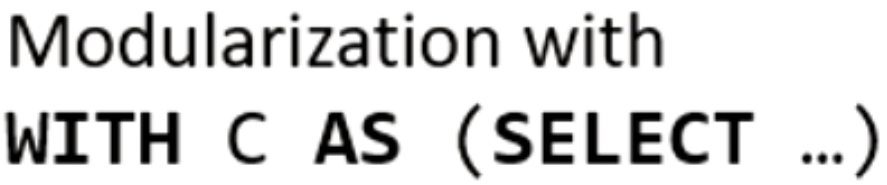
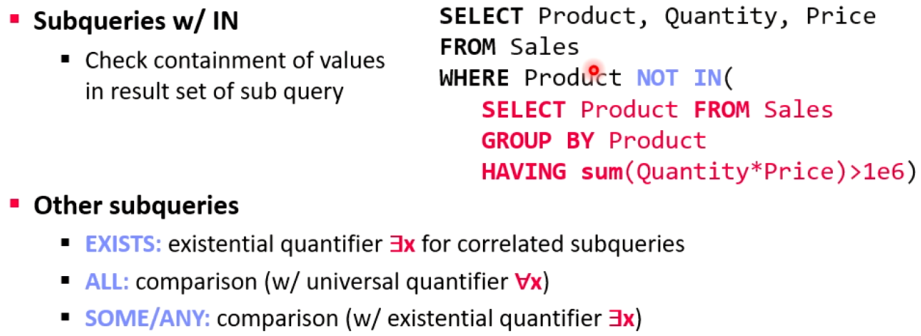
- use case
  - query result instead of [[SQL]] table
  - aggregation value instead of scalar value
- subqueries can be stored as variable
  - WITH VariableName AS (SELECT …)

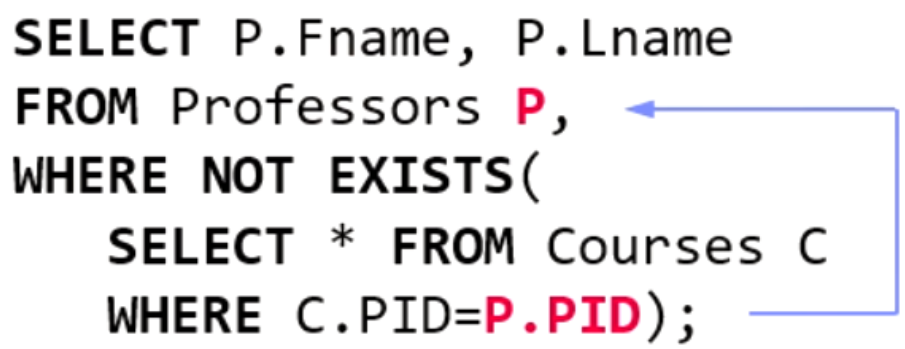


- common use cases:



Correlated and Uncorrelated Subqueries

- correlated if queries depend on each other
  - subquery exectuted for each tuple of outer query
    - nested for loop
  - inefficient



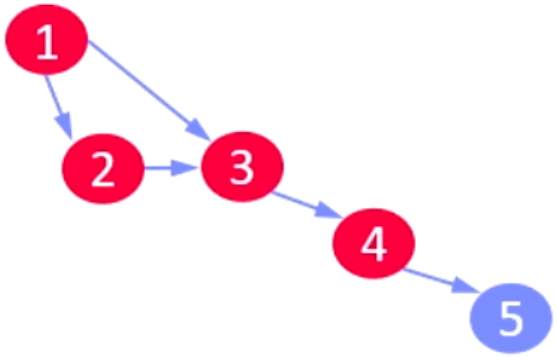
- uncorrelated if subquery executed once

```
SELECT P.Fname, P.Lname
FROM Professors P,
WHERE P.PID NOT IN(
    SELECT PID FROM Courses);
```

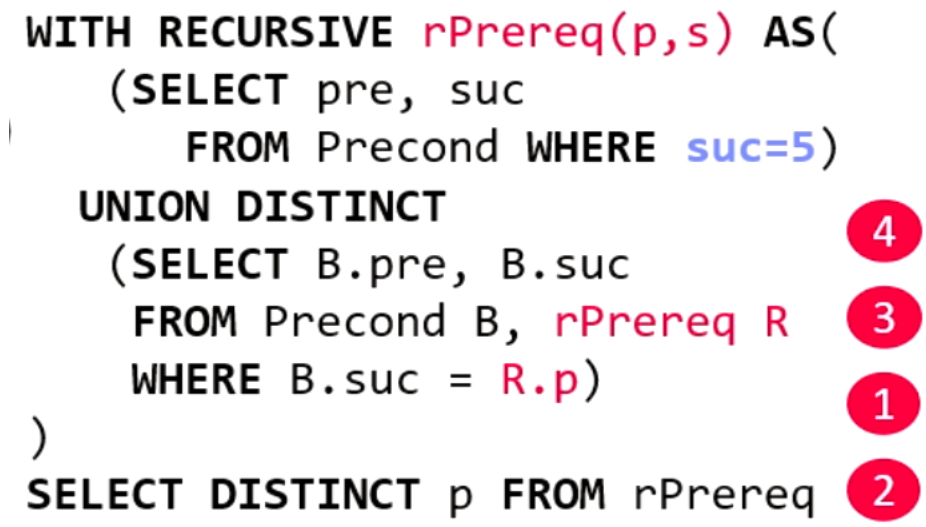
- correlated queries may be unnested (de-correlation)
  - can also be improved by “only” executing subquery for each distinct value

Recursive Queries

- terminates when recursive query returns empty table
  - **Approach**
    - `WITH RECURSIVE <name> (<arguments>)`
    - Compose recursive table from non-recursive term, union all/distinct, and recursive term
    - Terminates when recursive term yields empty result
-
  - **Example**
    - Courses(CID, Name),
      Precond(pre REF CID, suc REF CID)
    - Dependency graph (pre→suc)



-



[[Correlation]]