DB Assignment 3:

1 & 2:

Neo4J:

```
CREATE INDEX ON :Person(name);

CREATE CONSTRAINT ON (p:Person) ASSERT p.id IS UNIQUE;

USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM "file:///social_network_nodes.csv" AS row
MERGE (:Person {id: toInt(row.node_id), name: row.name, job: row.job, birthday:
row.birthday});

USING PERIODIC COMMIT 5000
LOAD CSV WITH HEADERS FROM "file:///social_network_edges.csv" AS row
MATCH (f:Person {id: toInt(row.source_node_id)}), (t:Person {id: toInt(row.target_node_id)})
CREATE (f)-[:ENDORSES]->(t);
```

MySQL:

```
CREATE TABLE t_user (
   id int NOT NULL,
         name varchar(255),
   job varchar(255),
   birthday varchar(255),
   PRIMARY KEY(id));

CREATE TABLE t_endorses (
source_node_id int NOT NULL,
target_node_id int NOT NULL,
PRIMARY KEY (source_node_id, target_node_id),
FOREIGN KEY(source_node_id) REFERENCES t_user(id),
FOREIGN KEY(target_node_id) REFERENCES t_user(id));

LOAD DATA LOCAL INFILE
'E:/Kasper/School/db_course_nosql/social_network/social_network_nodes.csv'
INTO TABLE sys.t_user
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(id,name,job,birthday)
;

LOAD DATA LOCAL INFILE
'E:/Kasper/School/db_course_nosql/social_network/social_network_edges.csv'
```

```
INTO TABLE sys.t_endorses
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(source_node_id, target_node_id)
;
```

3.
Neo4J:
Depth 1
```
MATCH(:Person{id:0})-[:ENDORSES]->(p:Person) RETURN distinct(p);
```
Depth 2
```
MATCH(:Person{id:0})-[:ENDORSES]->(:Person)-[:ENDORSES]->(p:Person) RETURN
distinct(p);
```
Depth 3
```
MATCH(:Person{id:0})-[:ENDORSES]->(:Person)-[:ENDORSES]->(:Person)-[:ENDORSES]-
>(p:Person) RETURN distinct(p);
```
Depth 4
```
MATCH(:Person{id:0})-[:ENDORSES]->(:Person)-[:ENDORSES]->(:Person)-[:ENDORSES]-
>(:Person)-[:ENDORSES]->(p:Person) RETURN distinct(p);
```
Depth 5
```
MATCH(:Person{id:0})-[:ENDORSES]->(:Person)-[:ENDORSES]->(:Person)-[:ENDORSES]-
>(:Person)-[:ENDORSES]->(:Person)-[:ENDORSES]->(p:Person) RETURN distinct(p);
```

MySQL:
Depth 1
```
SELECT target_node_id
FROM t_user JOIN t_endorses ON t_user.id = source_node_id
WHERE source_node_id = 0;
```

Depth 2
```
SELECT target_node_id
FROM t_user JOIN t_endorses ON t_user.id = source_node_id
WHERE source_node_id IN (SELECT target_node_id FROM t_user JOIN t_endorses ON t_
user.id = source_node_id WHERE source_node_id = 0);
```

Depth 3
```
SELECT target_node_id
FROM t_user JOIN t_endorses ON t_user.id = source_node_id
WHERE source_node_id IN (SELECT target_node_id FROM t_user JOIN t_endorses ON
t_user.id = source_node_id WHERE source_node_id IN (SELECT target_node_id FROM
t_user JOIN t_endorses ON t_user.id = source_node_id WHERE source_node_id = 0));
```

Depth 4
```
SELECT target_node_id
FROM t_user JOIN t_endorses ON t_user.id = source_node_id
```

```
WHERE source_node_id IN (SELECT target_node_id FROM t_user JOIN t_endorses ON
t_user.id = source_node_id WHERE source_node_id IN (SELECT target_node_id FROM
t_user JOIN t_endorses ON t_user.id = source_node_id WHERE source_node_id IN
(SELECT target_node_id FROM t_user JOIN t_endorses ON t_user.id = source_node_id
WHERE source_node_id = 0)));
```

Depth 5

```
SELECT target_node_id
FROM t_user JOIN t_endorses ON t_user.id = source_node_id
WHERE source_node_id IN (SELECT target_node_id FROM t_user JOIN t_endorses ON
t_user.id = source_node_id WHERE source_node_id IN (SELECT target_node_id FROM
t_user JOIN t_endorses ON t_user.id = source_node_id WHERE source_node_id IN
(SELECT target_node_id FROM t_user JOIN t_endorses ON t_user.id = source_node_id
WHERE source_node_id IN (SELECT target_node_id FROM t_user JOIN t_endorses ON
t_user.id = source_node_id WHERE source_node_id = 0))));
```