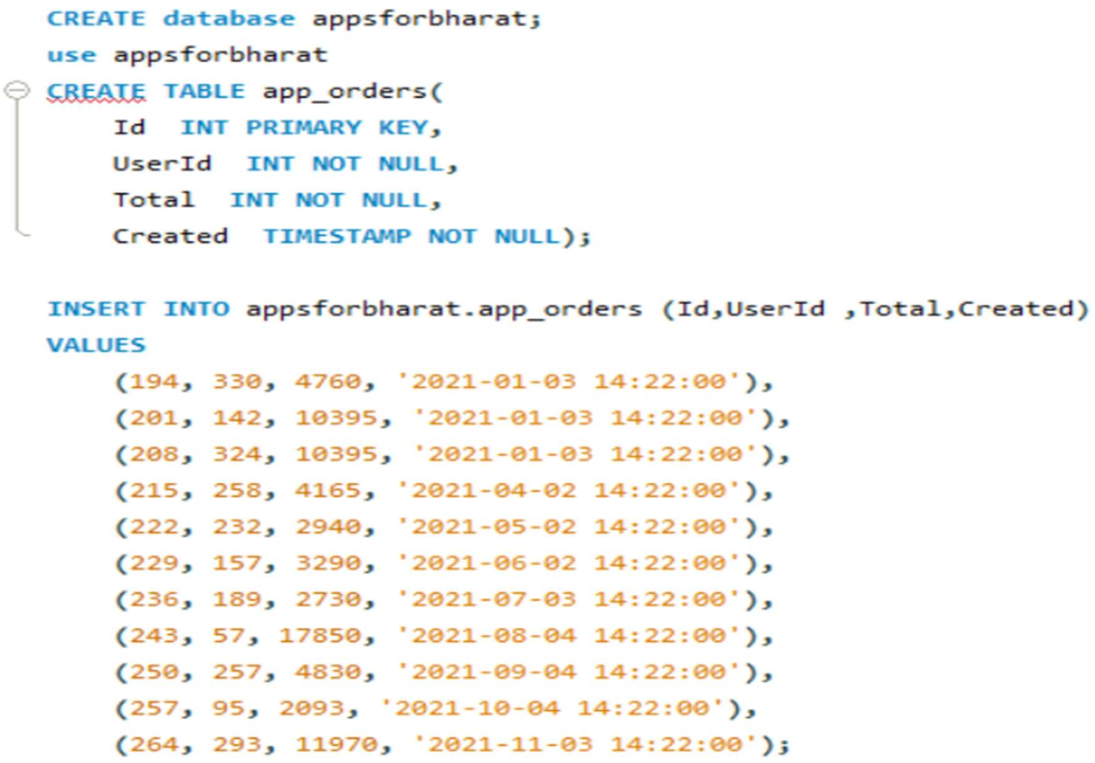
# Problem 1

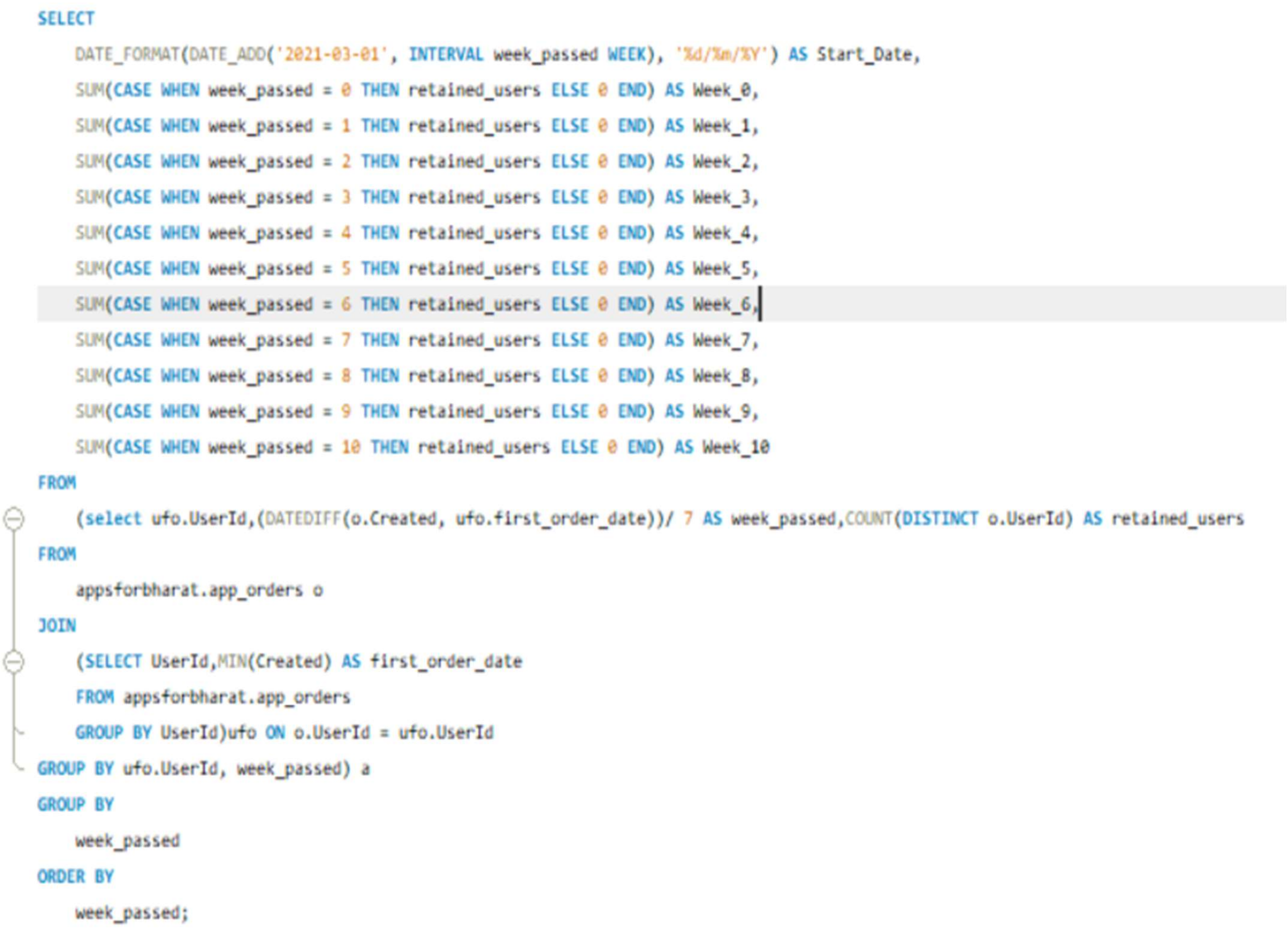
```
CREATE database appsforbharat;
use appsforbharat
CREATE TABLE app_orders(
   Id  INT PRIMARY KEY,
   UserId  INT NOT NULL,
   Total  INT NOT NULL,
   Created  TIMESTAMP NOT NULL);

INSERT INTO appsforbharat.app_orders (Id,UserId ,Total,Created)
VALUES
   (194, 330, 4760, '2021-01-03 14:22:00'),
   (201, 142, 10395, '2021-01-03 14:22:00'),
   (208, 324, 10395, '2021-01-03 14:22:00'),
   (215, 258, 4165, '2021-04-02 14:22:00'),
   (222, 232, 2940, '2021-05-02 14:22:00'),
   (229, 157, 3290, '2021-06-02 14:22:00'),
   (236, 189, 2730, '2021-07-03 14:22:00'),
   (243, 57, 17850, '2021-08-04 14:22:00'),
   (250, 257, 4830, '2021-09-04 14:22:00'),
   (257, 95, 2093, '2021-10-04 14:22:00'),
   (264, 293, 11970, '2021-11-03 14:22:00');
```



```
CREATE database appsforbharat;
use appsforbharat
CREATE TABLE app_orders(
    Id  INT PRIMARY KEY,
    UserId  INT NOT NULL,
    Total  INT NOT NULL,
    Created  TIMESTAMP NOT NULL);

INSERT INTO appsforbharat.app_orders (Id,UserId ,Total,Created)
VALUES
    (194, 330, 4760, '2021-01-03 14:22:00'),
    (201, 142, 10395, '2021-01-03 14:22:00'),
    (208, 324, 10395, '2021-01-03 14:22:00'),
    (215, 258, 4165, '2021-04-02 14:22:00'),
    (222, 232, 2940, '2021-05-02 14:22:00'),
