Easy

Table Activities:

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| sell_date   | date    |
| product     | varchar |
+-------------+---------+
```

There is no primary key (column with unique values) for this table. It may contain duplicates.
Each row of this table contains the product name and the date it was sold in a market.

Write a solution to find for each date the number of different products sold and their names.

The sold products names for each date should be sorted lexicographically.

Return the result table ordered by sell_date.

The result format is in the following example.

Example 1:

Input:
Activities table:

```
+------------+------------+
| sell_date  | product    |
+------------+------------+
| 2020-05-30 | Headphone  |
| 2020-06-01 | Pencil     |
| 2020-06-02 | Mask       |
| 2020-05-30 | Basketball |
| 2020-06-01 | Bible      |
| 2020-06-02 | Mask       |
| 2020-05-30 | T-Shirt    |
+------------+------------+
```

Output:

```
+------------+----------+------------------------------+
| sell_date  | num_sold | products                     |
+------------+----------+------------------------------+
| 2020-05-30 | 3        | Basketball,Headphone,T-shirt |
| 2020-06-01 | 2        | Bible,Pencil                 |
| 2020-06-02 | 1        | Mask                         |
+------------+----------+------------------------------+
```

Explanation:
For 2020-05-30, Sold items were (Headphone, Basketball, T-shirt), we sort them lexicographically and separate them by a comma.
For 2020-06-01, Sold items were (Pencil, Bible), we sort them lexicographically and separate them by a comma.
For 2020-06-02, the Sold item is (Mask), we just return it.

```
# Write your MySQL query statement below
SELECT sell_date, COUNT(DISTINCT product) AS num_sold,
GROUP_CONCAT( DISTINCT product ORDER BY product) AS products
FROM Activities
GROUP BY sell_date
```

```
ORDER BY sell_date;
```