DBMS Project

Focus Mode App

DBMS for a focus mode app

**Naman Kashyap**

PES1UG20CS260

V semester ‘E’ section

Roll no. 27

**Format:**

**Mini Project:**

1. A short description about the project and scope

2. ER Diagram

3. Relational Scheme

4. DDL statements to build the database

5. Different methods used to populate data. - Show statements used under different methods

6. JOIN queries

7. Aggregate Functions

8. SET Operators

9. Functions or Procedure

10. Triggers or cursors

11. Higher Level Programming - A simple frontend that talks to the backend database is required to be developed.

1. **A short description:**

**Focus mode DBMS**

A focus mode database, that keeps track of focus sessions, apps used, breaks taken, and app timers. This project uses MySQL and Python to simulate the features and functionalities of a focus mode assist app. The aim of this project is to enhance our understanding of MySQL and to create a minimalistic UI for the app using tkinter in Python.

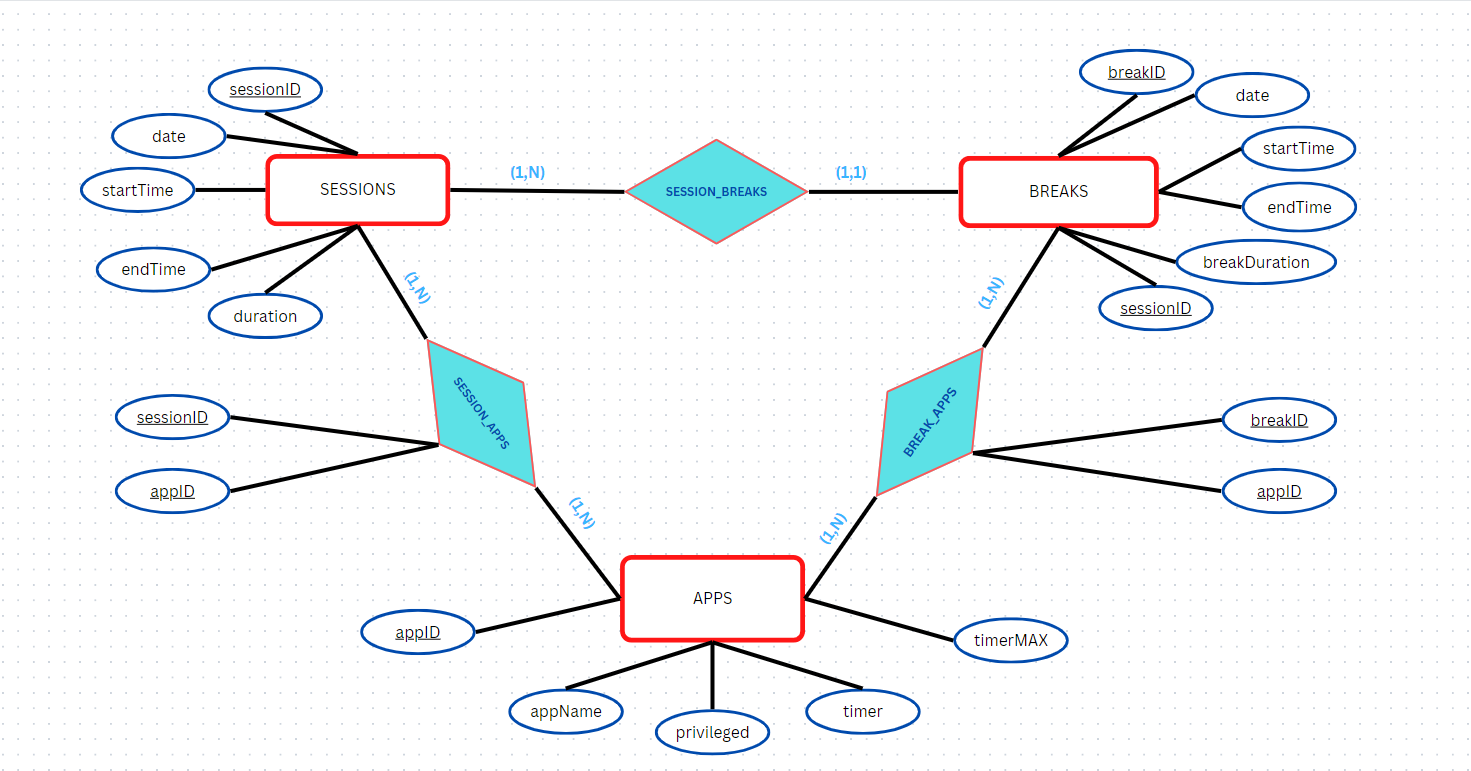
**Tables:**

* Sessions
* Breaks
* Apps

**Relations:**

* Session\_breaks
* Session\_apps
* Break\_apps

1. **ER diagram:**



1. **Relational schema:**

**1. Sessions:**

*A history of all sessions*

* sessionID (pk)
* date
* startTime
* endTime
* duration

**2. Breaks:**

*A history of all breaks*

* breakID (pk)
* date
* startTime
* endTime
* breakDuration
* sessionID (fk)