    (229, 157, 3290, '2021-06-02 14:22:00'),
    (236, 189, 2730, '2021-07-03 14:22:00'),
    (243, 57, 17850, '2021-08-04 14:22:00'),
    (250, 257, 4830, '2021-09-04 14:22:00'),
    (257, 95, 2093, '2021-10-04 14:22:00'),
    (264, 293, 11970, '2021-11-03 14:22:00');
```

#Q1. We need SQL query to identify weekly Retention for Users from their first created order Week in the following format.(Kindly mention if any assumptions are made)

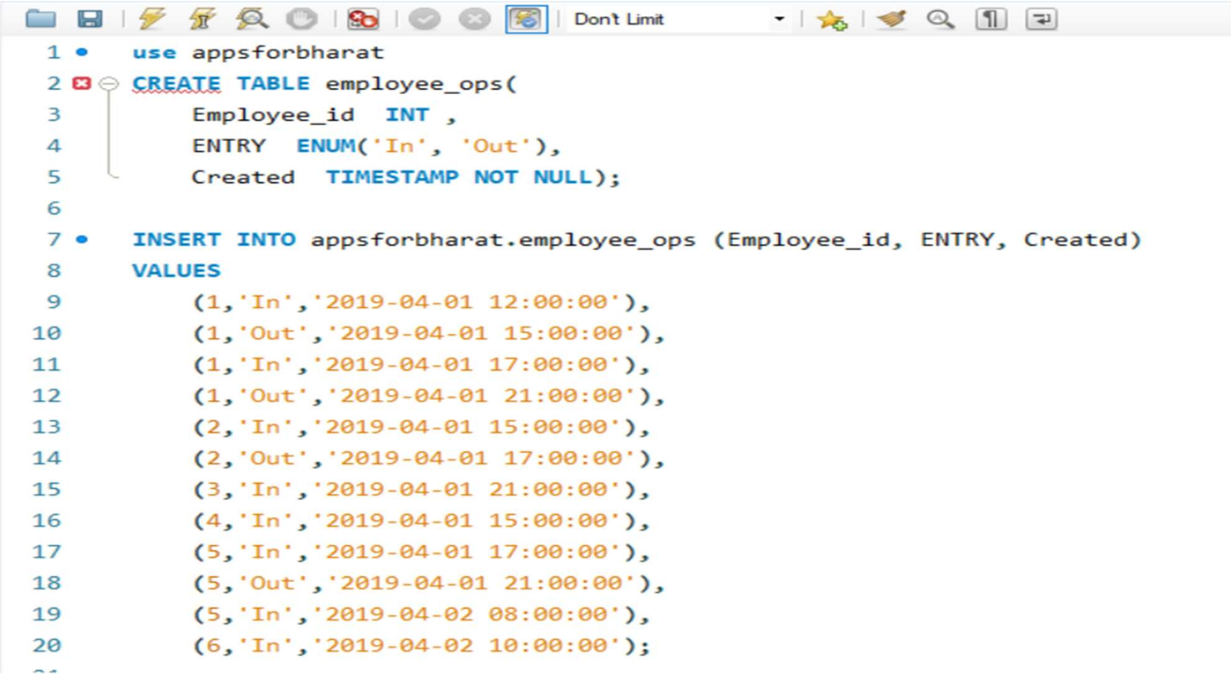
```
USE appsforbharat
SELECT
   DATE_FORMAT(DATE_ADD('2021-03-01', INTERVAL week_passed WEEK), '%d/%m/%Y') AS Start_Date,
   SUM(CASE WHEN week_passed = 0 THEN retained_users ELSE 0 END) AS Week_0,
   SUM(CASE WHEN week_passed = 1 THEN retained_users ELSE 0 END) AS Week_1,
   SUM(CASE WHEN week_passed = 2 THEN retained_users ELSE 0 END) AS Week_2,
   SUM(CASE WHEN week_passed = 3 THEN retained_users ELSE 0 END) AS Week_3,
   SUM(CASE WHEN week_passed = 4 THEN retained_users ELSE 0 END) AS Week_4,
   SUM(CASE WHEN week_passed = 5 THEN retained_users ELSE 0 END) AS Week_5,
   SUM(CASE WHEN week_passed = 6 THEN retained_users ELSE 0 END) AS Week_6,
   SUM(CASE WHEN week_passed = 7 THEN retained_users ELSE 0 END) AS Week_7,
   SUM(CASE WHEN week_passed = 8 THEN retained_users ELSE 0 END) AS Week_8,
