER JOIN Stock ON Depot.depo_id=Stock.Depo_id
WHERE prod_id='p1';
```

(b) with in/not in

```
SELECT Depot.addr
From Depot
WHERE Depot.depo_id IN
(SELECT Stock.depo_id
FROM Stock
WHERE prod_id='p1');
```

(c) with exists/not exists

```
SELECT Depot.addr
From Depot
WHERE EXISTS
(SELECT Stock.depo_id
FROM Stock
WHERE prod_id='p1');
```

5. #prods whose price is between $250.00 and $400.00.

(a) using intersect.

```
SELECT pl250.prod_id
FROM
(SELECT prod_id FROM Product WHERE price>=250) pl250
INNER JOIN
(SELECT prod_id FROM Product WHERE price<=400) ps400
ON (pl250.prod_id= ps400.prod_id)
```

(b) without intersect.

```
SELECT Prod_id FROM Product WHERE price>=250 AND price<=400;
```

6. How many products are out of stock?

```
SELECT COUNT(*)
FROM Stock
Where quantity<0;
```

7. Average of the prices of the products stocked in the "d2" depot.

```
SELECT AVG(Product.price)
FROM Product
WHERE Product.prod_id IN
(SELECT Stock.prod_id
FROM Stock
WHERE depo_id='d2');
```

8. #deps of the depot(s) with the largest capacity (volume).

```
SELECT Depot.depo_id
FROM Depot
WHERE volume IN
(SELECT MAX(volume)
FROM Depot);
```

9. Sum of the stocked quantity of each product.

```
SELECT prod_id, SUM(quantity)
FROM stock
GROUP BY(prod_id);
```

10. Products names stocked in at least 3 depots.

(a) using count

```
SELECT Product.pname
From Product
Where Product.prod_id IN
(SELECT Stock.prod_id
FROM stock
GROUP BY prod_id
HAVING COUNT(depo_id) >=3);
```

(b) without using count

```
/* this query can work using division method because in the model there is exactly 3
Depot in this database*/
SELECT Product.pname
FROM Product
WHERE NOT EXISTS
(SELECT Depot.depo_id
FROM Depot
LEFT JOIN
```

```
        (SELECT Stock.depo_id
        FROM Stock
        WHERE Stock.prod_id=Product.prod_id) s
        ON(Depot.depo_id = s.depo_id)
        WHERE s.depo_id IS NULL);
```

11. #prod stocked in all depots.

(a) using count

```
SELECT Stock.prod_id
FROM stock
GROUP BY Stock.prod_id
HAVING COUNT(Stock.depo_id) = ( SELECT COUNT(*) FROM Depot);
```

(b) using exists/not exists

```
SELECT Product.prod_id
FROM Product
WHERE NOT EXISTS(
SELECT Depot.depo_id
FROM Depot
LEFT JOIN
(SELECT * FROM Stock WHERE Stock.prod_id= Product.prod_id) s
ON(Depot.depo_id=s.depo_id)
WHERE s.depo_id IS NULL);
```