**3. Apps:**

*A list of all apps on device*

* appID (pk)
* appName
* privileged (yes/no)
* timer (if not privileged)
* timerMAX (if not privileged)

***Relationship Descriptions:***

**4. Session\_apps: (M:N)**

*Apps allowed during focus mode session. i.e the app(s) you had to focus on*

* sessionID
* appID

**5. Break\_apps: (M:N)**

*Apps used during breaks*

* breakID
* appID

**Relational Schema Diagram:**

SESSIONS:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sessionID | sessionDate | startTime | endTime | duration |

BREAKS:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| breakID | breakDate | startTime | endTime | breakDuration | sessionID |

APPS:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| appID | appName | privileged | timer | timerMAX |

SESSION\_APPS:

|  |  |
| --- | --- |
| sessionID | appID |

BREAK\_APPS:

|  |  |
| --- | --- |
| breakID | appID |

1. **DDL statements:**

*create\_tables.sql* **file:**

drop database focusModeDBMS;

create database focusModeDBMS;

use focusModeDBMS;

create table SESSIONS (

    sessionID varchar(10) not null,

    sessionDate date not null,

    startTime time not null,

    endTime time not null,

    duration varchar(20),

    primary key (sessionID)

);

create table BREAKS (

    breakID varchar(10) not null,

    breakDate date not null,

    startTime time not null,

    endTime time not null,

    breakDuration varchar(20) not null,

    sessionID varchar(10) not null,

    primary key (breakID),

    foreign key (sessionID) references SESSIONS(sessionID)

);

create table APPS (

    appID varchar(10) not null,

    appName varchar(50) not null,

    privileged bit not null,

    timer varchar(20),

    timerMAX varchar(20),

    primary key (appID)

);

create table SESSION\_APPS (

    sessionID varchar(10) not null,

    appID varchar(10) not null,

    primary key (sessionID, appID),

    foreign key (sessionID) references SESSIONS (sessionID),

    foreign key (appID) references APPS (appID)

);

create table BREAK\_APPS (

    breakID varchar(10) not null,

    appID varchar(10) not null,

    primary key (breakID, appID),

    foreign key (breakID) references BREAKS (breakID),

    foreign key (appID) references APPS (appID)

);

delimiter &&

create procedure getNEntries (in lim int)

begin

    select \* from APPS limit lim;

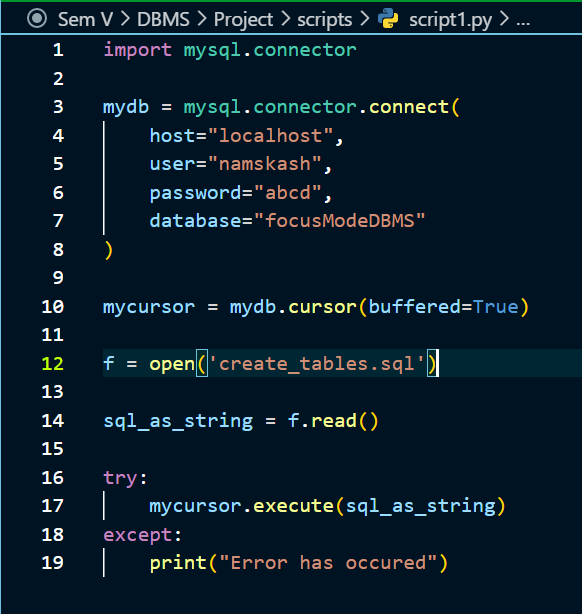
end &&

delimiter;

