DATABASE

Öğr. Gör. Evgin GÖÇERİ

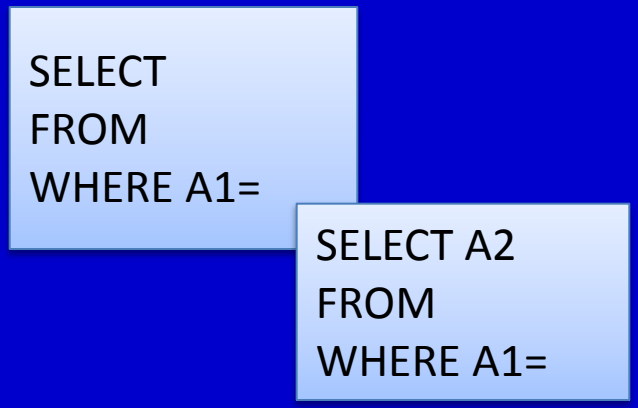
15 Ekim 2014

Nested-Block Features

- Four possible ways to specify nesting

  - classic SQL nesting via the simple predicate structure
  - lower block with a boolean value
  - [NOT] IN construct
  - [NOT] EXIST construct

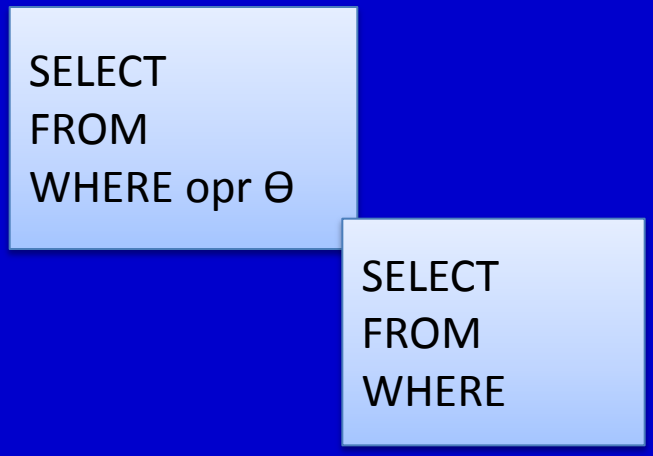
Nested-Block Features-1



- A1 Θ SQL structure
- Example

```
SELECT SNAME
FROM SUPPLIER
WHERE SNO=
        (SELECT SUPPLIER
          FROM SUPPLY
          WHERE ITEM="PAD")
```

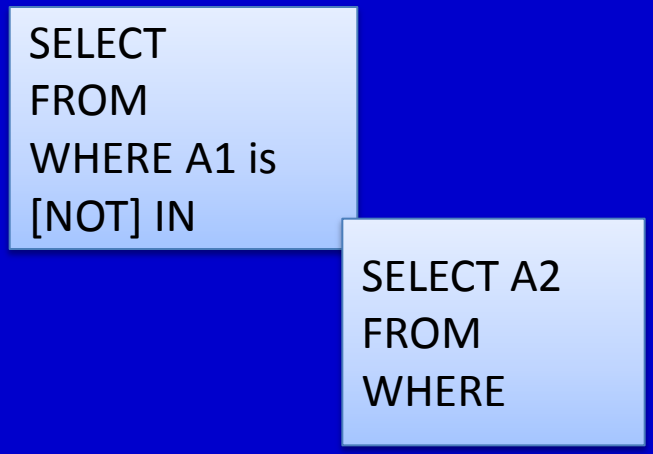
Nested-Block Features-2



- opr Θ SQL structure
- Example

```
SELECT DISTINCT ITEM
FROM SUPPLY SP
WHERE  1<
        (SELECT COUNT(*)
          FROM SUPPLY SPX
          WHERE SP.ITEM=SPX.ITEM)
```

Nested-Block Features-3



- [NOT] IN structure
- Example

```
SELECT SNAME
FROM SUPPLIER
WHERE SNO IS NOT IN
                    (SELECT SUPPLIER
                     FROM SUPPLY
                      WHERE ITEM="PAD")
```

Nested-Block Features-4

```
SELECT
FROM
WHERE
[NOT] EXISTS
```

```
SELECT *
FROM
WHERE A2=A1
```

- [NOT] EXISTS structure
- Example

```
SELECT SNAME
FROM SUPPLIER SP
WHERE NOT EXISTS
      (SELECT *
       FROM SUPPLY S
       WHERE SP.SNO=S.SUPPLIER
       AND ITEM="PAD")
```

Examples-1.1

- Get the names of suppliers who supply pads

```
SELECT SNAME
FROM SUPPLIER
WHERE SNO=
        (SELECT SUPPLIER
          FROM SUPPLY
          WHERE ITEM="PAD")
```

Examples-1.2

- Get the names of suppliers who supply pads

```
SELECT SNAME
 FROM SUPPLIER
 WHERE SNO IS IN
        (SELECT SUPPLIER
          FROM SUPPLY
          WHERE ITEM="PAD")
```

Examples-1.3

- Get the names of suppliers who supply pads

```
SELECT SNAME
FROM SUPPLIER
WHERE SNO EXISTS
        (SELECT *
          FROM SUPPLY S
          WHERE SP.SNO=S.SUPPLIER
              AND ITEM="PAD")
```

Examples-2.1

- Get the names of suppliers supplying items of type A

```
SELECT SNAME FROM SUPPLIER
WHERE SNO IS IN
        (SELECT SUPPLIER FROM SUPPLY
        WHERE ITEM IS IN
                (SELECT INAME FROM ITEM
                                WHERE TYPE="A"))
```

