**Q1. user\_activity**

create table user\_activity (

user\_id int not null,

activity\_date date not null

);

insert into user\_activity values

(1,'2023-09-10'),

(2,'2023-09-10'),

(1,'2023-09-11'),

(3,'2023-09-11'),

(2,'2023-09-12'),

(4,'2023-09-12'),

(3,'2023-09-12'),

(4,'2023-09-13');

SELECT

u1.activity\_date,

(SELECT COUNT(DISTINCT u2.user\_id)

FROM user\_activity u2

WHERE u2.activity\_date = u1.activity\_date

AND u2.activity\_date = (

SELECT MIN(u3.activity\_date)

FROM user\_activity u3

WHERE u3.user\_id = u2.user\_id

)

) AS new\_users,

(SELECT COUNT(DISTINCT u4.user\_id)

FROM user\_activity u4

WHERE DATEDIFF(u4.activity\_date, u1.activity\_date) = 1

AND u4.user\_id IN (

SELECT u5.user\_id

FROM user\_activity u5

WHERE u5.activity\_date = u1.activity\_date

AND u5.activity\_date = (

SELECT MIN(u6.activity\_date)

FROM user\_activity u6

WHERE u6.user\_id = u5.user\_id

)

)

) AS returned\_users,

IFNULL(

ROUND(

(SELECT COUNT(DISTINCT u4.user\_id)

FROM user\_activity u4

WHERE DATEDIFF(u4.activity\_date, u1.activity\_date) = 1

AND u4.user\_id IN (

SELECT u5.user\_id

FROM user\_activity u5

WHERE u5.activity\_date = u1.activity\_date

AND u5.activity\_date = (

SELECT MIN(u6.activity\_date)

FROM user\_activity u6

WHERE u6.user\_id = u5.user\_id

)

)

) /

(SELECT COUNT(DISTINCT u2.user\_id)

FROM user\_activity u2

WHERE u2.activity\_date = u1.activity\_date

AND u2.activity\_date = (

SELECT MIN(u3.activity\_date)

FROM user\_activity u3

WHERE u3.user\_id = u2.user\_id

)

), 2), 0) AS day\_1\_retention\_rate

FROM

(SELECT DISTINCT activity\_date FROM user\_activity) u1;

**Q2. user\_events**

create table user\_events (

user\_id int not null,

event\_type char(6) not null,

event\_time timestamp not null);

insert into user\_events values

(1,'click','2023-09-10 10:00:00'),

(1,'scroll','2023-09-10 10:10:00'),

(1,'click','2023-09-10 10:50:00'),

(1,'scroll','2023-09-10 11:40:00'),

(2,'click','2023-09-10 09:00:00'),

(2,'scroll','2023-09-10 09:20:00'),

(2,'click','2023-09-10 10:30:00');

SET @session\_id = 0;

SET @prev\_user = NULL;

SET @prev\_time = NULL; -- Initialize @prev\_time

SELECT

user\_id,

session\_id,

MIN(event\_time) AS session\_start\_time,

MAX(event\_time) AS session\_end\_time,

TIMEDIFF(MAX(event\_time), MIN(event\_time)) AS session\_duration,

COUNT(\*) AS event\_count

FROM (

SELECT

user\_id,

event\_time,

@session\_id := IF(

user\_id = @prev\_user AND TIMESTAMPDIFF(MINUTE, @prev\_time, event\_time) <= 30,

@session\_id,

IF(user\_id = @prev\_user, @session\_id + 1, 1) -- Reset session for new users

) AS session\_id,

@prev\_user := user\_id,

@prev\_time := event\_time

FROM user\_events

) AS session\_data

GROUP BY user\_id, session\_id;

**Q3. You are tasked with increasing the Day-on-Day (DoD) retention of users on the Seekho app. Currently, many users sign up and engage with content initially, but their activity drops off after the first day. What changes would you suggest to improve Day-on-Day retention? Consider both product features and data-driven strategies in your response.**

* Incorporate gamification concept by providing points to encourage users to engage with the application consistently. These points can be used to access premium videos. Also bonus points on maintaining weekly/monthly streaks.
* Integrate a feature that allows users to connect with friends, view their progress, and compare points, creating a competitive and interactive learning experience.
* Utilize **collaborative-based filtering** to suggest videos tailored to users' preferences and learning history by analysing similar user’s interactions and interests.