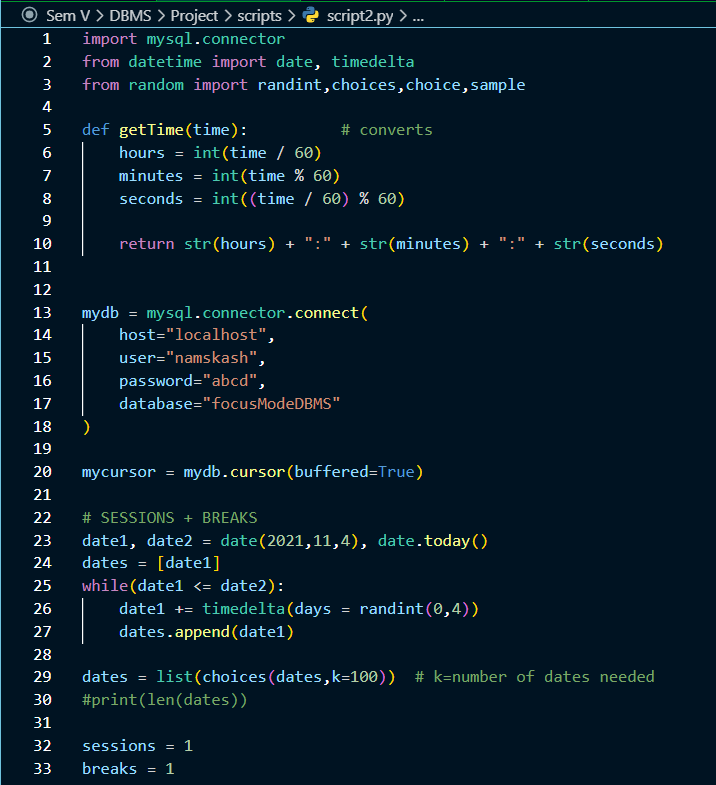
1. **Populating data:**

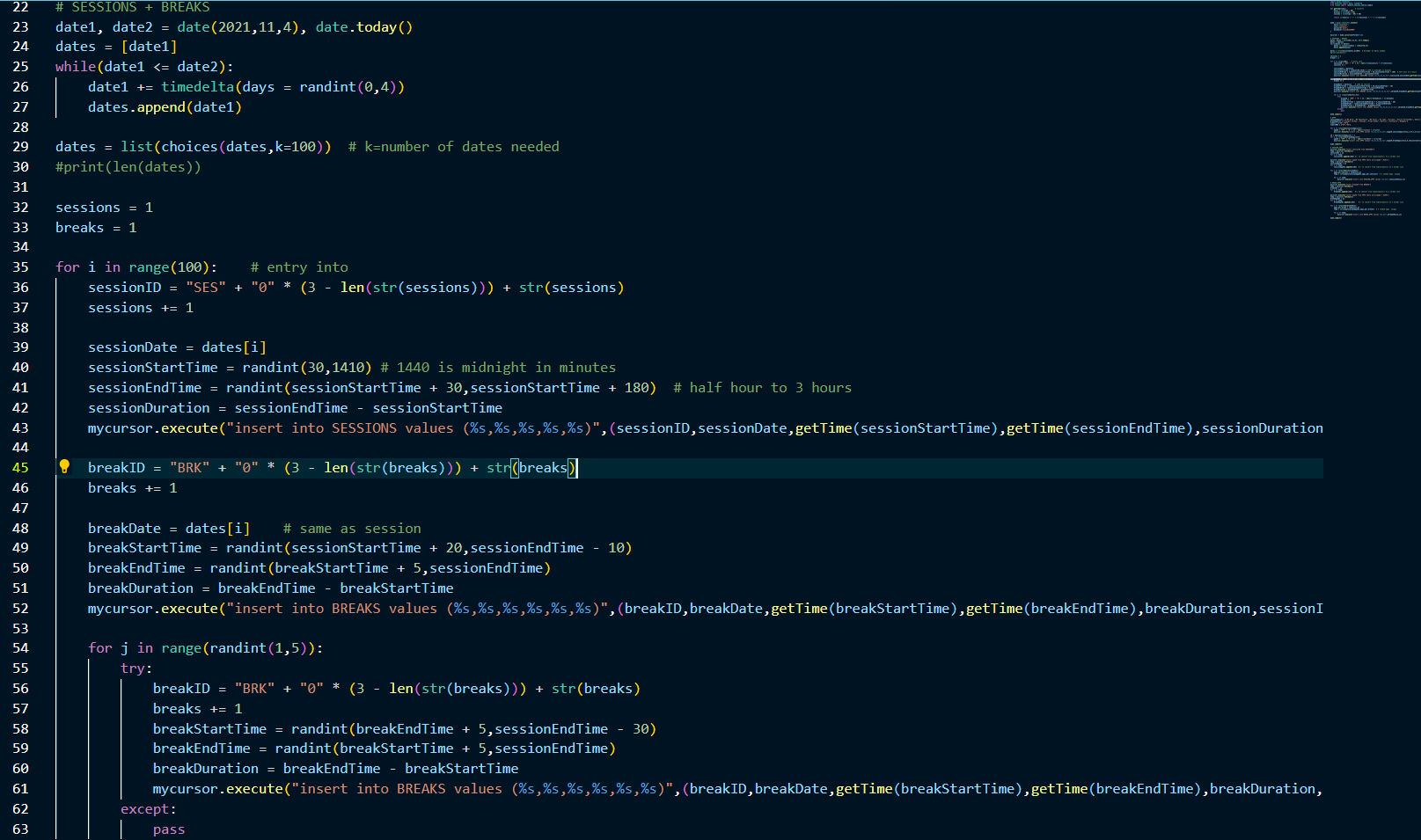
