Fragmentation with SQL server

Consider a hotel database. The Rooms relation is distributed over 2 sites and fragmented based on 2 scenario's:

Rooms (**Rid**, RoomNb, RoomType, Price, RoomFloor)

A. **Create an Horizontal Fragmentation of 2 fragments:**
   R1 room numbers >300, R2 room numbers <=300,

B. **Create Vertical Fragmentation 2 fragments :V1[Rid**, RoomNb, RoomType], V2[**Rid**, Price, RoomFloor]

Write the queries in each scenario A and B

QUERIES TO BE DONE SEPARATELY FOR A and B. YOU SHOULD HAVE A TOTAL OF 16 QUERIES.

1. Find the room numbers of rooms of type=" Suit".
2. Find the types of rooms with numbers greater than 700
3. Find the number of rooms with price > 300
4. Find the highest price
5. Find the total number of rooms
6. Find the Rid of the room having the highest Price.
7. Find the room Floors of rooms number less than 100
8. Find the Room Number and floor of rooms of type=" Double"

Rid	RoomNb	RoomType	Price	RoomFloor
1	101	Standard	100	1
2	102	Standard	100	2
3	201	Deluxe	150	2
4	202	Deluxe	150	2
5	50	Double	170	2
6	275	Double	450	3
7	280	Suit	200	3
8	105	Suit	215	3
9	401	Standard	500	4
10	402	Standard	100	4
11	501	Deluxe	320	5
12	502	Deluxe	150	7
13	701	Double	250	6
14	702	Double	250	7
15	801	Suit	315	5
16	80	Suit	300	8

A. Create a Horizontal Fragmentation of 2 fragments: R1 room numbers >300, R2 room numbers <=300

Table R1:

```
SELECT Rid, RoomNb, RoomType, Price, RoomFloor
FROM Rooms
WHERE RoomNb > 300;
```

Table R2:

```
SELECT Rid, RoomNb, RoomType, Price, RoomFloor
FROM Rooms
WHERE RoomNb <= 300;
```

1. Find the room numbers of rooms of type=" Suit".

```
SELECT RoomNb AS 'Room Nbr'
FROM R1
WHERE RoomType = 'Suit'
UNION
SELECT RoomNb AS 'Room Nbr'
FROM R2
WHERE RoomType = 'Suit';
```

Room Nbr
80
105
280
801

2. Find the types of rooms with numbers greater than 700

```
SELECT DISTINCT RoomType AS 'Room Type'
FROM R1
WHERE RoomNb > 700;
```

Room Type
Double
Suit

3. Find the number of rooms with price > 300

```
SELECT SUM(Nbr) AS 'Number Of Rooms'
FROM (
    SELECT COUNT(*) AS Nbr
    FROM R1
    WHERE Price > 300
    UNION ALL
    SELECT COUNT(*) AS Nbr
    FROM R2
    WHERE Price > 300
) AS RES;
```

Number Of Rooms
4

4. Find the highest price

```
SELECT MAX(HighestPrice) AS 'Highest Price'
FROM (
     SELECT MAX(Price) AS HighestPrice
     FROM R1
    UNION
    SELECT MAX(Price) AS HighestPrice
    FROM R2
) AS RES;
```

Highest Price
500

5. **Find the total number of rooms**

SOLUTION 1:

```
SELECT COUNT(Rid) AS 'Total Number Of Rooms'
FROM (
    SELECT Rid
    FROM R1
    UNION
    SELECT Rid
    FROM R2
) AS RES;
```

Total Number Of Rooms
16

SOLUTION 2:

(N.B: We should use UNION ALL to keep the duplication in case both table has the same count)

```
SELECT SUM(Nbr) AS 'Total Number Of Rooms'
FROM (
    SELECT COUNT(*) AS Nbr
    FROM R1
    UNION ALL
    SELECT COUNT(*) AS Nbr
    FROM R2
) AS RES;
```

