

Oracle SQL Practical question | SQL to convert comma separated string to rows and vice versa?

Write “SQL” to convert “row of data” into “comma separated string”?

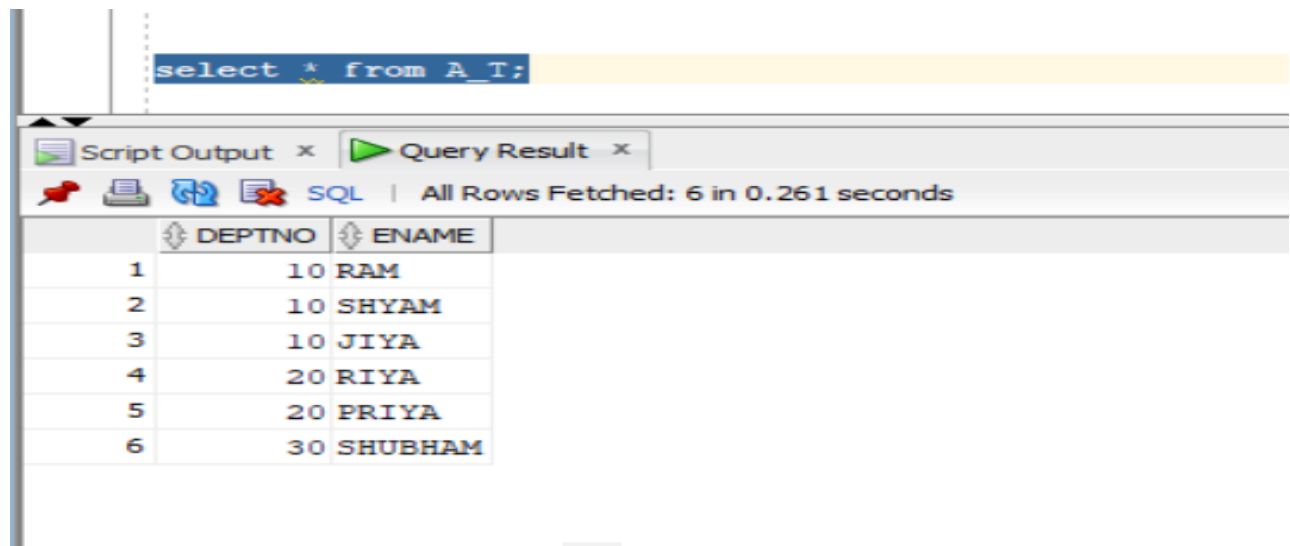
1: Selecting Data from Table A_T

Query:

```
SELECT * FROM A_T;
```

Explanation:

- This query retrieves all columns and rows from the table A_T.
- The result shows that the table A_T contains the following rows:
 - DEPTNO: 10, ENAME: RAM
 - DEPTNO: 10, ENAME: SHYAM
 - DEPTNO: 10, ENAME: JIYA
 - DEPTNO: 20, ENAME: RIYA
 - DEPTNO: 20, ENAME: PRIYA
 - DEPTNO: 30, ENAME: SHUBHAM



The screenshot shows the Oracle SQL Developer interface. At the top, a SQL script is entered: `select * from A_T;`. Below the script, the 'Query Result' tab is active, displaying the results of the query. The results are shown in a table with two columns: DEPTNO and ENAME. The table contains six rows of data, numbered 1 through 6 in the first column.

	DEPTNO	ENAME
1	10	RAM
2	10	SHYAM
3	10	JIYA
4	20	RIYA
5	20	PRIYA
6	30	SHUBHAM



Expected Output:

	DEPTNO	ENAME
1	10	JIYA, RAM, SHYAM
2	20	PRIYA, RIYA
3	30	SHUBHAM

2: Aggregating Names by Department Number

Query:

```
SELECT deptno, LISTAGG(ename, ',') WITHIN GROUP (ORDER BY ename) AS ENAME
FROM A_T
GROUP BY deptno;
```

Explanation:

