Name: Chuqi Wang

Student ID: 79167724

Q1:

A:

i:

LOAD CSV WITH HEADERS FROM 'file:///Users/chuqiwang/Desktop/UCI/CS224P/assignments/HW4/zot-music-dataset-assignment4/Users.csv' AS row

MERGE (u:User {user\_id: row.user\_id})

ON CREATE SET

u.email = row.email,

u.joined\_date = date(row.joined\_date),

u.nickname = row.nickname,

u.street = row.street,

u.city = row.city,

u.state = row.state,

u.zip = toInteger(row.zip),

u.genres = split(row.genres, ",");

LOAD CSV WITH HEADERS FROM 'file:///Users/chuqiwang/Desktop/UCI/CS224P/assignments/HW4/zot-music-dataset-assignment4/Listeners.csv' AS row

MERGE (u:User {user\_id: row.user\_id})

SET u:Listener,

u.subscription = row.subscription,

u.first\_name = row.first\_name,

u.last\_name = row.last\_name;

LOAD CSV WITH HEADERS FROM 'file:///Users/chuqiwang/Desktop/UCI/CS224P/assignments/HW4/zot-music-dataset-assignment4/Artists.csv' AS row

MERGE (u:User {user\_id: row.user\_id})

SET u:Artist,

u.bio = row.bio,

u.stagename = row.stagename;

LOAD CSV WITH HEADERS FROM 'file:///Users/chuqiwang/Desktop/UCI/CS224P/assignments/HW4/zot-music-dataset-assignment4/Records.csv' AS row

MERGE (r:Record {record\_id: row.record\_id})

ON CREATE SET

r.title = row.title,

r.genre = row.genre;

LOAD CSV WITH HEADERS FROM 'file:///Users/chuqiwang/Desktop/UCI/CS224P/assignments/HW4/zot-music-dataset-assignment4/Albums.csv' AS row

MERGE (r:Record {record\_id: row.record\_id})

SET r:Album,

r.description = row.description;

LOAD CSV WITH HEADERS FROM 'file:///Users/chuqiwang/Desktop/UCI/CS224P/assignments/HW4/zot-music-dataset-assignment4/Singles.csv' AS row

MATCH (r:Record {record\_id: row.record\_id})

SET r:Single,

r.video\_url = row.video\_url;

LOAD CSV WITH HEADERS FROM 'file:///Users/chuqiwang/Desktop/UCI/CS224P/assignments/HW4/zot-music-dataset-assignment4/Reviews.csv' AS row

MERGE (r:Review {review\_id: row.review\_id})

ON CREATE SET

r.rating = toInteger(row.rating),

r.body = row.body;

ii:

CREATE INDEX user\_id\_index FOR (u:User) ON (u.user\_id);

CREATE INDEX record\_id\_index FOR (r:Record) ON (r.record\_id);

iii:

LOAD CSV WITH HEADERS FROM 'file:///Users/chuqiwang/Desktop/UCI/CS224P/assignments/HW4/zot-music-dataset-assignment4/Artists\_Releases\_Record.csv' AS row

MATCH (a:Artist {user\_id: row.artist\_user\_id})

MATCH (r:Record {record\_id: row.record\_id})

MERGE (a)-[:RELEASE {release\_date: date(row.release\_date)}]->(r);

LOAD CSV WITH HEADERS FROM 'file:///Users/chuqiwang/Desktop/UCI/CS224P/assignments/HW4/zot-music-dataset-assignment4/Users\_Post\_Reviews.csv' AS row

MATCH (u:User {user\_id: row.user\_id})

MATCH (r:Review {review\_id: row.review\_id})

MERGE (u)-[:POSTED\_AT {posted\_at: datetime(replace(row.posted\_at, " ", "T"))}]->(r);

LOAD CSV WITH HEADERS FROM 'file:///Users/chuqiwang/Desktop/UCI/CS224P/assignments/HW4/zot-music-dataset-assignment4/Reviews\_About\_Record.csv' AS row

MATCH (r:Review {review\_id: row.review\_id})

MATCH (rec:Record {record\_id: row.record\_id})

MERGE (r)-[:ABOUT]->(rec);

LOAD CSV WITH HEADERS FROM 'file:///Users/chuqiwang/Desktop/UCI/CS224P/assignments/HW4/zot-music-dataset-assignment4/Upvotes.csv' AS row

MATCH (u:User {user\_id: row.user\_id})

MATCH (r:Review {review\_id: row.review\_id})

