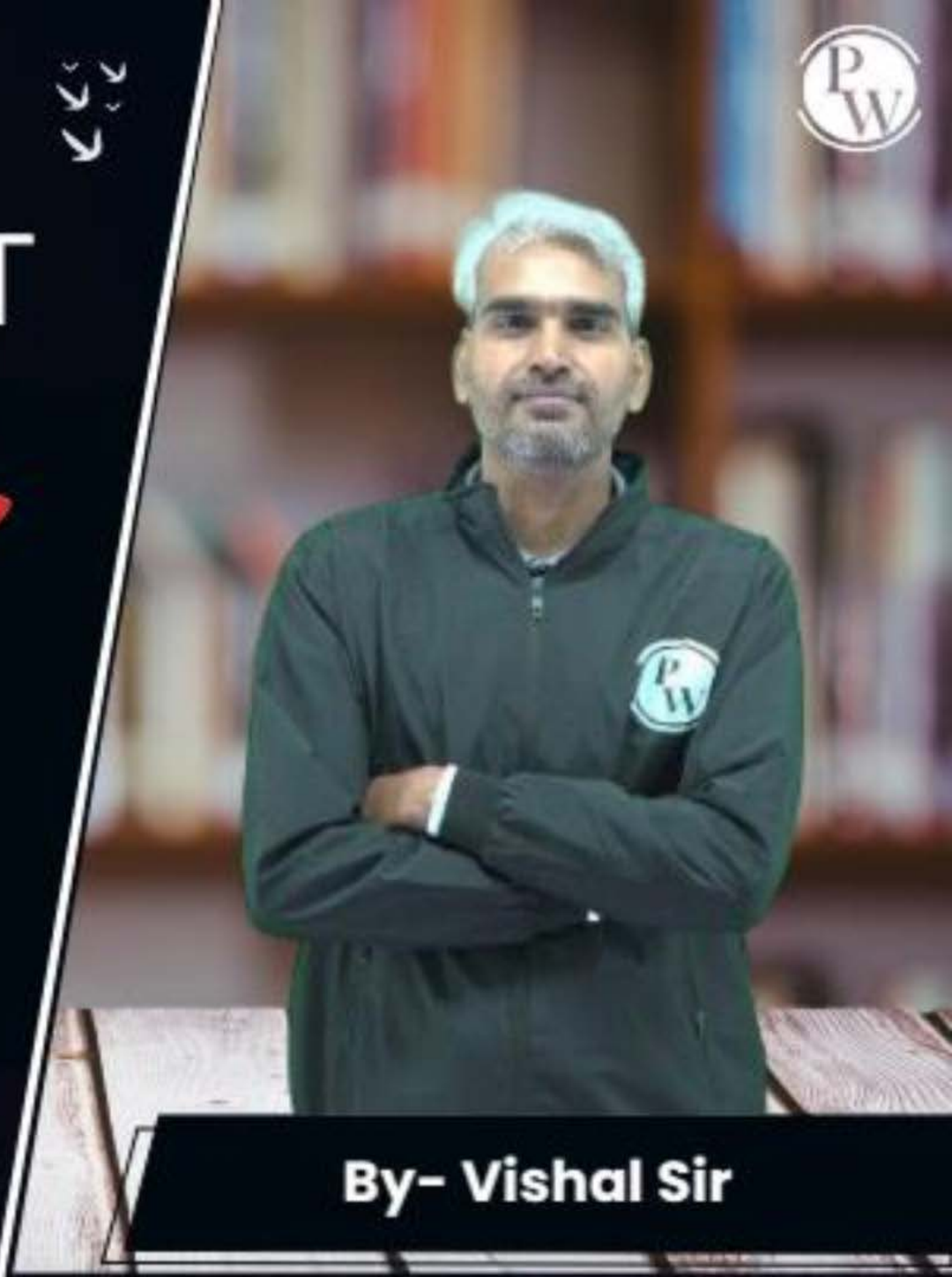
Computer Science & IT

Database Management System

Query Languages

Lecture No. 07



By- Vishal Sir

Recap of Previous Lecture

- Topic: SQL clauses
- Topic: Nested query (sub-query)
- Topic: Operators w.r.t. Nested query
- Topic: Order of execution of nested query

Topics to be Covered

- Topic: Execution of correlated sub-query
- Topic: AS clause
- Topic: WITH clause

Nested Query

✓ Independent Nested query (Independent Sub-query)

* If inner query can be executed independently, then it is called independent Sub-query

✓ Correlated Nested query (Correlated Sub-query)

* When execution of inner query requires value of an attribute from the relation specified in Outer query, then it is called Correlated Sub-query

Nested Query

Independent Nested query (Independent Sub-query)

eg:

```
Select Sid
From Student
Where (Marks) = (Select Max(Marks)
                 From Student)
```

Main query

Inner query

This query can be executed independently
∴ Independent Sub-query

Correlated Nested query (Correlated Sub-query)

eg:

```
Select R.A
From R
Where ..... (Select S.C
             From S
             Where (R.A : S.B))
```

Main query

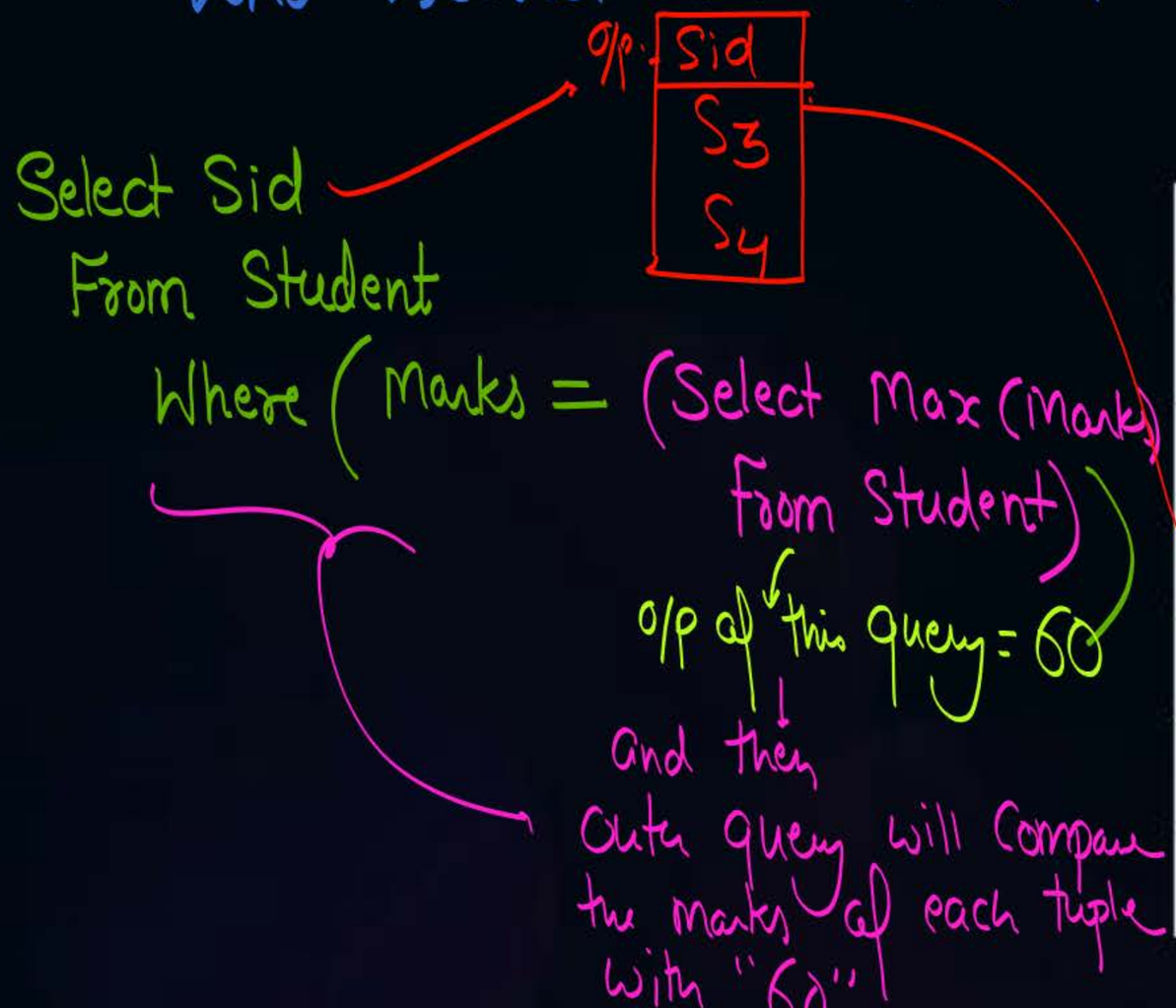
Inner query

Value of attribute 'A' is required from relation R (i.e. from the relation specified in outer query)
∴ Correlated Sub-query

(1) order of execution w.r.t. Independent Sub-query

- Inner query will be executed first And it will produce its output,
  - then outer query will use the output produced by inner query for its execution.

