SQL SELECT statements: 8 typical cases of information needs

Case 1: Calculate maximum/minimum value

Typical examples:

Example 1: Which president(s) was (were) the youngest when he (they) married? List name, age at marriage, name of spouse and her age at marriage.

```
SELECT PRES_NAME, PRES_AGE, SPOUSE_NAME, SP_AGE
FROM PRES_MARRIAGE
WHERE PRES_AGE = (SELECT MIN (PRES_AGE)
                     FROM PRES_MARRIAGE)
```

Example 2: Which president(s) had most children and how many?

```
SELECT PRES_NAME, SUM (NR_CHILDREN)
FROM PRES_MARRIAGE
GROUP BY PRES_NAME
HAVING SUM (NR_CHILDREN) >= ALL
    (SELECT SUM (NR_CHILDREN)
        FROM PRES_MARRIAGE
        GROUP BY PRES_NAME)
```

If you need more than one record, calculate order by counting the records that have bigger/lower values using a self-join (see also case 2 en 6). Below Top 5 for Example 1:

```
SELECT PM1.PRES_NAME, PM1.PRES_AGE, PM1.SPOUSE_NAME, PM1.SP_AGE, COUNT(PM2.PRES_NAME)
FROM PRES_MARRIAGE PM1
       LEFT JOIN PRES_MARRIAGE PM2 ON PM1.PRES_NAME <> PM2.PRES_NAME
              AND PM1.SPOUSE_NAME <> PM2.SPOUSE_NAME AND PM1.PRES_AGE > PM2.PRES_AGE
GROUP BY PM1.PRES_NAME, PM1.PRES_AGE, PM1.SPOUSE_NAME, PM1.SP_AGE
HAVING COUNT(PM2.PRES_NAME) < 5
```

The above version is like the window function RANK() in later versions, this might (as in this example) result in ties and therefore more than 5 records. Using the window function ROW_NUMBER will eliminate the “wrong” number of records (loosing ties!).

```
SELECT PRES_NAME, PRES_AGE, SPOUSE_NAME, SP_AGE FROM (
       SELECT PRES_NAME, PRES_AGE, SPOUSE_NAME, SP_AGE,
              ROW_NUMBER() OVER (ORDER BY PRES_AGE) AS OrderNr
       FROM PRES_MARRIAGE
) tmp
WHERE OrderNr <= 5
```

In SQL Server using TOP (not ANSI) (Example 2):

```
SELECT TOP 1 PRES_NAME, SUM (NR_CHILDREN)
FROM PRES_MARRIAGE
GROUP BY PRES_NAME
ORDER BY 2 DESC
```

General approach: If the value for which you want to calculate the maximum/minimum is available on record level, then this maximum/minimum value is determined in a subquery as part of the WHERE-clause. In many cases you first have to group and for each item count the number of records or calculate a sum. Then you determine for which item you get the maximum/minimum value using a subquery in the HAVING-clause.

Case 2: Data from different rows of a table in the same row of the result set

Typical example: For each election year give the winner, the number of votes he got, together with the losers of the that election and their number of votes.

```
SELECT W.ELECTION_YEAR, W.CANDIDATE, W. VOTES, L.CANDIDATE, L. VOTES
FROM ELECTION AS W
    INNER JOIN ELECTION AS L ON W.ELECTION_YEAR = L.ELECTION_YEAR
WHERE W.WINNER_LOSER_INDIC = 'W'
    AND L.WINNER_LOSER_INDIC = 'L'
ORDER BY W.ELECTION_YEAR ASC, L.VOTES
```

How to recognize: If data from different rows of the same table has to be displayed in the same row of the result table, then a self join can be (or sometimes: has to be) used. A self join can be typically used in tables with a self reference.

Case 3: Don't forget to count missing items

Typical example:
Give for *each* president the number of hobbies.
Remark: If there are no hobbies registered, then the president has to be displayed with the value 0.

