

01

```
MATCH (e:Employee)
WHERE (e.salary > 2200) AND (e.email ENDS WITH "@co.org")
RETURN e.number, e.firstName + " " + e.lastName AS name
```

```
... WHERE (e.salary > 2200) AND (e.email =~ ".*@co\\.org") ...
```

```
... WHERE (e.salary > 2200) AND (e.email =~ "(?i).*@co\\.org") ...
```

```
MATCH (e)
WHERE (e:Employee) AND ... ..
```

```
... WHERE ANY(t IN LABELS(e) WHERE (t = "Employee")) AND ... ..
```

02

```
MATCH (e:Employee)-[w:WorksIn]->(d:Department)
WHERE (w.from >= "2013-01-01")
RETURN e.number, d.id, d.name
```

```
MATCH (e:Employee)-[w:WorksIn]-(d:Department) ...
```

```
MATCH (e)-[w]->(d:Department)
WHERE ANY(t IN LABELS(e) WHERE (t = "Employee")) AND (TYPE(w) = "WorksIn") AND ... ..
```

```
... WHERE EXISTS(w.from) AND (w.from >= "2013-01-01") ...
```

```
... WHERE (w.from IS NOT NULL) AND (w.from >= "2013-01-01") ...
```

Incorrect:

```
... WHERE (w.from != NULL) AND (w.from >= "2013-01-01") ...
```

03

```
MATCH (d:Department)
WHERE (:Employee)-[:WorksIn]->(d)
RETURN d.id, d.name
ORDER BY d.name
SKIP 2
LIMIT 3
```

```
MATCH (:Employee)-[:WorksIn]->(d:Department)
RETURN DISTINCT d.id, d.name ...
```

```
... WHERE EXISTS((:Employee)-[:WorksIn]->(d)) ...
```

```
... WHERE SIZE((:Employee)-[:WorksIn]->(d)) >= 1 ...
```

04

```
MATCH (e:Employee)-[:WorksIn]->()-[:HasSubdepartment*]-(d:Department)
WHERE (d.id = "D1")
RETURN DISTINCT e.lastName
```

```
MATCH (e:Employee)-[:WorksIn]->()-[:HasSubdepartment*]-(d:Department{id:"D1"})
RETURN DISTINCT e.lastName
```

```
MATCH (e:Employee)-[:WorksIn]->()-[:HasSubdepartment*0..]-(d:Department) ...
```

```
START d = node:Department(id = "D1")
```

```
MATCH (e:Employee)-[:WorksIn]->()-[:HasSubdepartment*]-(d:Department) ...
```

05

```
MATCH (d:Department)<-[WorksIn]-(e:Employee)
WHERE EXISTS(d.name)
WITH d.id AS id, AVG(e.salary) AS avgSalary, COUNT(*) AS empCount
WHERE empCount >= 2
RETURN id, avgSalary
ORDER BY avgSalary DESCENDING, id
```

```
MATCH (e:Employee)-[:WorksIn]->(d:Department) ...
```

```
... WHERE ANY(k IN KEYS(d) WHERE (k = "name")) ...
```

```
... WHERE d.name IS NOT NULL ...
```

```
... ORDER BY avgSalary DESCENDING, d.id ASCENDING
```

```
... ORDER BY avgSalary DESC, d.id ASC
```

06

```
MATCH (d:Department)-[:HasSubdepartment*]->(s:Department)
WITH d.id AS id, s.id AS sid
ORDER BY id, sid
WITH id, COLLECT(sid) [1..] AS subdepartments
OPTIONAL MATCH (d)-[:HasManager]->(m)
RETURN id, subdepartments, m.firstName + " " + m.lastName AS manager

... WITH id, TAIL(COLLECT(sid)) AS subdepartments ...
```