✱ Retrieve Sids of all the students who scored maximum Marks.



Student

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	A	20	IT
S3	B	60	CS
S4	A	60	EC
S5	C	40	IT
S6	C	NULL	EC

Correlated Sub-query :-

operand of this operator is also the o/p of inner query.

Main query

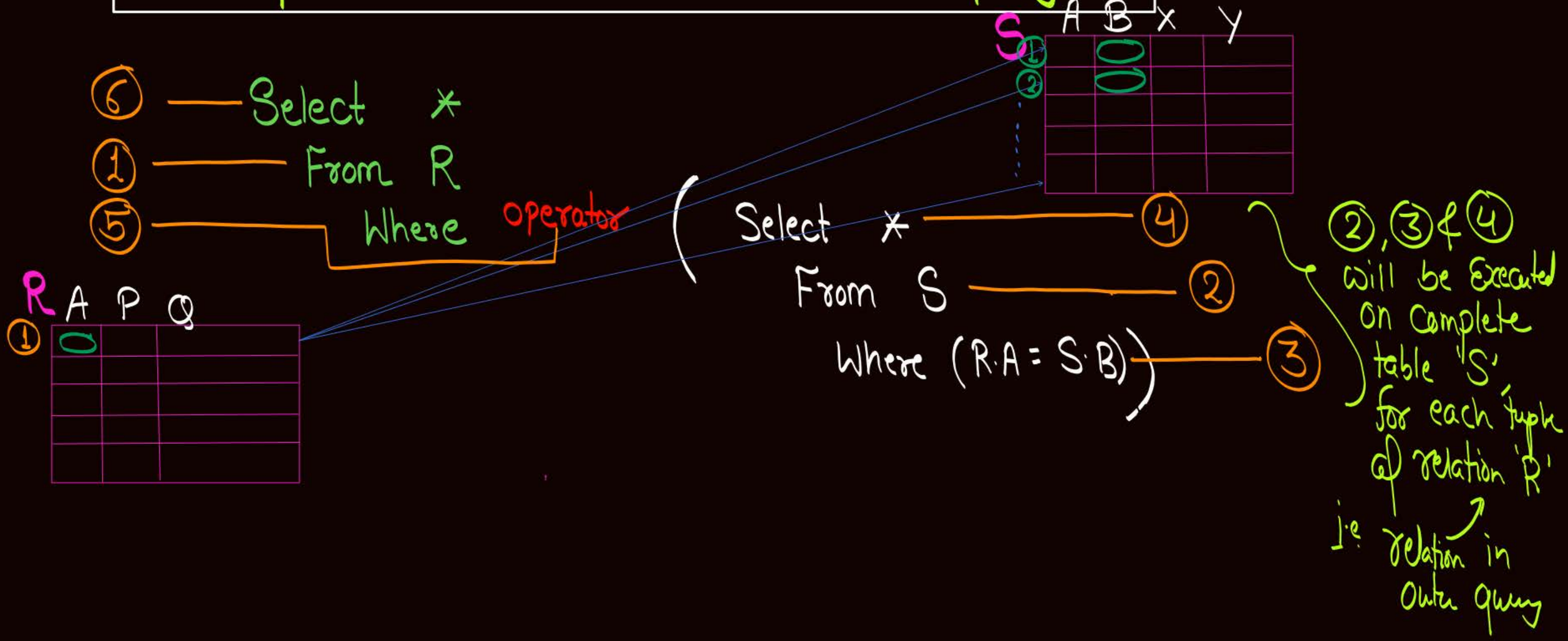
```
Select *
From R
Where operator ( Select *
                 From S
                 Where (R.A = S.B) )
```

Inner query

This operator will return True or False

- If operator returns true, then "Where" Cond^{n} is true
- If operator returns false, then 'Where' Cond^{n} is false

Order of Execution w.r.t. Correlated Subquery :-



Topic : Operators

* IN, ANY, ALL and EXISTS are the operators that are generally used with Sub-query Concept

IN operator

Operator "IN" is used to check whether the concerned tuple is a member of a set of tuples produced by the inner query or not?

Complement of 'IN' is 'NOT IN'

Check whether

X IN { 2, 5, 8, 9, 12, 13 }

Set of tuples Produced by inner query

Our Concerned tuple from outer table

- for eg: Select Eid From EMP

(i) If X = 8, then operator 'IN' will return true

(ii) If X = 10, then operator 'IN' will return false

Check whether

X IN {(1, A), (3, D), (4, D), (7, F)}

"tuple from outer table"

for eg

Select Sid, Sname
From Students

(i) If X = (4, D), then IN returns True

(ii) If X = (F, 7), then IN returns false.

≠ (7, F)

Note :-

If set of tuples is Empty {i.e, o/p produced by inner query is Empty}, then operator 'IN' will always return False

and hence operator 'NOT IN' will always return true if inner query result is Empty.

Q: Consider the following relational scheme

Supplier (Sid, Sname, Rating)

Parts (Pid, Pname, Color)

Catalog (Sid, Pid, Cost)

* Retrieve Sids of suppliers who supplied some Red Color Parts

Parts

P_1	Red
P_2	Green
P_3	Red

Catalog

S_1	P_1
S_1	P_2
S_2	P_2
S_3	P_2
S_3	P_3

R.A:-

$$\pi_{Catalog.Sid}\left(\sigma_{Catalog.Pid = Parts.Pid \wedge Parts.Color = 'Red'}(Catalog \times Parts)\right)$$

SQL: (Without IN)

```
Select C.sid
From Catalog C , Parts P
Where (C.Pid = P.Pid AND P.Color = 'Red')
```

Catalog renamed as C

Parts renamed as P

SQL: (Using IN)

```
Select Sid From Catalog
Where Pid IN ( Select Pid From Parts
               Where Color = 'Red' )
```

'ANY' operator

Operator 'ANY' is used along with Comparison operators

Check whether X=10 ✗ ✗ ✗ ✗ ✓

$X < \text{ANY} \{ 2, 5, 8, 9, 12, 14 \}$

X=20 ✗ ✗ ✗ ✗ ✗ ✗

if $X = 10$, then "ANY" will return True

if $X = 20$, then "ANY" will return False

'ANY' operator

Operator 'ANY' is used along with Comparison operators

$<, \leq, >, \geq, =, <>$

Note: Operator 'ANY' will return True, if and only if at least one tuple in the set of tuples satisfy the Comparison Condition