Solution 1 (using an outer join):

```
SELECT P.PRES_NAME, COUNT (HOBBY)
FROM PRESIDENT AS P
    LEFT OUTER JOIN PRES_HOBBY AS PH ON P.PRES_NAME = PH.PRES_NAME
GROUP BY P.PRES_NAME
```

Solution 2 (using a subquery in the SELECT-clause):

```
SELECT PRES_NAME, (SELECT COUNT (HOBBY)
                   FROM PRES_HOBBY AS PH
                   WHERE   PRES_NAME = P.PRES_NAME)
FROM PRESIDENT AS P
```

How to recognize: If in case of a join all rows from one table have to be displayed, also if there is no corresponding row in the other table.

Case 4: Calculate min/max value for each item

Typical example:
For each president give his name, first year of inauguration and his 'age' at that moment, in ascending order of inauguration year.

Solution 1:

```
SELECT A1.PRES_NAME, YEAR_INAUGURATED, YEAR_INAUGURATED - BIRTH_YR AS AGE
FROM ADMINISTRATION A
    INNER JOIN PRESIDENT P ON A.PRES_NAME = P.PRES_NAME
WHERE YEAR_INAUGURATED = (SELECT MIN(YEAR_INAUGURATED)
                            FROM ADMINISTRATION
                            WHERE PRES_NAME = A.PRES_NAME)
```

Solution 2 (in fact the same as solution 1, but now using a view):

```
CREATE VIEW PRES_AGE (PRES_NAME,YEAR_INAUGURATED,AGE) AS
SELECT A.PRES_NAME, YEAR_INAUGURATED, YEAR_INAUGURATED - BIRTH_YR
FROM ADMINISTRATION A
    INNER JOIN PRESIDENT P ON A.PRES_NAME = P.PRES_NAME

SELECT PRES_NAME, YEAR_INAUGURATED, AGE
FROM PRES_AGE PA
WHERE AGE = (SELECT MIN(AGE)
                FROM PRES_AGE
                WHERE PRES_NAME = PA.PRES_NAME)
```

Solution 3 (Using the view and a subquery in the FROM-clause):

```
SELECT P1.PRES_NAME, YEAR_INAUGURATED, AGE
FROM PRES_AGE P1
    INNER JOIN (SELECT PRES_NAME, MIN(AGE) AS MIN_AGE
                FROM PRES_AGE
                GROUP BY PRES_NAME) P2
    ON P1.PRES_NAME = P2.PRES_NAME AND AGE = MIN_AGE
```

General approach:
Question: List for each item the itemid, the value and the maximum number for this item.

Table ItemValue

itemid	value	number
i1	v1	n1
i1	v2	n2
i2	v3	n3
i2	v1	n4
i2	v4	n5

Solution 1:

```
SELECT itemid, value, number
FROM ItemValue IV
WHERE number = (SELECT MAX(number)
                    FROM ItemValue
                    WHERE itemid = IV.itemid)
```

Solution 2:

```
SELECT itemid, value, number
FROM ItemValue IV1
    INNER JOIN (SELECT itemid, MAX(number) AS maxnumber
                FROM ItemValue
                GROUP BY itemid) IV2
    ON IV1.itemid = IV2.itemid AND number = maxnumber
```

Case 5: Which … has all … (DIVIDE BY)

Typical example: Which presidents practiced all registered hobbies?

A nested NOT EXISTS query:

```
SELECT PRES_NAME
FROM PRESIDENT P
WHERE NOT EXISTS (SELECT *
                    FROM PRES_HOBBY PH
                    WHERE NOT EXISTS (SELECT *
                                        FROM PRES_HOBBY
                                        WHERE   PRES_NAME = P.PRES_NAME
                                        AND HOBBY = PH.HOBBY))
```

Different solution (by counting hobbies):

```
SELECT PRES_NAME
FROM PRES_HOBBY
GROUP BY PRES_NAME
HAVING COUNT (*) = (SELECT COUNT (DISTINCT HOBBY)
                    FROM PRES_HOBBY)
```

Another solution (using EXCEPT):

```
SELECT PRES_NAME
FROM PRESIDENT P
WHERE NOT EXISTS (
    SELECT HOBBY
    FROM PRES_HOBBY
    EXCEPT
    SELECT HOBBY
    FROM PRES_HOBBY
    WHERE PRES_NAME = P.PRES_NAME)
```

General approach:
You can choose between three types of solutions:

1. Rephrase the question to “For which … there exists no … that … did not …”. The resulting query has a correlated subquery with a nested correlated subquery and two times NOT EXISTS. *For example: For which presidents there exists no hobby that this president did not practice?*
2. Count the number of ‘things’ for each item and compare it with the total number of ‘things’. *For example: Count number of hobbies for this president and compare it with total number of registered hobbies.*
3. Alternative for type 1 using an EXCEPT instead of the deepest nested subquery).