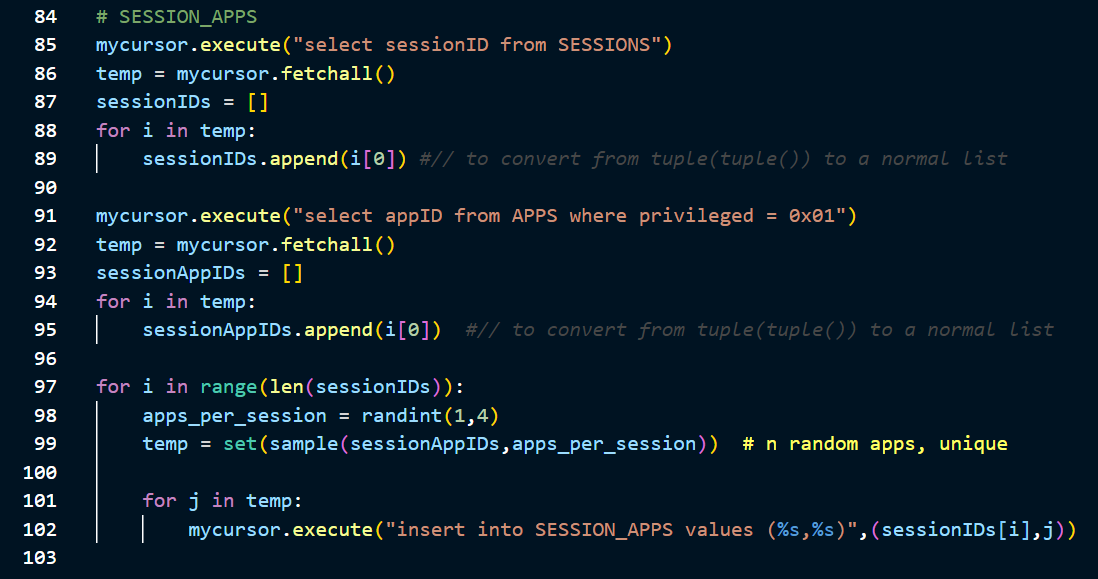
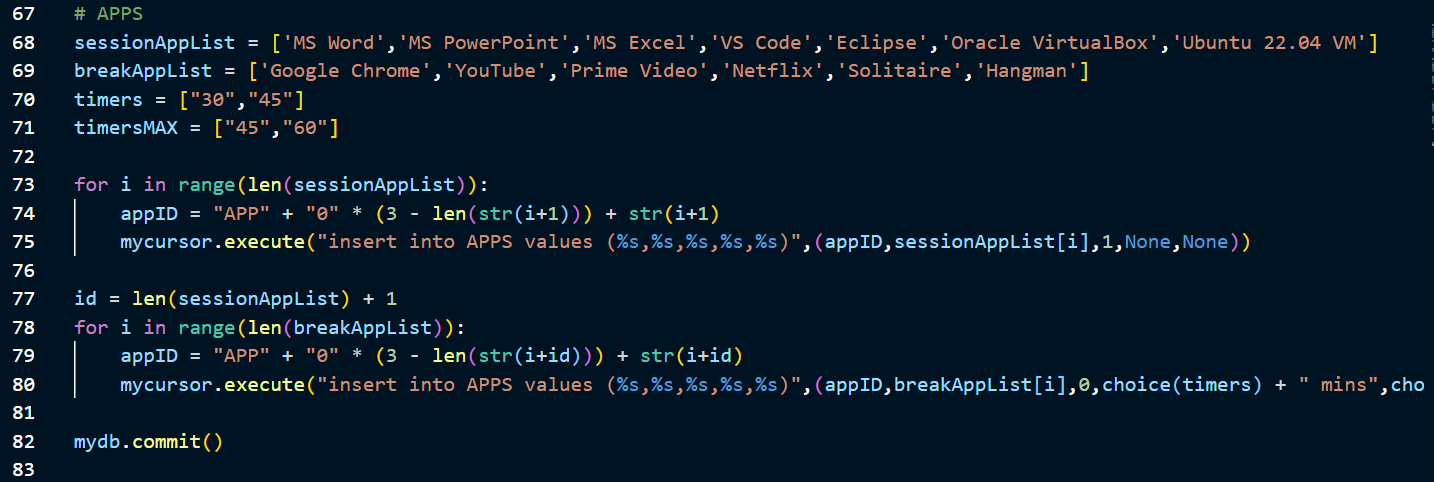
**Python scripts:**

