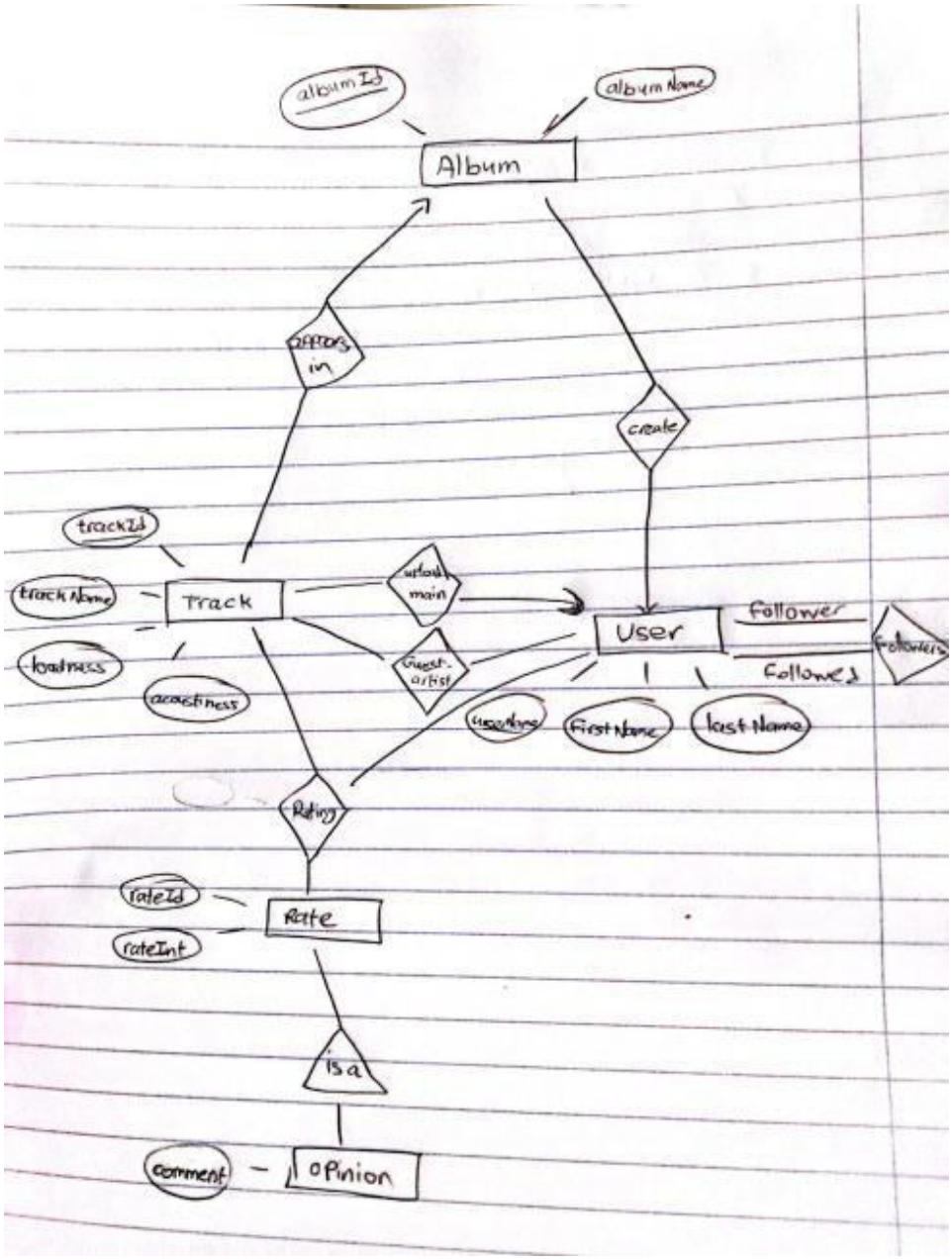
אביגיל דאנש

אופק ימיני

Question 1 a)



b) Tables

User(<u>userName</u>, firstName, lastName)

Album(<u>albumId</u>, albumName, creatorUserName)

Followers(followerName, followedName)

Track(trackId, trackName, acousticness, loudness, uploaderUserName, albumId)

Rate(rateId, rateInt)

Opinion(opinionId, comment)

GuestArtist(trackId, guestUserName)