Case 6: Determine for each item the highest/lowest n

Typical example: List for each candidate the first two times he joined the elections.

Solution 1: query with correlated subquery

```
SELECT CANDIDATE, ELECTION_YEAR
FROM ELECTION E
WHERE (SELECT COUNT(*)
        FROM ELECTION
        WHERE CANDIDATE = E.CANDIDATE
            AND ELECTION_YEAR < E.ELECTION_YEAR) < 2
```

Solution 2: using self join

```
SELECT E1.CANDIDATE, E1.ELECTION_YEAR
FROM ELECTION E1 LEFT
    JOIN ELECTION E2 ON E1.CANDIDATE = E2.CANDIDATE AND E2.ELECTION_YEAR < E1.ELECTION_YEAR
GROUP BY E1.CANDIDATE, E1.ELECTION_YEAR
HAVING COUNT(E2.ELECTION_YEAR) < 2
```

Solution 3: a relatively simple query using a CTE and Window functions

```
WITH CANDIDATE_PARTICIPATION (CANDIDATE, ELECTION_YEAR, SEQNO) AS
(
    SELECT CANDIDATE, ELECTION_YEAR,
    RANK() OVER (PARTITION BY CANDIDATE ORDER BY ELECTION_YEAR)
    FROM ELECTION
)
SELECT *
FROM CANDIDATE_PARTICIPATION
WHERE SEQNO <= 2
```

These solutions are easily extendable to the first three (and also four, five, …) times he joined the elections.
For example:

```
SELECT CANDIDATE, ELECTION_YEAR
FROM ELECTION E
WHERE (SELECT COUNT(*)
        FROM ELECTION
        WHERE CANDIDATE = E.CANDIDATE
            AND ELECTION_YEAR < E.ELECTION_YEAR) < 3 -etc..
```

Case 7: Find items with exactly the same set of rows

Typical example: Which presidents have exactly the same hobbies?

```
SELECT PH1.PRES_NAME, PH2.PRES_NAME
FROM PRES_HOBBY PH1
    INNER JOIN PRES_HOBBY PH2
        ON PH1.PRES_NAME < PH2.PRES_NAME AND PH1.HOBBY = PH2.HOBBY
GROUP BY PH1.PRES_NAME, PH2.PRES_NAME
HAVING COUNT(*) = -- number of hobbies of 1st president
    (SELECT COUNT(*)
    FROM PRES_HOBBY
    WHERE PRES_NAME = PH1.PRES_NAME)
 AND COUNT(*) = -- number of hobbies of 2nd president
    (SELECT COUNT(*)
    FROM PRES_HOBBY
    WHERE PRES_NAME = PH2.PRES_NAME)
```

Other solution:

```
CREATE VIEW PRESIDENT_WITH_HOBBY AS
SELECT * FROM PRESIDENT
    WHERE PRES_NAME IN (SELECT PRES_NAME FROM PRES_HOBBY)

SELECT P1.PRES_NAME, P2.PRES_NAME
FROM PRESIDENT_WITH_HOBBY P1, PRESIDENT_WITH_HOBBY P2
WHERE P1.PRES_NAME < P2.PRES_NAME
    AND NOT EXISTS -- a hobby of 1st president which is not a hobby of 2nd president
    (SELECT HOBBY
        FROM PRES_HOBBY
        WHERE PRES_NAME = P1.PRES_NAME
    EXCEPT
        SELECT HOBBY
        FROM PRES_HOBBY
        WHERE PRES_NAME = P2.PRES_NAME)
    AND NOT EXISTS -- a hobby of 2nd president which is not a hobby of 1st president
    (SELECT HOBBY
        FROM PRES_HOBBY
        WHERE PRES_NAME = P2.PRES_NAME
    EXCEPT
        SELECT HOBBY
        FROM PRES_HOBBY
        WHERE PRES_NAME = P1.PRES_NAME)
```

