

FINAL QUERY for 1 by 1 approach:

```
MATCH (lib:Artifact {id: 'DESIREDID'})<-[:dependency]-(proj:Release)
  WITH lib,
  count(proj) AS dependent_projects,
  avg(size((proj)-[:dependency*]->())) AS avg_dependency_depth,
  max(size((proj)-[:dependency*]->())) AS max_dependency_chain,
  max(proj.timestamp) AS most_recent_dependency
MATCH (lib)-[:relationship_AR]->(release:Release)
WITH lib, dependent_projects, avg_dependency_depth, max_dependency_chain,
most_recent_dependency, release
ORDER BY release.timestamp DESC
LIMIT 1
RETURN
  dependent_projects,
  size((release)-[:dependency]->()) AS total_dependencies,
  avg_dependency_depth,
  max_dependency_chain,
  most_recent_dependency
```

Get top 50 projects:

```
MATCH (artifact:Artifact)-[:relationship_AR
]->(r:Release)
WITH artifact,
count(distinct r) AS releaseCount,
sum(size((r)-[:dependency]->())) AS
total_deps
WHERE releaseCount > 10
AND total_deps > 0
ORDER BY total_deps DESC
LIMIT 50
```

Get random 50:

```
MATCH (artifact:Artifact)-[:relationship_AR
]->(r:Release)
WITH artifact,
count(distinct r) AS release_count,
size((r)-->()) AS dependency_count
WHERE release_count > 10 AND
dependency_count > 0
WITH artifact, release_count, dependency_count
ORDER BY rand()
LIMIT 50
```