Note: If set of tuples is empty, then operator 'ANY' will always return False

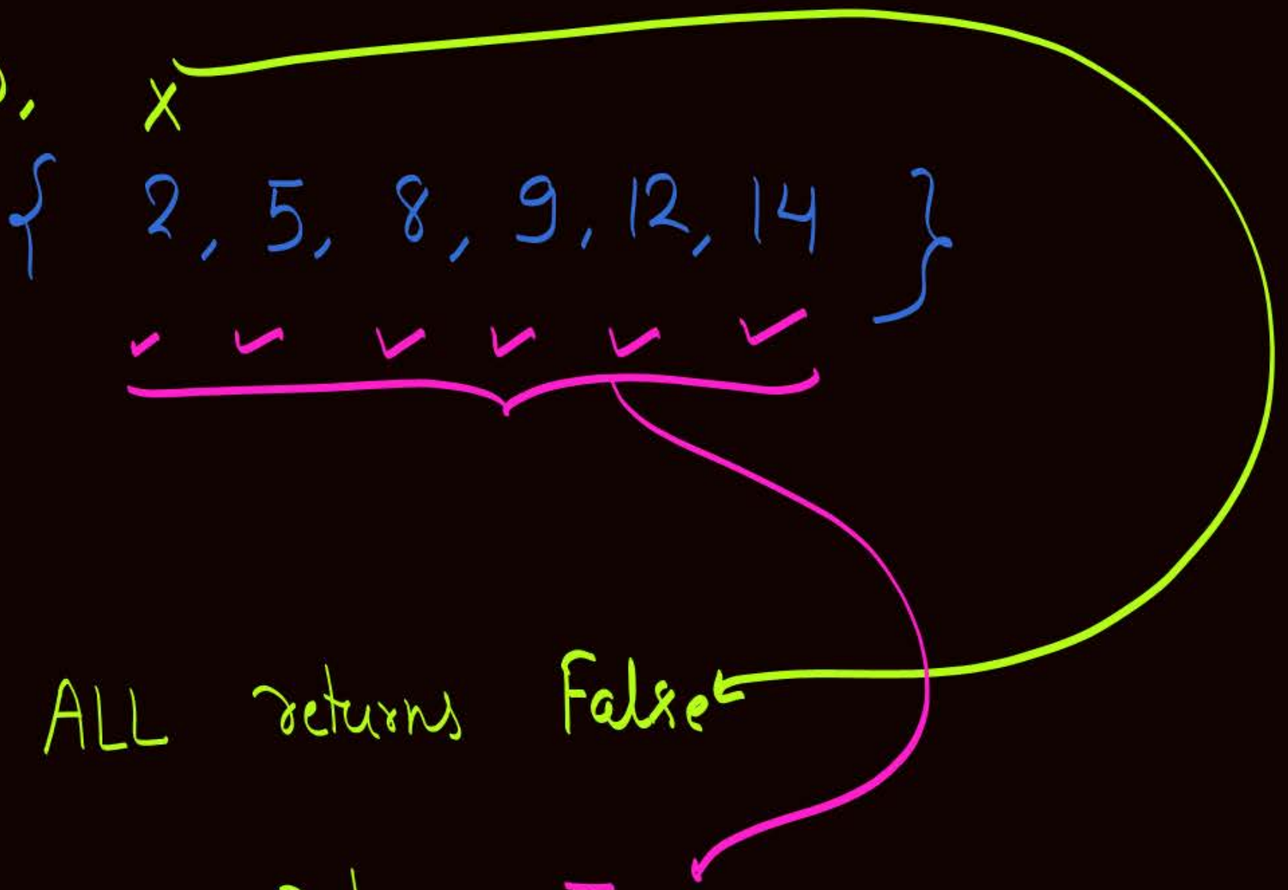
'ALL' operator

Operator 'ALL' is also used along with Comparison operators

$<, \leq, >, \geq, =, <>$

Check whether $x=10$, $x < \text{ALL } \{2, 5, 8, 9, 12, 14\}$

$x=1$



if $x = 10$, then ALL returns False

if $x = 1$, then ALL return True

'ALL' operator

Operator 'ALL' is also used along with Comparison operators

$<, \leq, >, \geq, =, <>$

- operator 'ALL' can return false if and only if at least one tuple in the set of tuples fails the Comparison Condition
- If set of tuples is Empty, then operator 'ALL' will always return True

EXISTS operator

- EXISTS returns True if and only if inner query result is not Empty.

- If inner query result is not Empty, then EXISTS will return True

&

- If inner query result is Empty, then EXISTS will return False.

Complement of 'EXISTS' is 'NOT EXISTS'

Today's Topic

H.W.

#e.g.

```
SELECT C.sid
FROM Catalog C
WHERE EXISTS (SELECT *
              FROM Parts P
              WHERE (P.Pid=C.Pid AND P.color='RED'))
```

Correlated Sub-query

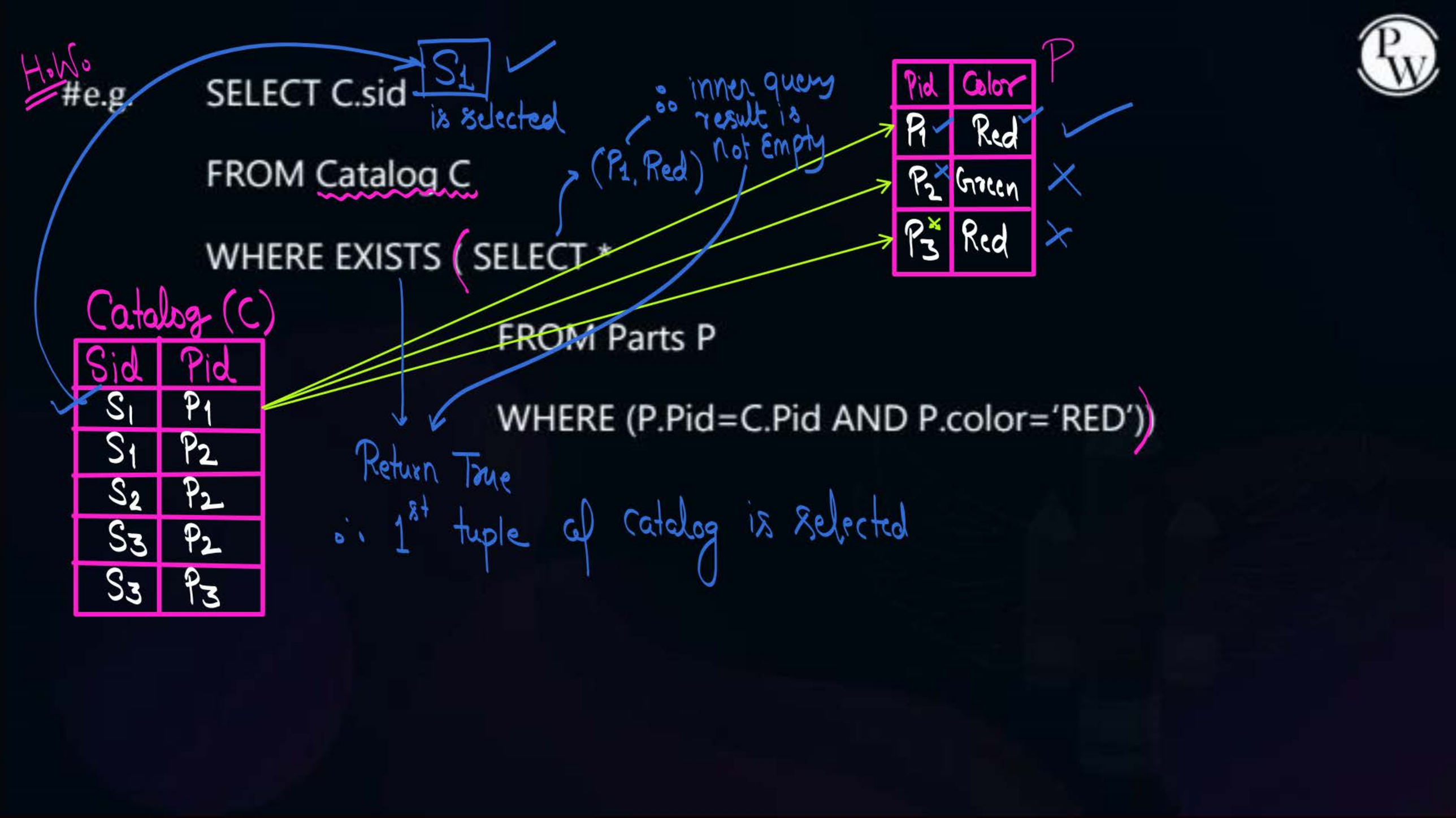
Parts

Pid	Color
P_1	Red
P_2	Green
P_3	Red

Catalog

Sid	Pid
S_1	P_1
S_1	P_2
S_2	P_2
S_2	P_1
S_3	P_3

What output will be produced by the above query if it is executed on the given instances of Parts & Catalog table


H.W.
#e.g.
SELECT C.sid
FROM Catalog C
WHERE EXISTS (SELECT *
FROM Parts P
WHERE (P.Pid=C.Pid AND P.color='RED'))
S1
is selected
(P1, Red)
∵ inner query result is not Empty
P
Pid
Color
P1
Red
P2
Green
P3
Red
Catalog (C)
Sid
Pid
S1
P1
S1
P2
S2
P2
S3
P2
S3
P3
Return True
∴ 1st tuple of catalog is selected

H.W.

#e.g.

```
SELECT C.sid
FROM Catalog C
WHERE EXISTS ( SELECT *
               FROM Parts P
               WHERE (P.Pid=C.Pid AND P.color='RED'))
```

S_1 is selected

w.r.t 2^{nd} tuple of Catalog o/p is Empty

Catalog (C)

Sid	Pid
S_1	P_1
S_1	P_2
S_2	P_2
S_3	P_2
S_3	P_3

P

Pid	Color
P_1	Red
P_2	Green
P_3	Red

Return False

$\therefore$ 2^{nd} tuple of catalog is not selected

H.W.

#e.g.

```
SELECT C.sid
FROM Catalog C
WHERE EXISTS ( SELECT *
               FROM Parts P
               WHERE (P.Pid=C.Pid AND P.color='RED'))
```

S_1 is selected

w.r.t 3rd tuple of Catalog o/p is Empty

Catalog (C)

Sid	Pid
S_1	P_1
S_1	P_2
S_2	P_2
S_3	P_2
S_3	P_3

P

Pid	Color
P_1	Red
P_2	Green
P_3	Red

Return False

∴ 3rd tuple of catalog is not selected

H.W.