- **LISTAGG Function:**
 - LISTAGG(ename, ',') WITHIN GROUP (ORDER BY ename) concatenates ename values within each group defined by deptno, separated by commas.
 - The ORDER BY ename clause ensures that the names are concatenated in alphabetical order.
- **GROUP BY Clause:**
 - GROUP BY deptno groups the rows by the deptno column.
- **Result:**
 - For DEPTNO: 10, the concatenated ENAME values are JIYA, RAM, SHYAM.
 - For DEPTNO: 20, the concatenated ENAME values are PRIYA, RIYA.
 - For DEPTNO: 30, the ENAME value is SHUBHAM



```
select deptno,listagg(ename,',') within group (order by ename) as ENAME
from A_T
group by deptno;
```

DEPTNO	ENAME
1	10 JIYA, RAM, SHYAM
2	20 PRIYA, RIYA
3	30 SHUBHAM

3: Final Query Result Explanation:

Query:

```
SELECT deptno, LISTAGG(ename, ',') WITHIN GROUP (ORDER BY ename) AS
ENAME FROM A_T GROUP BY deptno;
```

Explanation:

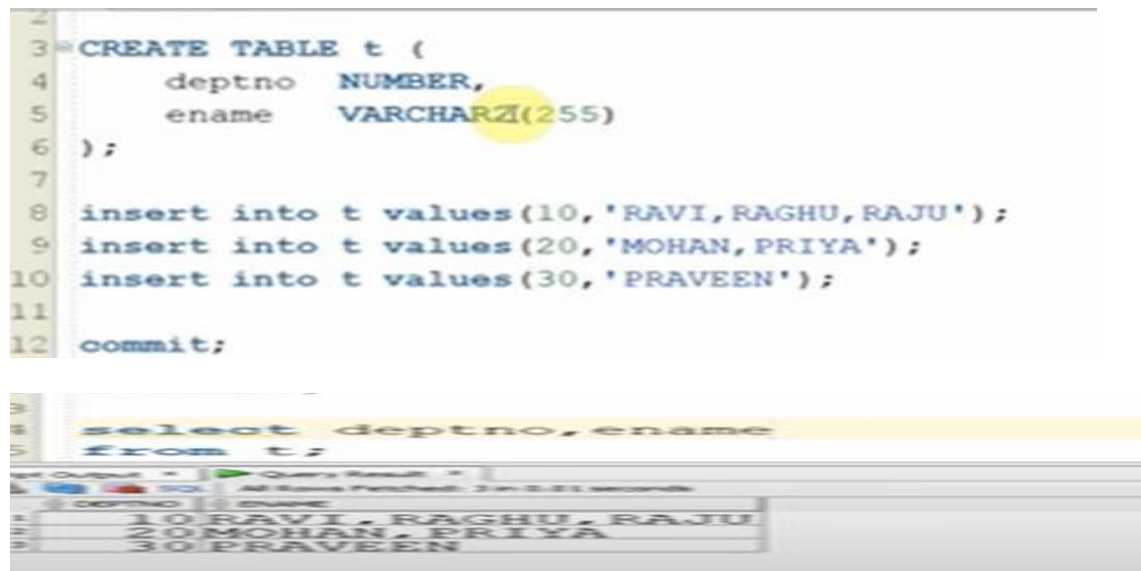
- The query in this screenshot is the same as the one in Screenshot 2.
- **Result:**
 - The query results in the following output:
 - DEPTNO: 10, ENAME: JIYA, RAM, SHYAM
 - DEPTNO: 20, ENAME: PRIYA, RIYA
 - DEPTNO: 30, ENAME: SHUBHAM

Summary

- The provided screenshots and queries demonstrate how to:
 1. Retrieve all rows from a table.
 2. Concatenate `ename` values within each `deptno` group, separated by commas, using the `LISTAGG` function in Oracle SQL.
 3. Ensure the concatenated names are ordered alphabetically within each group.

Question 2: Explain the process of converting a comma-separated string into individual rows and counting the elements in Oracle SQL and converting the strings in to rows?