*script1.py* **file:**

**

*script2.py* **file:**

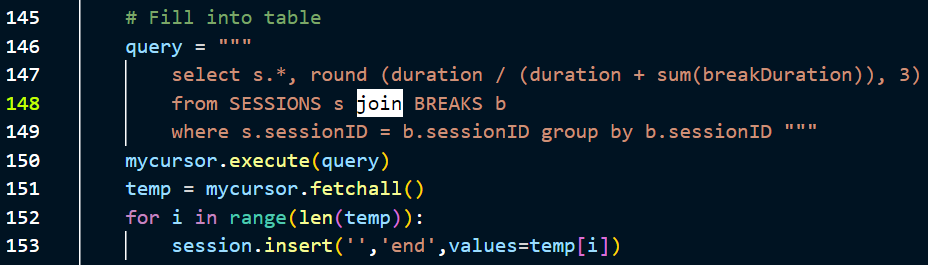
****

******

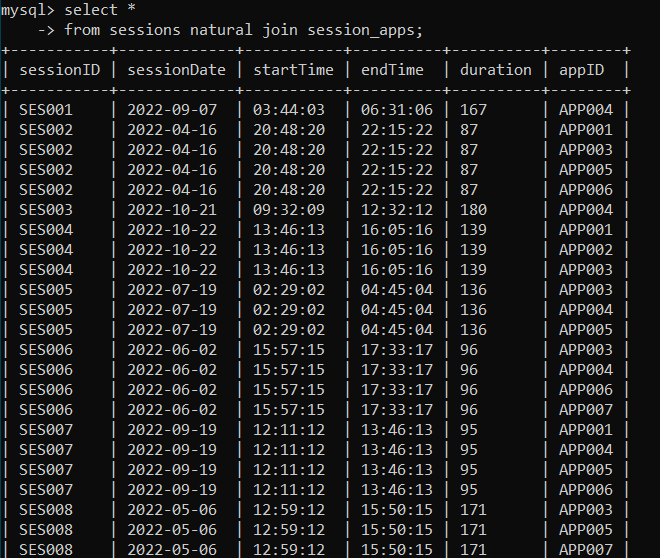
**

1. **JOIN queries:**

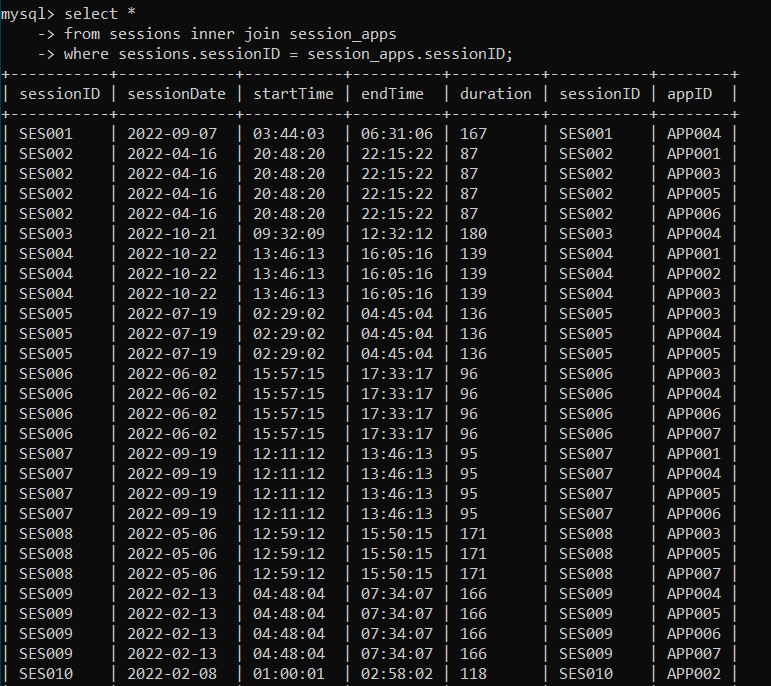
Objective: Display the past sessions page.