#e.g.

```
SELECT C.sid
FROM Catalog C
WHERE EXISTS ( SELECT *
               FROM Parts P
               WHERE (P.Pid=C.Pid AND P.color='RED'))
```

S_1 is selected

w.r.t 4th tuple of Catalog o/p is Empty

Catalog (C)

Sid	Pid
S_1	P_1
S_1	P_2
S_2	P_2
S_3	P_2
S_3	P_3

P

Pid	Color
P_1	Red
P_2	Green
P_3	Red

Return False

∴ 4th tuple of catalog is not selected

H.W.

#e.g.

```
SELECT C.sid
FROM Catalog C
WHERE EXISTS ( SELECT *
               FROM Parts P
               WHERE (P.Pid=C.Pid AND P.color='RED'))
```

S_1 is selected

S_3 is selected

o/p: (P_3, Red) not Empty

Catalog (C)

Sid	Pid
S_1	P_1
S_1	P_2
S_2	P_2
S_3	P_2
S_3	P_3

P

Pid	Color
P_1	Red
P_2	Green
P_3	Red

Return True

5^{th} tuple of catalog is selected

H.W.

#e.g.

```
SELECT C.sid
FROM Catalog C
WHERE EXISTS ( SELECT *
               FROM Parts P
               WHERE (P.Pid=C.Pid AND P.color='RED'))
```

Catalog

Sid	Pid
S_1	P_1
S_1	P_2
S_2	P_2
S_3	P_2
S_3	P_3

Pid	Color
P_1	Red
P_2	Green
P_3	Red

final op :-

Sid
S_1
S_3

H.W.

#e.g.

```
SELECT C1.sid
FROM Catalog C1
WHERE NOT EXISTS (SELECT P.Pid
                  FROM Parts P
                  WHERE NOT EXISTS (SELECT C2.Sid
                                    FROM Catalog C2
                                    WHERE(C2.Pid=P.Pid AND  C2.Sid=C1.Sid)))
```

What output is produced by above SQL query:

A. Sids of suppliers who supplied some parts

B. Sids of suppliers who supplied only proper subset of parts from all parts

C. Sids of suppliers who supplied all parts

D. Sids of suppliers who did not supply any part

#e.g. HW:

```
SELECT C1.sid
FROM Catalog C1
WHERE NOT EXISTS ( SELECT P.Pid
                   FROM Parts P
                   WHERE NOT EXISTS (SELECT C2.Sid
                                     FROM Catalog C2
                                     WHERE(C2.Pid=P.Pid AND  C2.Sid=C1.Sid)))
```

Catalog

Sid	Pid
S_1	P_1
S_2	P_2
S_1	P_2

Parts

Pid
P_1
P_2

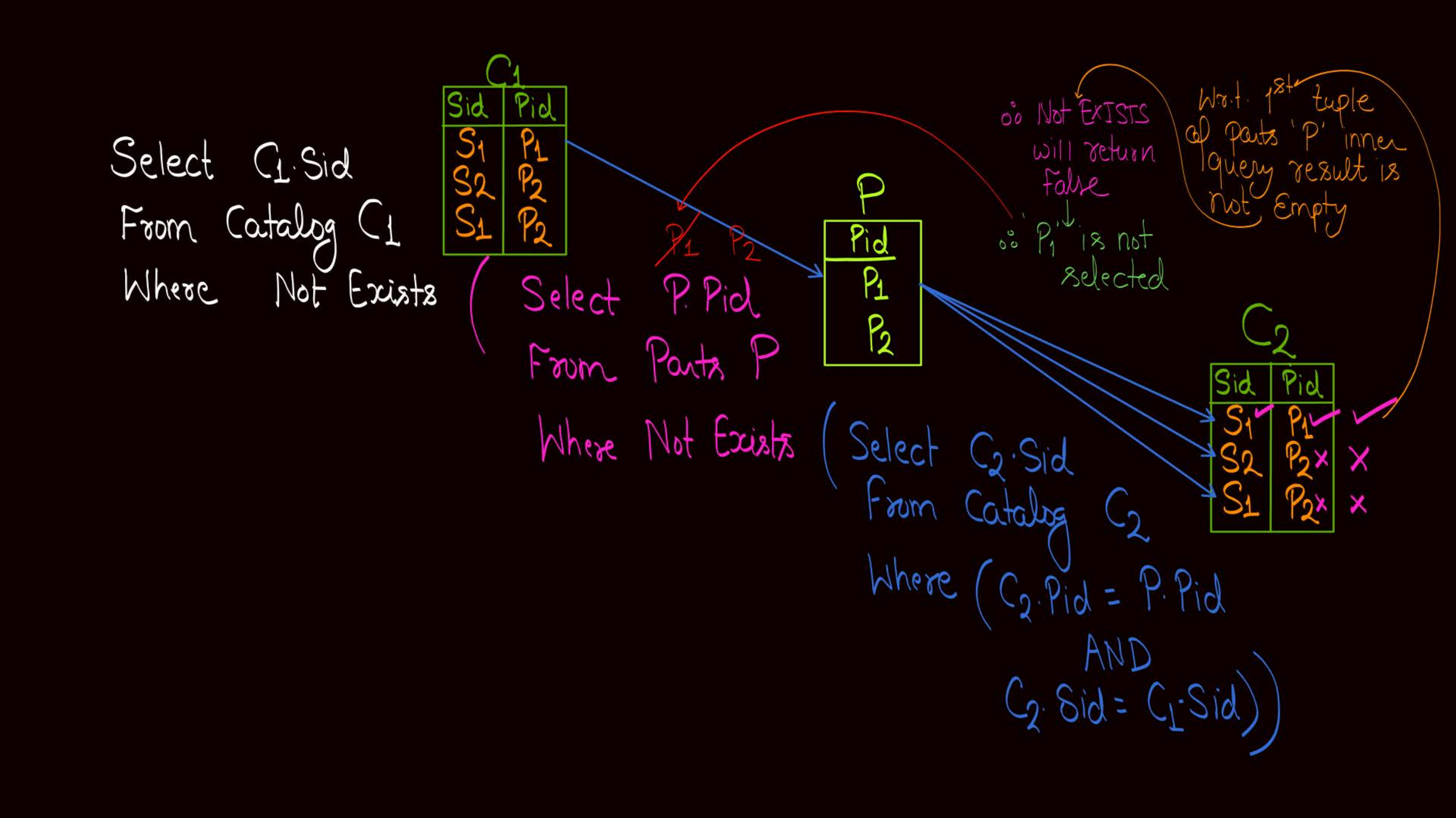
What output is produced by above SQL query:

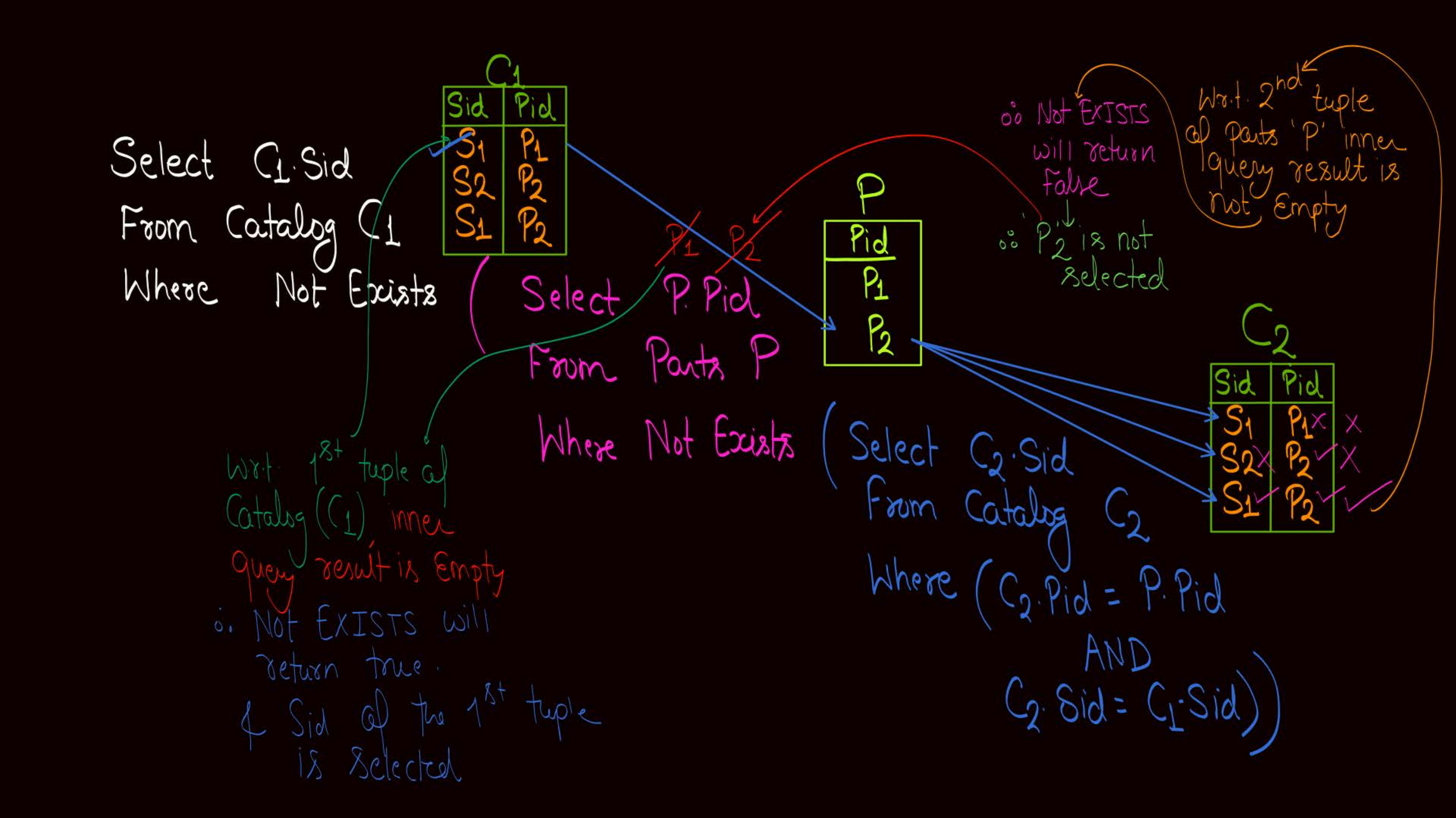
A. $\{S_1, S_2\}$ Sids of suppliers who supplied some parts