Screenshot 1: Creating and Populating the Table



```
3 CREATE TABLE t (  
4     deptno    NUMBER,  
5     ename     VARCHAR2(255)  
6 );  
7  
8 insert into t values(10, 'RAVI, RAGHU, RAJU');  
9 insert into t values(20, 'MOHAN, PRIYA');  
10 insert into t values(30, 'PRAVEEN');  
11  
12 commit;
```



```
4 select deptno,ename  
5 from t;
```

deptno	ename
10	RAVI, RAGHU, RAJU
20	MOHAN, PRIYA
30	PRAVEEN

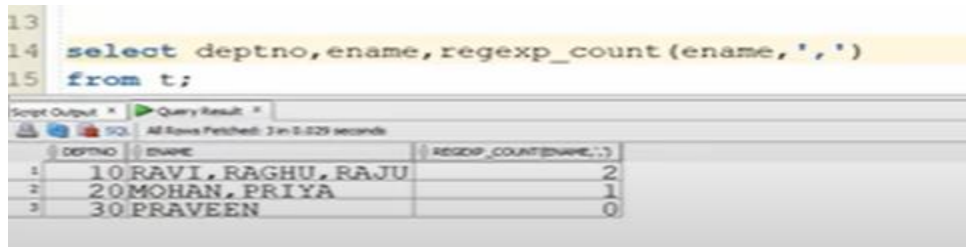
Explanation:

- This script creates a table `t` with columns `deptno` and `ename`.
- It then inserts three rows into the table:
 - `deptno = 10, ename = 'RAVI, RAGHU, RAJU'`
 - `deptno = 20, ename = 'MOHAN, PRIYA'`
 - `deptno = 30, ename = 'PRAVEEN'`

- The `COMMIT` statement saves these changes to the database.

Screenshot 2: Selecting Data from Table `t` and Counting Commas in `ename`

Query: `select deptno,ename ,regexp_count(ename,',') from t;`



```
13
14 select deptno,ename,regexp_count(ename,',')
15 from t;
```

DEPTNO	ENAME	REGEXP_COUNT(ENAME,',')
10	RAVI, RAGHU, RAJU	2
20	MOHAN, PRIYA	1
30	PRAVEEN	0

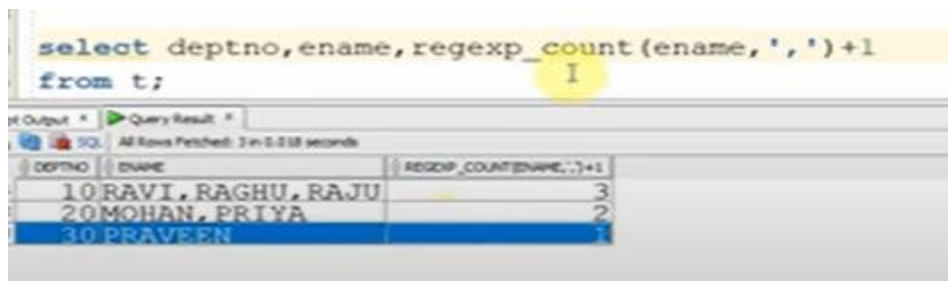
Explanation:

- `REGEXP_COUNT(ename, ',')` counts the number of commas in the `ename` column.
- The result shows the count of commas in each `ename`:
 - `deptno = 10` has 2 commas
 - `deptno = 20` has 1 comma
 - `deptno = 30` has 0 commas

Screenshot 3: Counting Names in `ename`

Query:

`select deptno,ename ,regexp_count(ename,',')+1 from t;`



```
select deptno,ename,regexp_count(ename,',')+1
from t;
```

DEPTNO	ENAME	REGEXP_COUNT(ENAME,',')+1
10	RAVI, RAGHU, RAJU	3
20	MOHAN, PRIYA	2
30	PRAVEEN	1

Explanation:

- `REGEXP_COUNT(ename, ',') + 1` calculates the number of names in the `ename` column by counting the commas and adding 1.
- The result shows the count of names in each `ename`:
 - `deptno = 10` has 3 names
 - `deptno = 20` has 2 names
 - `deptno = 30` has 1 name

Screenshot 4: Extracting First Word in *ename*

Query:

```
SELECT deptno, ename, REGEXP_COUNT(ename, ',') + 1, REGEXP_SUBSTR(ename, '\w+',
1, 1) FROM t, LATERAL (SELECT LEVEL AS lvl
FROM dual
CONNECT BY LEVEL <= REGEXP_COUNT(ename, ',') + 1
);
```

