GROUP 96 Ludwing Berglind Simon Severinsson

1. No feedbeack was given, hence no changes.

2.

users(userID: integer, name: string)

friendships(userID1: integer, userID2: integer)

posts(postID: integer, userID: integer, title: string, date: date, place: string)

tags(postID: integer, tag: string)

image_posts(postID: integer, imgURL: string, filter: string)

text_posts(postID: integer, textContent: string)

video_posts(postID: integer, vidURL: string, codec: string)

likes(postID: integer, userID: integer, timestamp: timestamp)

events(eventID: integer, userID: integer, title: string, place; string, startDate: timestamp, endDate: timestamp)

attendees(eventID: integer, userID: integer)

subscription(subscriptionID: integer, userID: integer, dop: date, paymeth: string)

3.

a)

$$\pi_{UserID}\left(\text{Users}\right) - \pi_{UserID}\left(\sigma_{(dop+30)\geq today}\left(\text{Subscription}\right)\right)$$

We select all entries from subscription where dop (date of payment) + 30 is greater than or equal to today, which means we get all current subscriptions, we project UserID onto that selection, then take the difference of the projection of UserID onto Users and the previously mentioned projection, so that all UserID that have a current subscription are removed.

b)

$$\pi_{UserID}\left(\text{Users}\right) - \delta(\pi_{UserID}\left(\sigma_{StartDate\leq today}\left(\text{Events} \bowtie_{Events.EventID=Attendees.EventID} \text{Attendees}\right)\right)$$

On the right hand side of the difference, the deepest operation is a join of Event and Attendees on EventID, this returns all events and all attendees that attended each event, we then select all entries where StartDate $\leq$ today to eliminate events which have not yet taken place. On this we then project UserID and remove any duplicates, which leaves us with the userID of all users that have attended an event. On the left hand side of the difference we project UserID on all users, which means that the difference removes all UserIDs that have attended an event from all UserIDs, resulting in only the UserIDs which have not yet attended any event remaining.

c)

$$\pi_{name}\left(\text{Users} \bowtie \sigma_{noLikes>5}\left(\Upsilon_{userID,\ \text{COUNT}(postID)\longrightarrow noLikes}\left(\text{Likes}\right)\right)\right)$$

The deepest operation here is grouping of Likes by userID and the Count of postID, which yields the number of posts each user has liked. From this we select all entries where the number of likes is greater than 5, and join that with users (the only common attribute between Likes and Users is UserID, so this join connects every User to the amount of likes they have made), we then project Name on to that join.

d)

$$\delta\left(\pi_{codec}\left(\sigma_{eventID=1368}\left(\text{Events}\right) \bowtie_{Events.place=VP.place} \rho_{VP}\left(\text{Posts} \bowtie \text{VideoPosts}\right)\right)\right)$$

The deepest operation here is a join of Posts and VideoPosts, this connects the Codec of each VideoPost to their respective Post, and call the result of the join VP. We then join the selection of Events where eventID = 1368 with VP with the condition that they have the same location. Which basically results in all posts that were taken in the same place as the event with eventID 1368. We project codec onto this and then remove any duplicates.

e)

$$\pi_{postID}\left(\rho_{CPosts}\left(\sigma_{tag='Crypto'}\left(\text{Tags}\right)\right) \bowtie_{CPosts.postID=SPosts.PostID} \rho_{SPosts}\left(\sigma_{tag='Studying'}\left(\text{Tags}\right)\right)\right)$$
$$-\pi_{postID}(\sigma_{date<2024/01/01}\left(\text{Posts}\right)$$

Here we do a selection on Tags where tag is Crypto, called CPosts and one where tag is Studying called SPosts. We then join CPosts and SPosts on the condition that they share the same postID, which results in all posts that have the tag Crypto and the tag Studying. Onto that join we project postID, we then take the difference of said projection and the projection of postID on the selection of Posts where date is earlier than 2024, which removes all posts from the right hand side that were made before 2024, leaving all postIDs from Posts posted after the year 2023.