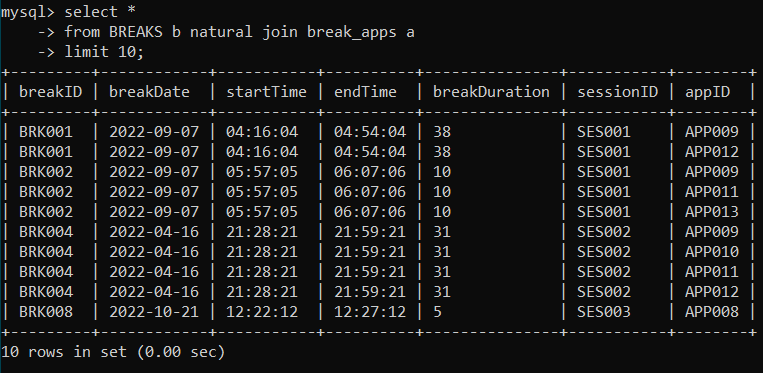
Objective: Get sessions and all apps used in it.



Objective: Get sessions and all apps used using inner join.

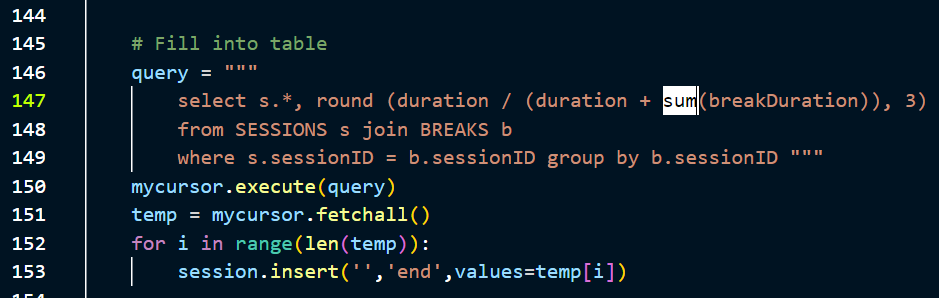


Objective: Get 10 breakIDs and all apps used in it.

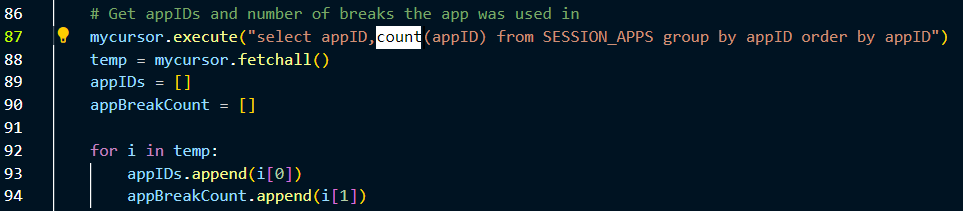


1. **AGGREGATE functions:**

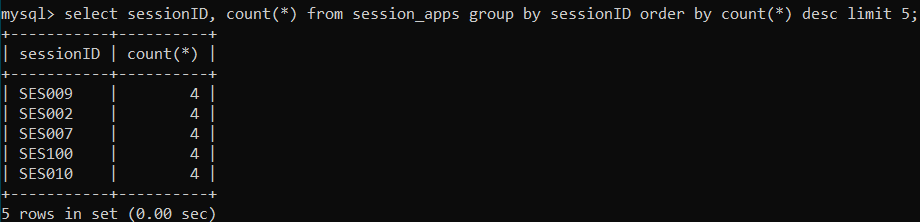
Objective: Get sum of all breakDurations.



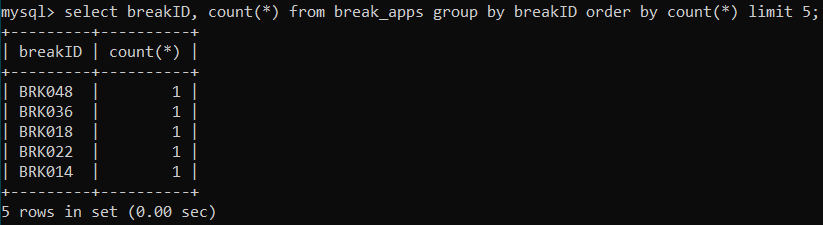
Objective: Count all appIDs from session\_apps



Objective: Count the number of apps in the top 5 sessions in terms of apps used.