Total Number Of Rooms
16

6. **Find the Rid of the room having the highest Price.**

SOLUTION 1: Using MAX

```
SELECT Rid
FROM (
    SELECT Rid, Price
    FROM R1
    UNION
    SELECT Rid, Price
    FROM R2
) AS Rooms
WHERE Price = (
    SELECT MAX(MaxPrice) AS 'Max Price'
    FROM (
        SELECT MAX(Price) AS MaxPrice
        FROM R1
        UNION
        SELECT MAX(Price) AS MaxPrice
        FROM R2
    ) AS RES
);
```

Rid
9

SOLUTION 2: TOP is use to get first row in SQL server

```
SELECT TOP 1 Rid
FROM (
    SELECT Rid, Price
    FROM R1
    UNION
    SELECT Rid, Price
    FROM R2
) AS RES
ORDER BY Price DESC;
```

Rid
9

7. **Find the room Floors of rooms number less than 100**

```
SELECT DISTINCT RoomFloor AS 'Room Floor'
FROM R2
WHERE RoomNb < 100;
```

Room Floor
2
8

8. **Find the Room Number and floor of rooms of type=" Double"**

```
SELECT RoomNb AS 'Room Number', RoomFloor AS 'Room Floor'
FROM R1
WHERE RoomType = 'Double'
UNION
SELECT RoomNb, RoomFloor
FROM R2
WHERE RoomType = 'Double';
```

Room Number	Room Floor
50	2
275	3
701	6
702	7

B. Create Vertical Fragmentation 2 fragments:

V1[<u>Rid</u>, RoomNb, RoomType], V2[**<u>Rid</u>**, Price, RoomFloor]

Table V1:

```
SELECT Rid, RoomNb, RoomType
FROM Rooms;
```

Table V2:

```
SELECT Rid, Price, RoomFloor
FROM Rooms;
```

1. **Find the room numbers of rooms of type=" Suit".**

```
SELECT RoomNb AS 'Room Nbr'
FROM V1
WHERE RoomType = 'Suit';
```

Room Nbr
80
105
280
801

2. **Find the types of rooms with numbers greater than 700**

```
SELECT DISTINCT RoomType AS 'Room Type'
FROM V1
WHERE RoomNb > 700;
```

Room Type
Double
Suit

3. **Find the number of rooms with price > 300**

```
SELECT COUNT(*) AS 'Number Of Rooms'
FROM V2
WHERE Price > 300;
```

Number Of Rooms
4

4. **Find the highest price**

```
SELECT MAX(Price) AS 'Highest Price'
FROM V2;
```

Highest Price
500

5. **Find the total number of rooms**

SOLUTION 1:

```
SELECT COUNT(*) AS 'Total Number Of Rooms'
FROM V1;
```

Total Number Of Rooms
16

SOLUTION 2:

```
SELECT COUNT(*) AS 'Total Number Of Rooms'
FROM V2;
```

6. **Find the Rid of the room having the highest Price.**

SOLUTION 1: Using MAX

```
SELECT Rid
FROM V2
WHERE Price = (
    SELECT MAX(Price) AS 'Max Price'
    FROM V2
);
```

Rid
9

SOLUTION 2: TOP is use to get first row in SQL server

```
SELECT TOP 1 Rid
FROM V2
ORDER BY Price DESC;
```

Rid
9

7. **Find the room Floors of rooms number less than 100**

```
SELECT DISTINCT V2.RoomFloor AS 'Room Floor'
FROM V2
JOIN V1 ON V1.Rid = V2.Rid
WHERE V1.RoomNb < 100;
```

Room Floor
2
8

8. **Find the Room Number and floor of rooms of type=" Double"**

```
SELECT V1.RoomNb AS 'Room Number', V2.RoomFloor AS 'Room Floor'
FROM V1
JOIN V2 ON V1.Rid = V2.Rid
WHERE V1.RoomType = 'Double';
```

Room Number	Room Floor
50	2
275	3
701	6
702	7