MERGE (u)-[:UPVOTED]->(r);

B:

i:

A screenshot of a computer

Description automatically generated

ii:

A screenshot of a computer

Description automatically generated

Q2:

Query:

MATCH (r:Record)

WITH r LIMIT 1000

UNWIND keys(r) AS recordKeys

RETURN DISTINCT recordKeys;

Results (screenshot below):

A screenshot of a computer

Description automatically generated

Q3 A:

Query:

MATCH (u:User)

RETURN u

ORDER BY u.joined\_date DESC

LIMIT 5;

Results (screenshot below):

A screenshot of a computer code

Description automatically generated

Q3 B:

Query:

MATCH (u:User {user\_id: "user\_WKWGxlZa"})-[rel:RELEASE]->(r:Record)

RETURN r, rel.release\_date

ORDER BY r.record\_id ASC;

Results (screenshot below):

A screenshot of a computer

Description automatically generated

Q3 C:

Query:

MATCH (l:Listener)

WHERE NOT (l)-[:POSTED\_AT]->(:Review)

RETURN COUNT(l) AS count\_of\_listeners\_without\_reviews;

Results (screenshot below):

A screenshot of a computer

Description automatically generated

Q3 D:

Query:

MATCH (l:Listener)-[:POSTED\_AT]->(r:Review)-[:ABOUT]->(rec:Record {title: "Audience star apply"})

WHERE r.rating = 5

RETURN DISTINCT l.first\_name, l.last\_name

ORDER BY l.first\_name ASC;

Results (screenshot below):

A screenshot of a computer

Description automatically generated

Q3 E:

Query:

MATCH (rec:Record)<-[:ABOUT]-(r:Review)

WITH rec, COUNT(r) AS review\_count

WHERE review\_count > 25

RETURN rec.record\_id, rec.title, review\_count

ORDER BY review\_count DESC;

Results (screenshot below):

A screenshot of a computer code

Description automatically generated

Q3 F:

Query:

MATCH (l:Listener)-[:POSTED\_AT]->(r:Review)

WITH l, COUNT(r) AS review\_count

WHERE review\_count > 60

MATCH (l)-[:UPVOTED]->(u:Review)

WITH l, review\_count, COUNT(u) AS upvote\_count

WHERE upvote\_count > 500

RETURN l.user\_id

ORDER BY l.user\_id ASC

LIMIT 5;

Results (screenshot below):

A screenshot of a computer

Description automatically generated

Q3 G:

Query:

MATCH (l1:Listener)-[:POSTED\_AT]->(r1:Review)-[:ABOUT]->(rec:Record)<-[:ABOUT]-(r2:Review)<-[:POSTED\_AT]-(l2:Listener)

WHERE l1 <> l2 AND r1.rating = r2.rating AND id(l1) < id(l2)

RETURN DISTINCT l1.last\_name AS listener1\_last\_name, l2.last\_name AS listener2\_last\_name, rec.record\_id

ORDER BY rec.record\_id ASC

LIMIT 10;

Results (screenshot below):

A screenshot of a computer

Description automatically generated

Q3 H:

Query:

MATCH (a:User)-[:POSTED\_AT]->(r1:Review)-[:ABOUT]->(rec1:Record)<-[:RELEASE]-(b:User),

(b)-[:POSTED\_AT]->(r2:Review)-[:ABOUT]->(rec2:Record)<-[:RELEASE]-(a)

WHERE id(a) < id(b)

WITH DISTINCT a.nickname AS user\_a\_nickname, b.nickname AS user\_b\_nickname

RETURN user\_a\_nickname, user\_b\_nickname

ORDER BY user\_a\_nickname ASC

LIMIT 10;

Results (screenshot below):

A screenshot of a computer

Description automatically generated

Q3 I:

i

Query:

MATCH (start:Artist {user\_id: "user\_0ZIeALBX"}), (target:Artist)

WHERE start <> target

WITH shortestPath((start)-[\*]-(target)) AS path

WHERE path IS NOT NULL

RETURN MIN(length(path)) AS min\_shortest\_path\_length;

Results (screenshot below):

A screenshot of a computer

Description automatically generated

ii

Query:

MATCH (start:Artist {user\_id: "user\_0ZIeALBX"}), (target:Artist)

WHERE start <> target

WITH shortestPath((start)-[\*]-(target)) AS path, target

WHERE length(path) = 6

RETURN target.user\_id AS artist\_user\_id

ORDER BY artist\_user\_id ASC

LIMIT 5;

Results (screenshot below):