   SUM(CASE WHEN week_passed = 9 THEN retained_users ELSE 0 END) AS Week_9,
   SUM(CASE WHEN week_passed = 10 THEN retained_users ELSE 0 END) AS Week_10
FROM
   (select ufo.UserId,(DATEDIFF(o.Created, ufo.first_order_date))/ 7 AS week_passed,COUNT(DISTINCT o.UserId) AS retained_users
FROM
   appsforbharat.app_orders o
JOIN
   (SELECT UserId,MIN(Created) AS first_order_date
   FROM appsforbharat.app_orders
   GROUP BY UserId)ufo ON o.UserId = ufo.UserId
GROUP BY ufo.UserId, week_passed) a
GROUP BY
   week_passed
ORDER BY
   week_passed;
```



```sql
SELECT
    DATE_FORMAT(DATE_ADD('2021-03-01', INTERVAL week_passed WEEK), '%d/%m/%Y') AS Start_Date,
    SUM(CASE WHEN week_passed = 0 THEN retained_users ELSE 0 END) AS Week_0,
    SUM(CASE WHEN week_passed = 1 THEN retained_users ELSE 0 END) AS Week_1,
    SUM(CASE WHEN week_passed = 2 THEN retained_users ELSE 0 END) AS Week_2,
    SUM(CASE WHEN week_passed = 3 THEN retained_users ELSE 0 END) AS Week_3,
    SUM(CASE WHEN week_passed = 4 THEN retained_users ELSE 0 END) AS Week_4,
    SUM(CASE WHEN week_passed = 5 THEN retained_users ELSE 0 END) AS Week_5,
    SUM(CASE WHEN week_passed = 6 THEN retained_users ELSE 0 END) AS Week_6,
    SUM(CASE WHEN week_passed = 7 THEN retained_users ELSE 0 END) AS Week_7,
    SUM(CASE WHEN week_passed = 8 THEN retained_users ELSE 0 END) AS Week_8,