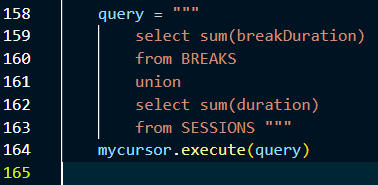


Objective: Count the number of apps in the bottom 5 breaks in terms of apps used.

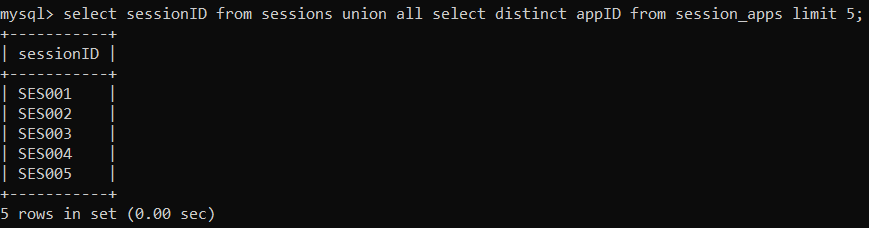


1. **SET operators:**

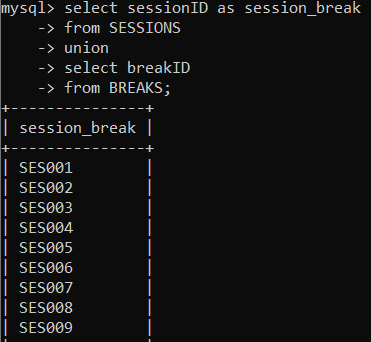
Objective: Get a union of all sessionDurations and breakDurations.



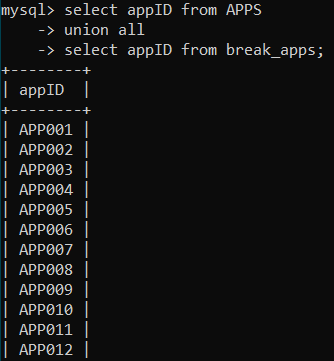
Objective: Get all session IDs in both sessions and session\_apps.



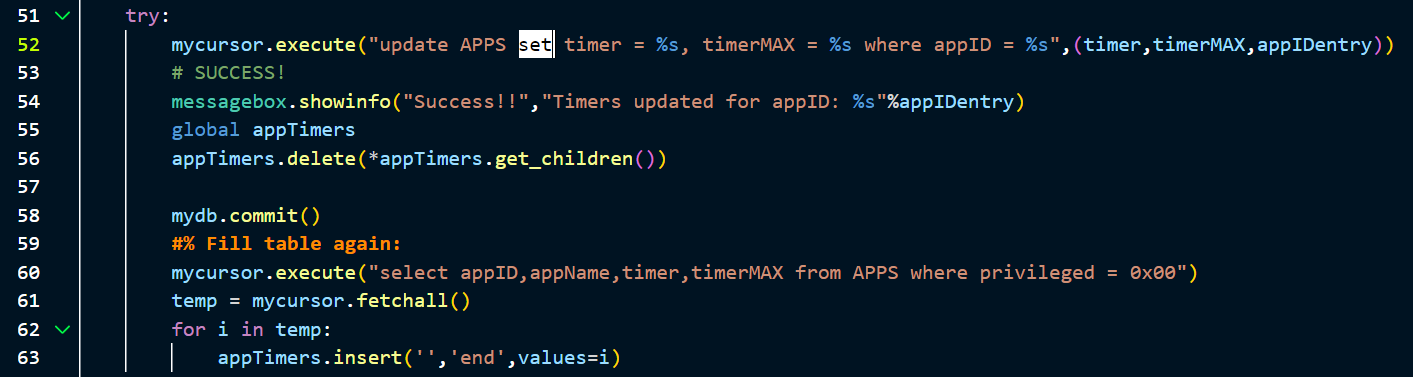
Objective: Get a union of all session IDs and break IDs.



Objective: Get all app IDs from both APPS and break\_apps.