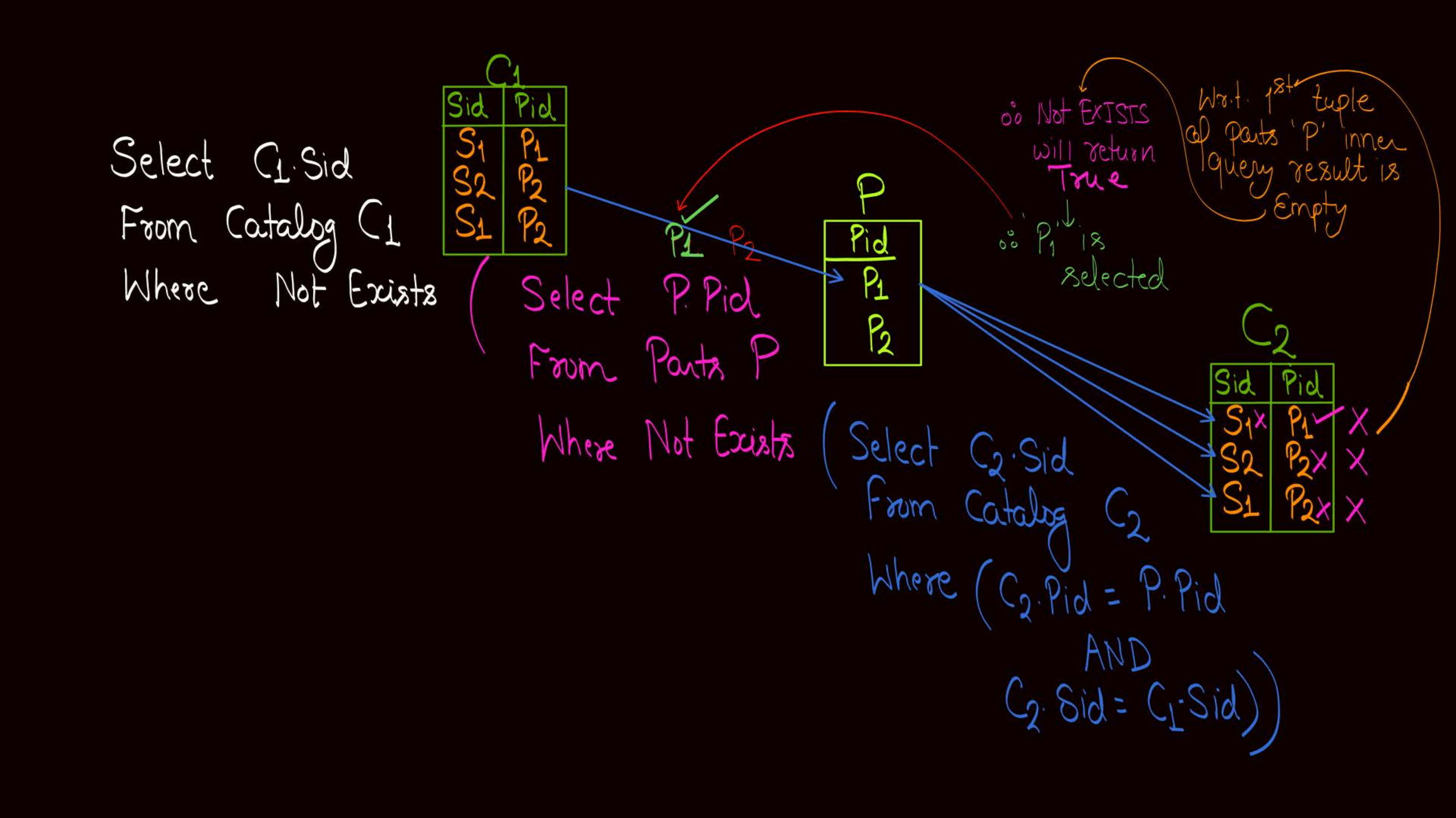
B. $\{S_2\}$ Sids of suppliers who supplied only proper subset of parts from all parts

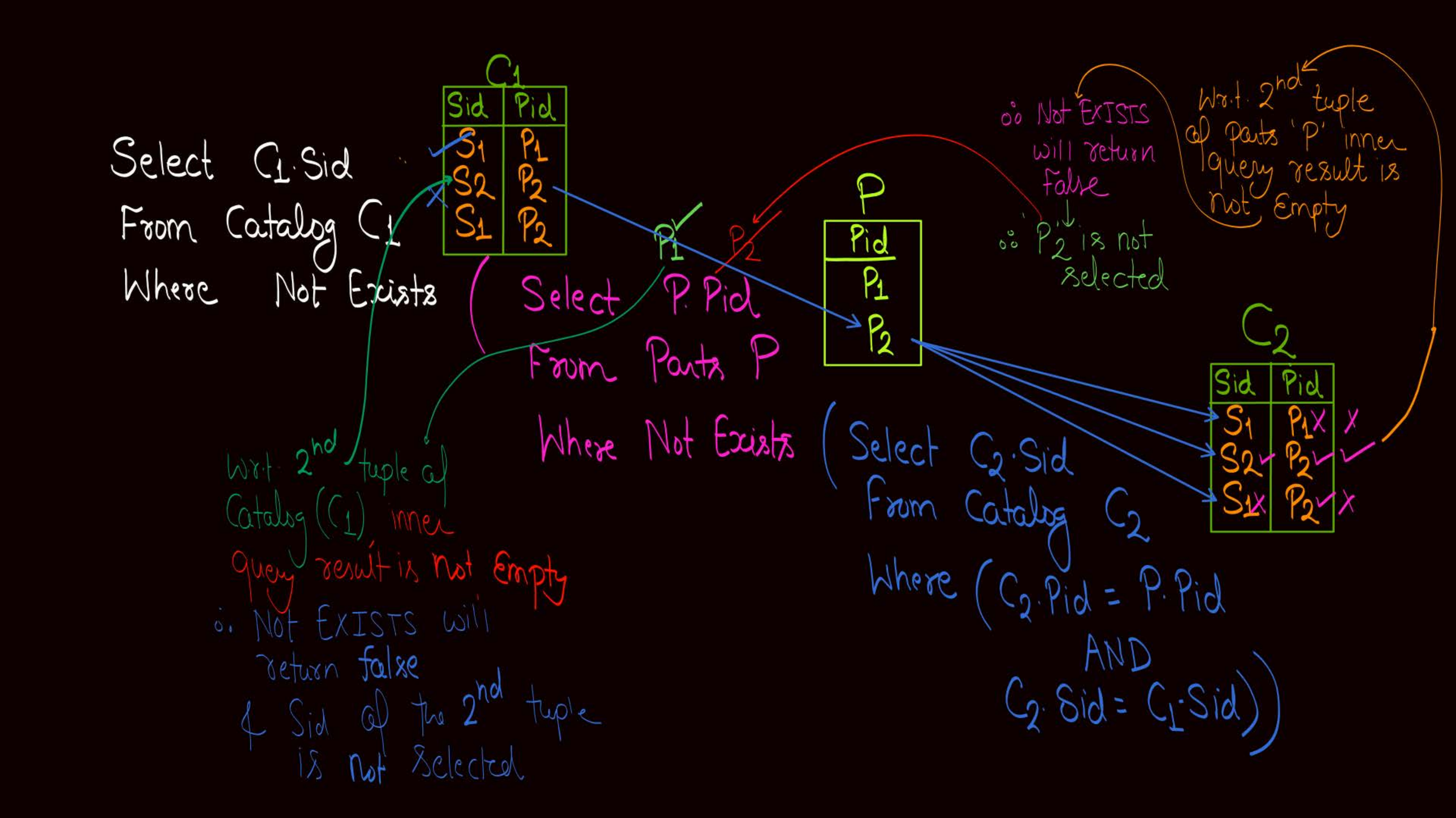
C. $\{S_1\}$ Sids of suppliers who supplied all parts