Explanation:

- REGEXP_SUBSTR(ename, '\w+', 1, 1) extracts the first word from the *ename* column using a regular expression.
- LATERAL subquery generates rows based on the count of names in the *ename* column.
- The result shows the first word extracted from each *ename*:
 - o deptno = 10, ename = RAVI, RAGHU, RAJU extracts RAVI

The screenshot shows a SQL query window with the following query:

```
select deptno,ename,regexp_count(ename,',')+1,
      regexp_substr(ename,'\w+',1,1)
from t,
      lateral(select level l
              from dual
```

The query result is displayed in a table with the following data:

DEPTNO	ENAME	REGEXP_COUNT(ENAME,',')+1
10	RAVI, RAGHU, RAJU	3
10	RAVI, RAGHU, RAJU	3
10	RAVI, RAGHU, RAJU	3
20	MOHAN, PRIYA	2
20	MOHAN, PRIYA	2
30	PRAVEEN	1

Screenshot 5: Extracting All Words in *ename*

The screenshot shows a SQL query window with the following query:

```
select deptno,ename,regexp_count(ename,',')+1,
      regexp_substr(ename,'\w+',1,1)
from t,
      lateral(select level l
              from dual
              connect by level <= regexp_count(ename,',')+1);
```

The query result is displayed in a table with the following data:

DEPTNO	ENAME	REGEXP_COUNT(ENAME,',')+1	REGEXP_SUBSTR(ENAME,'\w+',1,1)
1	10 RAVI, RAGHU, RAJU	3	RAVI
2	10 RAVI, RAGHU, RAJU	3	RAGHU
3	10 RAVI, RAGHU, RAJU	3	RAJU
4	20 MOHAN, PRIYA	2	MOHAN
5	20 MOHAN, PRIYA	2	PRIYA
6	30 PRAVEEN	1	PRAVEEN

Explanation of Queries:

```
SELECT deptno,
       --ename,regexp_count(ename,',')+1,
       regexp_substr(ename,'\w+',1,1)
FROM t,
LATERAL (SELECT LEVEL 1
         FROM dual
         CONNECT BY LEVEL <= regexp_count(ename,',')+1);
```

Explanation:

1. REGEXP_SUBSTR Function:

- `regexp_substr(ename,'\w+',1,1)` extracts the first word from the `ename` column. The `\w+` pattern matches one or more word characters.
- The parameters `1,1` specify the starting position and the occurrence to return, respectively.

2. LATERAL Clause:

- The `LATERAL` clause allows a subquery to reference columns from the preceding table `t` in the `FROM` clause.
- This subquery generates a sequence of numbers from 1 to the count of comma-separated values in the `ename` column plus one.

3. CONNECT BY LEVEL:

- `CONNECT BY LEVEL <= regexp_count(ename,',')+1` creates a hierarchical query that generates rows for each element in the comma-separated string.
- `LEVEL` is a pseudocolumn in Oracle used in hierarchical queries to denote the level number of a node in a tree structure. Here, it is used to generate rows.

Expected Output:

For an input where `ename` is 'RAVI,RAGHU,RAJU':

DEPTNO | REGEXP_SUBSTR(ENAME,'\W+',1,1)

```
-----
10      | RAVI
10      | RAGHU
10      | RAJU
20      | MOHAN

20      | PRIYA
30      | PRAVEEN
```

```
select deptno, --ename, regexp_count(ename, ',')+1,
        regexp_substr(ename, '\w+', 1, 1)
from t,
     lateral(select level 1
              from dual
              connect by level <= regexp_count(ename, ',')+1);
```

Query Result

All Rows Fetched: 6 in 0.014 seconds

DEPTNO	REGEXP_SUBSTR(ENAME, '\w+', 1, 1)
10	RAVI
10	RAGHU
10	RAJU
20	MOHAN
20	PRIYA
30	PRAVEEN

Final query:

```
SELECT
    deptno,
    regexp_substr(ename, '\w+', 1, 1) || ename
FROM
    t,
    LATERAL (
        SELECT
            ROWNUM 1
        FROM
            dual
        CONNECT BY
            level <= regexp_count(ename, ',') + 1
    );
```