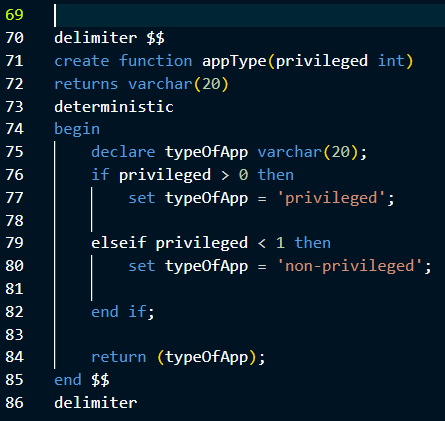
Update:



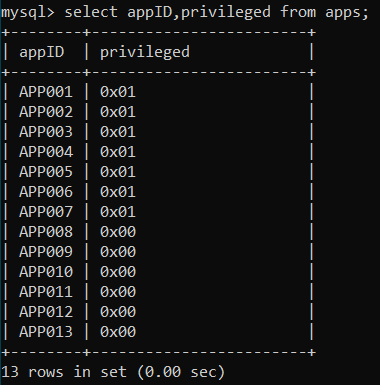
1. **Functions and procedures:**

**Function :**

Objective: To display “privileged” or “not-privileged” instead 0x01 or 0x00.



Without function:

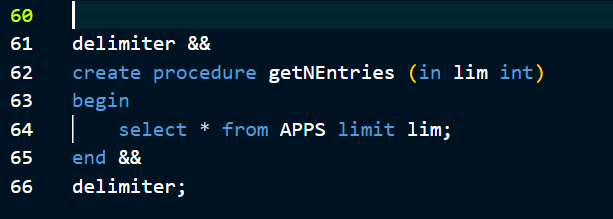


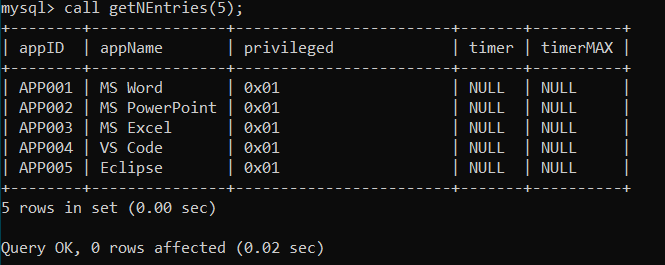
With function:



**Procedure:**

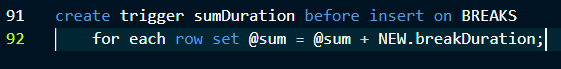
Objective: Get N entries from APPS, n passed as argument.



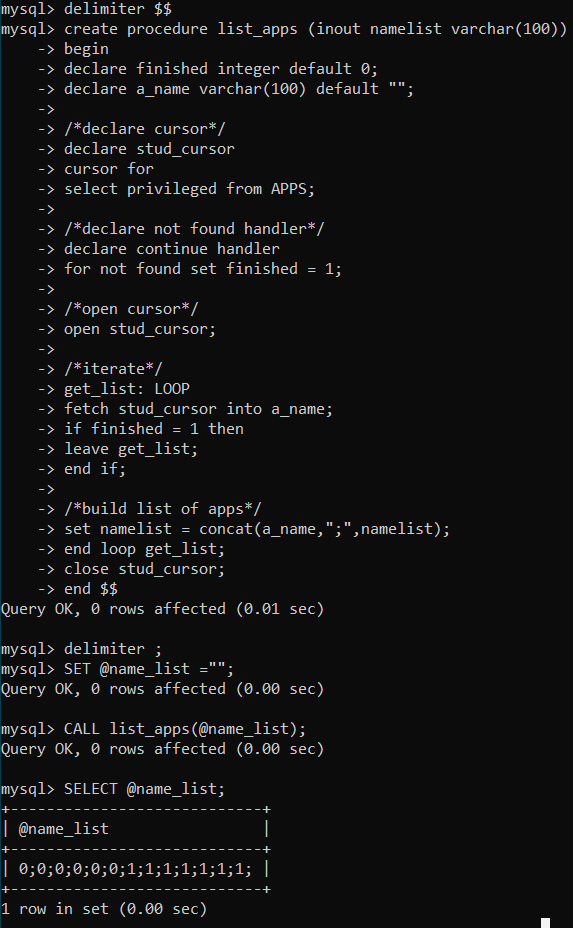


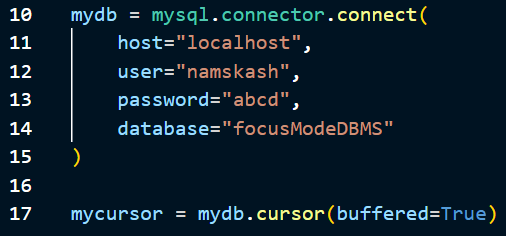
1. **Triggers and cursors:**

Objective: To automatically update variable sum when a new row is inserted.



Objective: To get all the values of privileged bits.





1. **Higher Level Programming (GUI):**