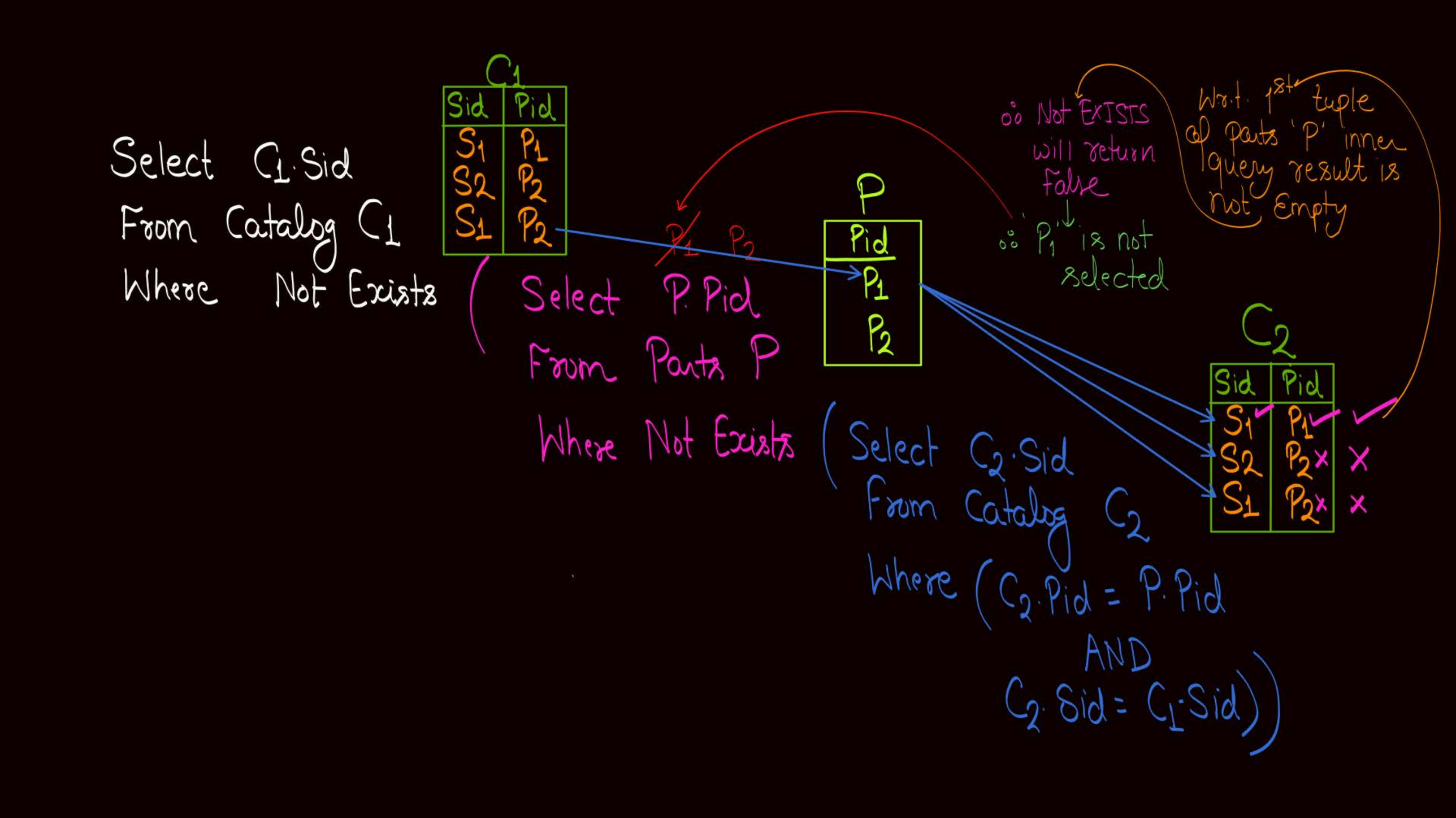
~~D.~~ Sids of suppliers who did not supply any part

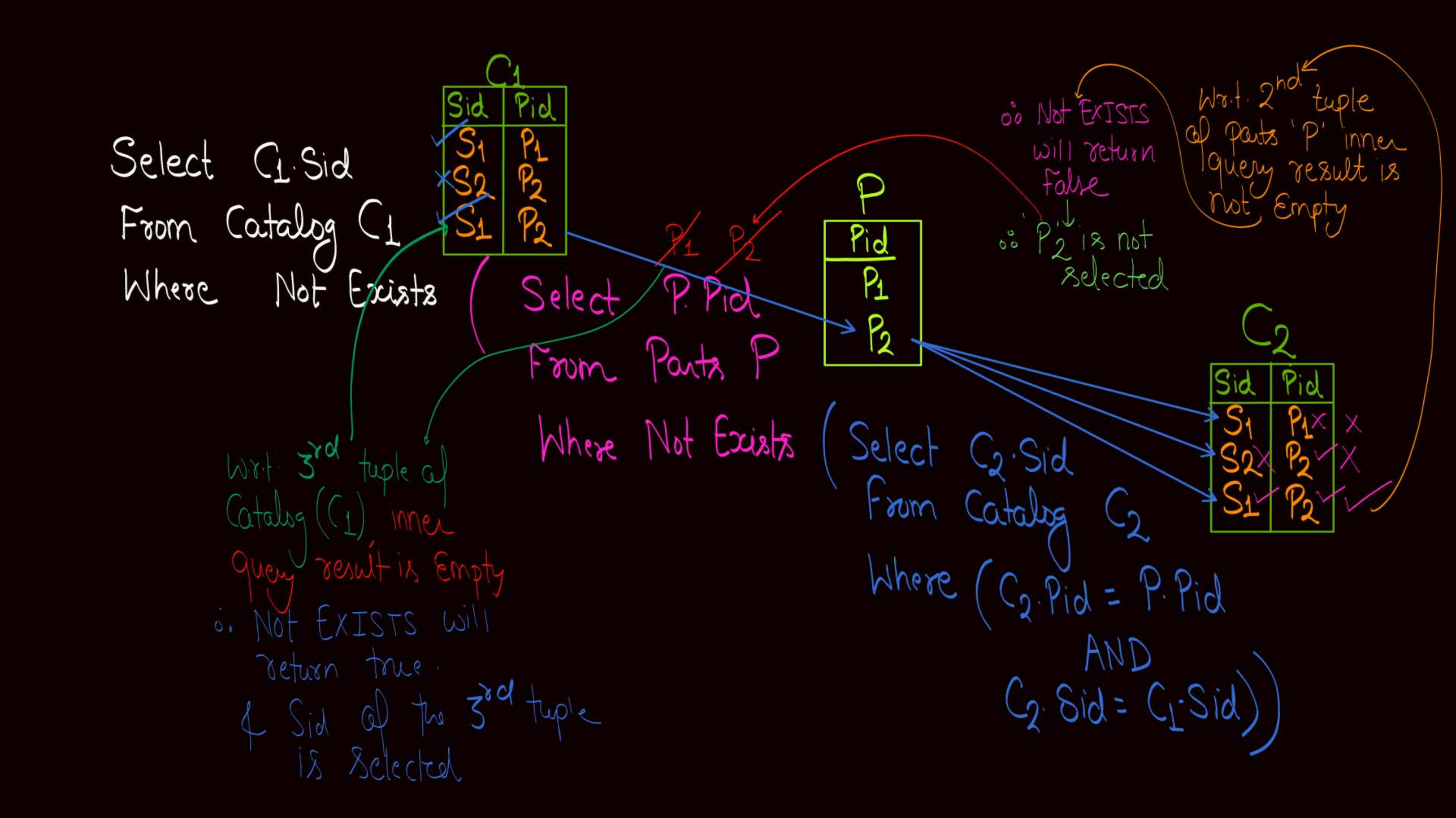

Select C1.Sid
From Catalog C1
Where Not Exists
C1
Sid Pid
S1 P1
S2 P2
S1 P2
P1 P2
Select P.Pid
From Parts P
Where Not Exists
P
Pid
P1
P2
∴ Not EXISTS will return false
∴ P1 is not selected
Wr.t. 1st tuple of Parts 'P' inner query result is not Empty
Select C2.Sid
From Catalog C2
Where (C2.Pid = P.Pid
AND
C2.Sid = C1.Sid))
C2
Sid Pid
S1 P1
S2 P2 ×
S1 P2 ×