Rating(rankerUserName, trackId, rateId)

Foreign keys:

Track(uploaderUserName) -> User(userName)

Track(albumId) -> Album(albumId)

Album(creatorUserName) -> User(userName)

Followers(followerName) -> User(userName)

Followers(followedName) -> User(userName)

Opinion(opinionId) –> Rate(rateId)

GuestArtist(trackId) -> Track(trackId)

GuestArtist(guestUserName) -> User(userName)

Rating(rankerUserName) -> User(userName)

Rating(trackId) -> Track(trackId)

Rating(rateId) -> Rate(rateId)

Question 2:

A -> B, B -> C, C -> D, D -> B

Not trivial: A -> B,C,D B-> C,D C -> D,B

Section Aleph: A is a minimal key because he derives all the others. We can see that B->C is a violation of the BCNF because B is not a key so lets divide it to 2 relations:

R1 = (B,C) R2=(B,D,A)

R1 has no violations so we have done with it, for R2 we can see that A is its minimal key and D->B is the violation of BCNF so lets divide it into 2 relations:
R12=(D,B) R22(D,A) and now they both have no violations so we done .

Section beth: There is a decomposition dependency preserving: lets find for every table its transitive closure of its sub set:
R1(A,B) => {A}+ = {A,B}, {B}+ = {B}
R2(B,C) => {B}+ = {B,C}, {C}+ = {C,B}
R3(B,D) => {B}+ = {B,D}, {D}+ = {D,B}
We can see that from {A}+ we get that A->B
From {B}+ we get that B->C and also B->D and C->B
From {C}+ we get that C->D because we had B->D and C->B earlier so from transitive we get this one also.
From {D}+ we get that D->B
So overall we get all we need and there is a decomposition dependency preserving.

Question 3:

Section Aleph:

DepartmentCourses(department_id, course_id, department_head, course_name)
department_id -> department_head
course_id -> course_name
department_id, course_id -> department_head, course_name
the minimal key is {department_id,course_id}. We can see that "department_id->department_head" is a violation to BCNF because department_id is not a minimal key by itself. (we can also say that course_id -> course_name is a violation from the same reason).

Section Beth:

1. The BCNF is correcting every table a single logical connection, this is bringing to a modular programming code that is more easy to work with. for example, if department_head is changing, we will need to update it in only one place in the table. However, without BCNF we would have update it in many places in the table.
2. When we are using BCNF we divide our relations into small parts and it help us to avoid duplicates as possible as we can. For example, for every course there will be the same department_head, so without the BCNF there will be many duplicates and as a result we will se the department_head near every course although we already now it and we don't need to write it on every course. With the BCNF we can have department_head once in the table and all the courses will be under it.

Question 4:

1)

A, D -> H B, C -> D C -> E, B H -> F, G

->

1 2 3 4 5 6

A, D -> H B, C -> D C -> E C-> B H -> F H-> G

->

1+5+6 1+2 2+4

A,D -> F,G A,B,C -> H C->D

->

1+2+4 1+2+5+6

A,C ->H A,B,C -> F,G

->

1+2+4+5+6= *

A,C -> F,G

2) {A,C} is minimal key because 4+*+3+2 A,C -> A,B,C,D,E,F,G,H

3)

A, D -> H B, C -> D C -> E, B H -> F, G

We are many option of violation, we choose this one:

->

C-> E,B violate, so R1=(C,E,B) R2=(C,A,D,F,G,H)

In R1 C is minimal key, so everything is fine and there is no violations.

In R2=(C,A,D,F,G,H) we get to A, D -> H H -> F, G

-> {A,C} the minimal key (because A,C -> F,G).

H -> F, G violate because H is not minimal key, so the relations are: R21=(H,F,G) R22=(H,A,C,D).

-> in R21 H is minimal key, so everything is fine and there is no violations.

In R22=(H,A,C,D) we have the rule A, D -> H and there is violation

Because A,D is not minimal key (cant reach to C), so:

R221=(A,D,H) R222=(A,D,C)

->

in R221=(A,D,H) the rule are A, D -> H, and there is no violation because A,D is minimal key.

In R222=(A,D,C) we have the rule C->D so lets part it to R2221=(C,D) R2222=(C,A) and they have length of 2 so they are in the BCNF.