Examples-2.2

- Get the names of suppliers supplying items of type A

```
SELECT SNAME FROM SUPPLIER
WHERE EXISTS
        (SELECT * FROM SUPPLY
          WHERE SUPPLIER.SNO=SUPPLY.SUPPLIER
          AND EXISTS
                    (SELECT * FROM ITEM
                     WHERE SUPPLY.ITEM=ITEM.INAME
                                                    AND TYPE="A"))
```

Examples-2.3

- Get the names of suppliers supplying items of type A (flattened structure)

```
SELECT S.SNAME
FROM SUPPLIER S, SUPPLY SP, ITEM I
WHERE S.SNO=SP.SUPPLIER
AND SP.ITEM=I.INAME
AND I.TYPE="A"
```

Examples-3

- Find the names of items supplied by more than one supplier (Group By)

```
SELECT ITEM
FROM SUPPLY
GROUP BY ITEM
HAVING COUNT(*)>1
```

Set Operations-1.1

- Find the employees who are also managers
- By set operations

SELECT *ENO* FROM EMPLOYEE
INTERSECT
SELECT *MGR* FROM EMPLOYEE

Set Operations-1.2

- Find the employees who are also managers
- By EXISTS keyword

```
SELECT ENO FROM EMPLOYEE E
WHERE EXISTS
        (SELECT * FROM EMPLOYEE M
        WHERE E.MGR=M.ENO)
```

Set Operations-1.3

- Find the employees who are also managers
- By flattened structure

SELECT *E.ENO*
FROM *EMPLOYEE E, EMPLOYEE M*
WHERE *E.MGR=M.ENO*

Set Operations-2.1

- Find the suppliers who does not supply any item
- By set operations

SELECT *SNO* FROM *SUPPLIER*
DIFFERENCE
SELECT *SUPPLIER* FROM *SUPPLY*

Set Operations-2.2

- Find the suppliers who does not supply any item
- By NOT EXISTS keyword

SELECT *SNO* FROM *SUPPLIER*
WHERE NOT EXISTS
(SELECT * FROM *SUPPLY*
WHERE *SUPPLIER.SNO=SUPPLY.SUPPLIER)*

Set Operations-2.3

- Find the suppliers who does not supply any item
- By flattened structure

SELECT *SP.SNO*
FROM *SUPPLIER SP, SUPPLY S*
WHERE *SP.SNO<>S.SUPPLIER*

Updates-1

- Change the type of “Pen” in ITEM to “B” and its color to “Black”

```
UPDATE ITEM
SET TYPE=“B”
        COLOR=“BLACK”
WHERE INAME=“PEN”
```

Updates-2

- Add 5000 to the salary of everyone in “DBSYSTEMS” department

```
UPDATE EMPLOYEE
SET SALARY=SALARY+5000
WHERE DEPT=“DBSYSTEMS”
```

Delete

- Delete item “Cabinet” from ITEM

  DELETE FROM ITEM
  WHERE INAME=“Cabinet”

- Delete all items of type “A”

  DELETE FROM ITEM
  WHERE TYPE=“A”

Examples

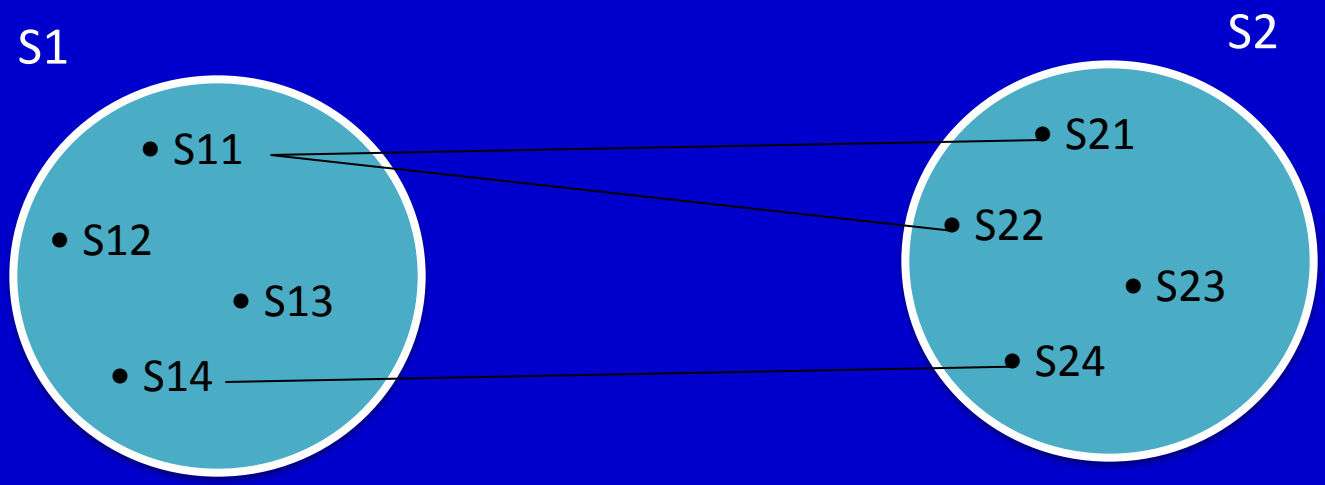
Functional Mapping

Functional Mapping-1

- A functional mapping or function can be defined as $f:D \rightarrow R$ where D is domain and R is range

- If every value in D maps to R, then f is called as *total function*, otherwise f is *partial function*

Functional Mapping-2



- S1 → S2 is Nonfunctional
- S2 → S1 is Partially Functional

Functional Mapping-3

S1		S2	Mapping
One		Many	Nonfunctional
One		One	Functional in either direction
Many		One	Functional
Many		Many	Nonfunctional