Select C1.Sid
From Catalog C1
Where Not Exists
C1
Sid Pid
S1 P1
S2 P2
S1 P2
Select P.Pid
From Parts P
Where Not Exists
P
Pid
P1
P2
Select C2.Sid
From Catalog C2
Where (C2.Pid = P.Pid
AND
C2.Sid = C1.Sid))
C2
Sid Pid
S1 P1
S2 P2
S1 P2
Wrt 1st tuple of Catalog (C1) inner query result is Empty
∴ Not EXISTS will return true.
& Sid of the 1st tuple is selected
∴ Not EXISTS will return false
∴ P2 is not selected
Wrt 2nd tuple of Parts 'P' inner query result is not Empty


Select C1.Sid
From Catalog C1
Where Not Exists
C1
Sid
Pid
S1
P1
S2
P2
S1
P2
P1
P2
P
Pid
P1
P2
∴ Not EXISTS will return True
∴ P1 is selected
Wrt 1st tuple of Parts 'P' inner query result is Empty
C2
Sid
Pid
S1
P1
S2
P2
S1
P2
X
X
X
(Select P.Pid
From Parts P
Where Not Exists (Select C2.Sid
From Catalog C2
Where (C2.Pid = P.Pid
AND
C2.Sid = C1.Sid))


Select C1.Sid
From Catalog C1
Where Not Exists
C1
Sid
Pid
S1
P1
S2
P2
S1
P2
(Select P.Pid
From Parts P
Where Not Exists (Select C2.Sid
From Catalog C2
Where (C2.Pid = P.Pid
AND
C2.Sid = C1.Sid))
P
Pid
P1
P2
C2
Sid
Pid
S1
P1
S2
P2
S1
P2
w.r.t 2nd tuple of Catalog (C1) inner query result is not Empty
∴ Not EXISTS will return false
& Sid of the 2nd tuple is not Selected
∴ Not EXISTS will return false
∴ P2 is not selected
w.r.t 2nd tuple of Parts 'P' inner query result is not Empty


Select C1.Sid
From Catalog C1
Where Not Exists
C1
Sid Pid
S1 P1
S2 P2
S1 P2
P1 P2
P
Pid
P1
P2
∴ Not EXISTS will return false
∴ P1 is not selected
Wrt 1st tuple of Parts 'P' inner query result is not Empty
C2
Sid Pid
S1 P1
S2 P2 ×
S1 P2 ×
Select P.Pid
From Parts P
Where Not Exists (Select C2.Sid
From Catalog C2
Where (C2.Pid = P.Pid
AND
C2.Sid = C1.Sid))


Select C1.Sid
From Catalog C1
Where Not Exists
C1
Sid
Pid
S1
P1
S2
P2
S1
P2
Select P.Pid
From Parts P
Where Not Exists
P1
P2
P
Pid
P1
P2
Select C2.Sid
From Catalog C2
Where (C2.Pid = P.Pid
AND
C2.Sid = C1.Sid))
C2
Sid
Pid
S1
P1
S2
P2
S1
P2
Wrt 3rd tuple of Catalog (C1) inner query result is Empty
∴ Not EXISTS will return true.
& Sid of the 3rd tuple is selected
∴ Not EXISTS will return false
∴ P2 is not selected
Wrt 2nd tuple of Parts 'P' inner query result is not Empty

C_1

Sid	Pid
S_1	P_1
S_2	P_2
S_1	P_2

Select C_1.Sid
From Catalog C_1
Where Not Exists (Select P.Pid
From Parts P
Where Not Exists (Select C_2.Sid
From Catalog C_2
Where (C_2.Pid = P.Pid
AND
C_2.Sid = C_1.Sid))

P

Pid
P_1
P_2

C_2

Sid	Pid
S_1	P_1
S_2	P_2
S_1	P_2

Final o/p =

Sid
S_1
S_1

H.W.

#e.g.

```
SELECT C1.sid
FROM Catalog C1
WHERE NOT EXISTS ( SELECT P.Pid
                   FROM Parts P
                   WHERE NOT EXISTS (SELECT C2.Sid
                                     FROM Catalog C2
                                     WHERE(C2.Pid=P.Pid AND  C2.Sid=C1.Sid)))
```

Catalog

Sid	Pid
S_1	P_1
S_2	P_2
S_1	P_2

Parts

Pid
P_1
P_2

What output is produced by above SQL query:

A. Sids of suppliers who supplied some parts

B. Sids of suppliers who supplied only proper subset of parts from all parts

C. Sids of suppliers who supplied all parts

D. Sids of suppliers who did not supply any part

Topic : AS clause

- AS clause is used to rename a column or table with an alias.
- An alias only exists for the duration of the query.

{Note: Using 'AS' we can create an alias for almost Everything}

#e.g. GATE-PYQ

Consider a database that has the relation schema EMP (EmpId, EmpName, and DeptName).

An instance of the schema EMP and a SQL query on it are given below:

EmpId	Emp Name	DeptName
1	XYA	AA
2	XYB	AA
3	XYC	AA
4	XYD	AA
5	XYE	AB
6	XYF	AB
7	XYG	AB
8	XYH	AC
9	XYI	AC
10	XYJ	AC
11	XYK	AD
12	XYL	AD
13	XYM	AE

```
SELECT AVG(EC.Num)
FROM EC
WHERE (DeptName, Num) IN
    (SELECT DeptName, COUNT(EmpId) AS EC(DeptName, Num)
     FROM EMP
     GROUP BY DeptName)
```

Main Query

aggregate function

The output of executing the SQL query is _____

Independent Sub-query ∴ it will be executed first

#e.g. GATE-PYQ

Consider a database that has the relation schema EMP (EmpId, EmpName, and DeptName).

An instance of the schema EMP and a SQL query on it are given below:

EmpId	Emp Name	DeptName
1	XYA	AA
2	XYB	AA
3	XYC	AA
4	XYD	AA
5	XYE	AB
6	XYF	AB
7	XYG	AB
8	XYH	AC
9	XYI	AC
10	XYJ	AC
11	XYK	AD
12	XYL	AD
13	XYM	AE

```
SELECT AVG(EC.Num)
FROM EC
WHERE (DeptName, Num) IN
    (SELECT DeptName, COUNT(EmpId) AS EC(DeptName, Num)
        FROM EMP
        GROUP BY DeptName)
```

Main Query

o/p is given an alias using

The output of executing the SQL query is _____

name given to the table w.r.t. o/p

new name w.r.t first attribute , new name w.r.t 2nd attribute

its a new schema

#e.g. Consider a database that has the relation schema EMP (EmpId, EmpName, and DeptName).
An instance of the schema EMP and a SQL query on it are given below:

```
SELECT AVG(EC.Num)
FROM EC
WHERE (DeptName, Num) IN
    (SELECT DeptName, COUNT(EmpId) AS EC(DeptName, Num)
                                    FROM EMP
                                    GROUP BY DeptName)
```

The output of executing the SQL query is _____

- inner query
- it will be executed first.
- And because of the Execution of this query a new Schema "EC (DeptName, Num)" will come in to the picture.
- then Main query will execute w.r.t. table EC (Deptname, Num)

#e.g. Consider a database that has the relation schema EMP (EmpId, EmpName, and DeptName).
An instance of the schema EMP and a SQL query on it are given below:

```
SELECT AVG(EC.Num)
FROM EC
WHERE (DeptName, Num) IN
  (SELECT DeptName, COUNT(EmpId) AS EC(DeptName, Num)
   FROM EMP
   GROUP BY DeptName)
```

EC

DeptName	Num
DeptName	Count(EmpID)
A A	4
A B	3
A C	3
A D	2
A E	1

The output of executing the SQL query is _____

#e.g. Consider a database that has the relation schema EMP (EmpId, EmpName, and DeptName).
An instance of the schema EMP and a SQL query on it are given below:

$\frac{SUM(EC.Num)}{Count(EC.Num)} = \frac{13}{5} = 2.6$

```
SELECT AVG(EC.Num)
FROM EC
WHERE (DeptName, Num) IN
    (SELECT DeptName, COUNT(EmpId) AS EC(DeptName, Num)
     FROM EMP
     GROUP BY DeptName)
```

The output of executing the SQL query is _____

EC

DeptName	Num
AA	4
AB	3
AC	3
AD	2
AE	1

IN (True)
IN (True)
IN (True)
IN (True)
IN (True)