Yet another solution:

```
SELECT PH1.PRES_NAME, PH2.PRES_NAME
FROM PRESIDENT_WITH_HOBBY PH1, PRESIDENT_WITH_HOBBY PH2
WHERE PH1.PRES_NAME < PH2.PRES_NAME
    AND NOT EXISTS -- a hobby of 1st president which is not a hobby of 2nd president
    (SELECT * FROM PRES_HOBBY PH3
        WHERE PRES_NAME = PH2.PRES_NAME
        AND NOT EXISTS (SELECT *
                        FROM PRES_HOBBY
                        WHERE PRES_NAME = PH1.PRES_NAME AND HOBBY = PH3.HOBBY))
    AND NOT EXISTS -- a hobby of 2nd president which is not a hobby of 1st president
    (SELECT * FROM PRES_HOBBY PH4
        WHERE PRES_NAME = PH1.PRES_NAME
        AND NOT EXISTS (SELECT *
                        FROM PRES_HOBBY
                        WHERE PRES_NAME = PH2.PRES_NAME AND HOBBY = PH4.HOBBY))
```

General approach: Suppose you deal with the following table.

Table ItemValue

Itemid	Value	Number
i1	v1	n1
i1	v2	n2
i2	v1	n1
i2	v2	n3
I3	v1	n1
I3	v2	n2
I3	v3	n3
I4	v1	n1
I4	v2	n2

Question: Show all items with exactly the same combinations of value and number.

```sql
SELECT IV1.itemid, IV2.itemid
FROM ItemValue IV1
    INNER JOIN ItemValue IV2 IV1.itemid < IV2.itemid
        AND IV1.value = IV2.value AND IV1.number = IV2.number
GROUP BY IV1.itemid, IV2.itemid
HAVING COUNT(*) = (SELECT COUNT(*)
                    FROM ItemValue
                    WHERE itemid = IV1.itemid)
AND    COUNT(*) = (SELECT COUNT(*)
                    FROM ItemValue
                    WHERE itemid = IV2.itemid)
```

Case 8: Generate sequence numbers

Typical example:
For each state give state name and a rank number for the year that the state joined the the Union. The state that joined first gets rank number 1, etc. States that joined the Union in the same year get same rank numbers, whilst next rank number will not be given in that case.

Solution 1:

```
SELECT S1.STATE_NAME, COUNT (S2.STATE_NAME) + 1 AS SEQNO
FROM STATE AS S1
    LEFT OUTER JOIN STATE AS S2 ON S1.YEAR_ENTERED > S2.YEAR_ENTERED
GROUP BY S1.STATE_NAME
ORDER BY SEQNO
```

Solution 2:

```
SELECT STATE_NAME, (SELECT COUNT (*) + 1
                    FROM STATE
                    WHERE YEAR_ENTERED < S.YEAR_ENTERED) AS SEQNO
FROM STATE S
ORDER BY SEQNO
```

Solution 3:
In SQL Server there is a function RANK() which makes it easier:

```
SELECT STATE_NAME, RANK() OVER (ORDER BY YEAR_ENTERED)
FROM STATE
```

Another typical example:
For each candidate in an election give the election year, the name of the candidate, the number of votes obtained, the rank number of this candidate in order of the number of votes, and the total number of votes of the election.

Solution 1:

```
SELECT ELECTION_YEAR, CANDIDATE, VOTES,
    (SELECT COUNT (*) + 1
        FROM ELECTION
        WHERE ELECTION_YEAR = E.ELECTION_YEAR AND VOTES > E.VOTES) AS SEQNO,
    (SELECT SUM(VOTES)
        FROM ELECTION
        WHERE ELECTION_YEAR = E.ELECTION_YEAR) AS SUMVOTES
FROM ELECTION E
ORDER BY ELECTION_YEAR, SEQNO
```

Solution 2, using Window functions:

```
SELECT ELECTION_YEAR, CANDIDATE, VOTES,
    RANK() OVER (PARTITION BY ELECTION_YEAR ORDER BY VOTES DESC) AS SEQNO,
    SUM(VOTES) OVER (PARTITION BY ELECTION_YEAR ORDER BY ELECTION_YEAR) AS SUMVOTES
FROM ELECTION
ORDER BY ELECTION_YEAR, SEQNO
```