Set of tuple produced by this query is

{(AA,4), (AB,3), (AC,3), (AD,2), (AE,1)}

#e.g. Consider a database that has the relation schema EMP (EmpId, EmpName, and DeptName).
An instance of the schema EMP and a SQL query on it are given below:

```
SELECT AVG(EC.Num)
FROM EC
WHERE (DeptName, Num) IN
    (SELECT DeptName, COUNT(EmpId) AS EC(DeptName, Num)
                                    FROM EMP
                                    GROUP BY DeptName)
```

The output of executing the SQL query is_____

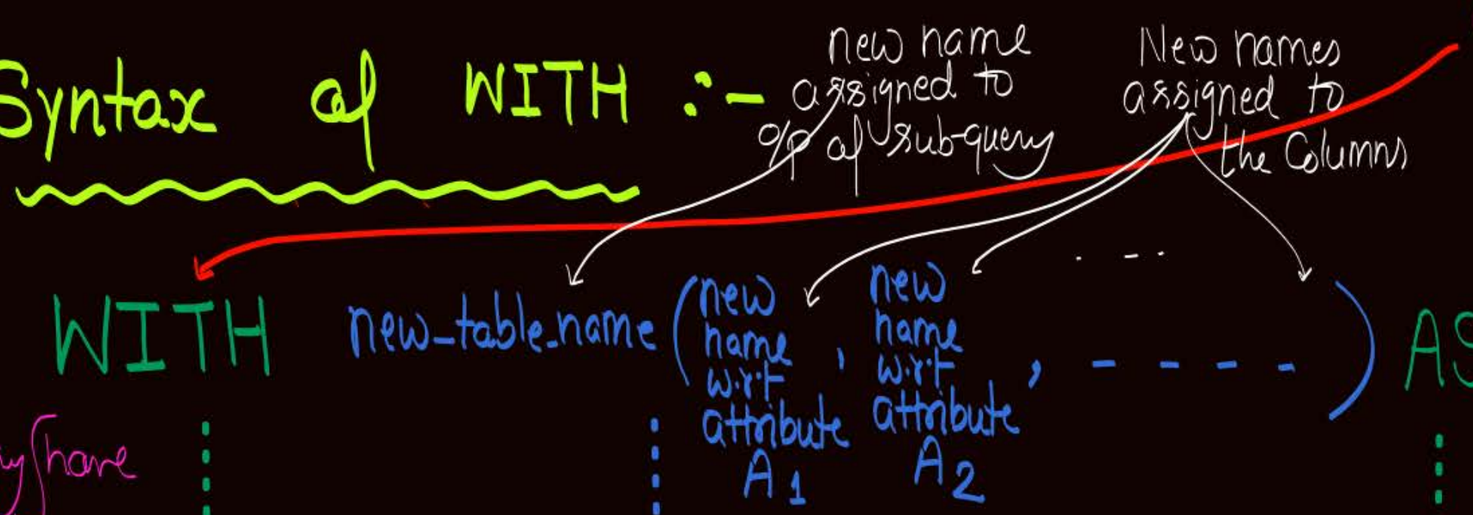
Ans = 2.6

Topic : WITH clause

WITH is used along with AS

The WITH Clause is mainly used to provide a subquery block a name that can be referenced by the main SQL query or the subqueries that follows.

Syntax of WITH :-



o/p of this subquery will be recognized by a "new schema" defined using WITH clause

Sub-query

o/p =

A_1	A_2	. . .

We may have multiple sub-query using "WITH" clause → all of them will execute in a sequence one after another

In the end we will have Main query

Select - - - -
- - - - ← Main query
- - - -

#e.g. Consider the following database table named water_scheme

GATE PYQ

water_scheme		
scheme_no	district_name	capacity
1	Ajmer	20
1	Bikaner	10
2	Bikaner	10
3	Bikaner	20
1	Churu	10
2	Churu	20
1	Dungargarh	10

order of execution

The number of tuples returned by the following SQL query is_____

①
```
with total(name, capacity) as
        select district_name, sum(capacity)
        from water_schemes
        group by district_name
```

②
```
with total_avg(capacity) as
        select avg(capacity)
        from total
```

③
```
select name from total, total_avg
        where total.capacity >= total_avg.capacity
```

Main query

#e.g. Consider the following database table named water_scheme

GATE PYQ

water_scheme		
scheme_no	district_name	capacity
1	Ajmer	20
1	Bikaner	10
2	Bikaner	10
3	Bikaner	20
1	Churu	10
2	Churu	20
1	Dungargarh	10

order of execution

The number of tuples returned by the following SQL query is_____

```
(4) with total(name, capacity) as
        (3) select district_name, sum(capacity)
        (1) from water_schemes
        (2) group by district_name
(7) with total_avg(capacity) as
        (6) select avg(capacity)
        (5) from total
(10) select name from total, total_avg
        (8) where total.capacity >= total_avg.capacity
        (9)
```

Subquery

Subquery

#e.g. Consider the following database table named water_scheme

total

name Capacity

district_name	Sum (Capacity)
Ajmer	20
Bikaner	40
Churu	30
Dungargarh	10

The number of tuples returned by the following SQL query is_____

```
with total(name, capacity) as
        select district_name, sum(capacity)
        from water_schemes
        group by district_name
with total_avg(capacity) as
        select avg(capacity)
        from total
select name from total, total_avg
        where total.capacity >= total_avg.capacity
```

#e.g. Consider the following database table named water_scheme

total

district_name	Sum(Capacity)
Ajmer	20
Bikaner	40
Churu	30
Dungargarh	10

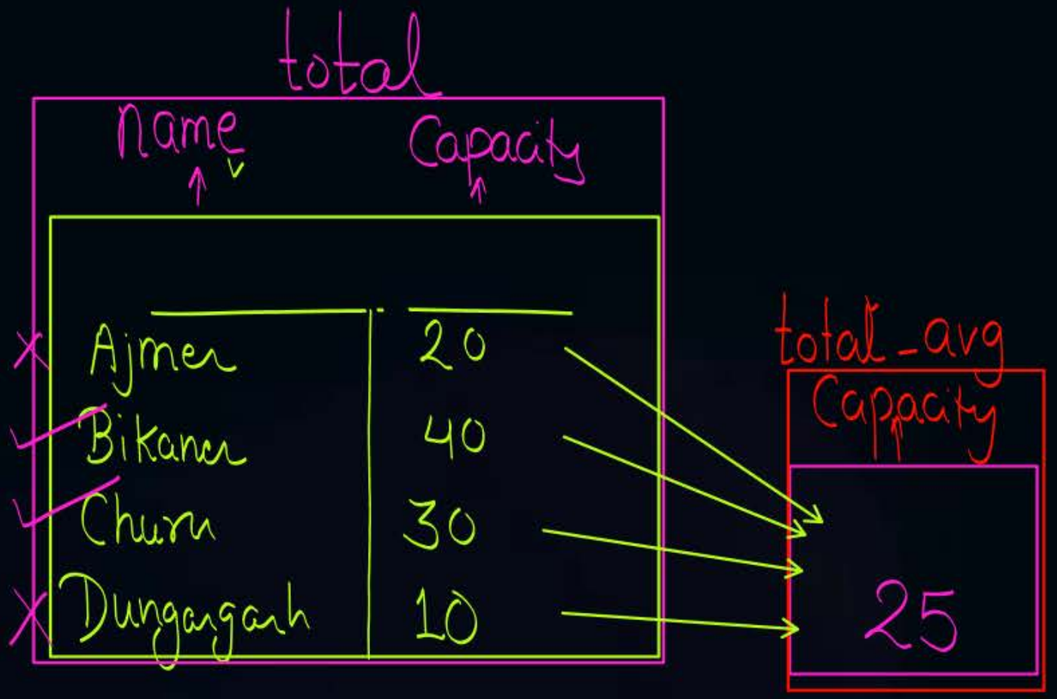
total_avg

Avg(Capacity)
25

The number of tuples returned by the following SQL query is_____

```
with total(name, capacity) as
        select district_name, sum(capacity)
        from water_schemes
        group by district_name
with total_avg(capacity) as
        select avg(capacity)
        from total
select name from total, total_avg
        where total.capacity >= total_avg.capacity
```

#e.g. Consider the following database table named water_scheme



The number of tuples returned by the following SQL query is_____

```
with total(name, capacity) as
        select district_name, sum(capacity)
        from water_schemes
        group by district_name
with total_avg(capacity) as
        select avg(capacity)
        from total
select name from total, total_avg
        where total.capacity >= total_avg.capacity
```

o/p=

name
Bikaner
Churu

=> # tuples in o/p = 2

2 mins Summary

Topic Execution of correlated sub-query

Topic AS clause

Topic WITH clause

Slide

THANK - YOU

Slide