    SUM(CASE WHEN week_passed = 9 THEN retained_users ELSE 0 END) AS Week_9,
    SUM(CASE WHEN week_passed = 10 THEN retained_users ELSE 0 END) AS Week_10
FROM
    (select ufo.UserId,(DATEDIFF(o.Created, ufo.first_order_date))/ 7 AS week_passed,COUNT(DISTINCT o.UserId) AS retained_users
FROM
    appsforbharat.app_orders o
JOIN
    (SELECT UserId,MIN(Created) AS first_order_date
    FROM appsforbharat.app_orders
    GROUP BY UserId)ufo ON o.UserId = ufo.UserId
GROUP BY ufo.UserId, week_passed) a
GROUP BY
    week_passed
ORDER BY
    week_passed;
```

# Problem 2

```
use appsforbharat
CREATE TABLE employee_ops(
   Employee_id  INT ,
   ENTRY  ENUM('In', 'Out'),
   Created  TIMESTAMP NOT NULL);

INSERT INTO appsforbharat.employee_ops (Employee_id, ENTRY, Created)
VALUES
   (1,'In','2019-04-01 12:00:00'),
            (1,'Out','2019-04-01 15:00:00'),
            (1,'In','2019-04-01 17:00:00'),
            (1,'Out','2019-04-01 21:00:00'),
            (2,'In','2019-04-01 15:00:00'),
            (2,'Out','2019-04-01 17:00:00'),
            (3,'In','2019-04-01 21:00:00'),
            (4,'In','2019-04-01 15:00:00'),
            (5,'In','2019-04-01 17:00:00'),
   (5,'Out','2019-04-01 21:00:00'),
            (5,'In','2019-04-02 08:00:00'),
            (6,'In','2019-04-02 10:00:00');
```

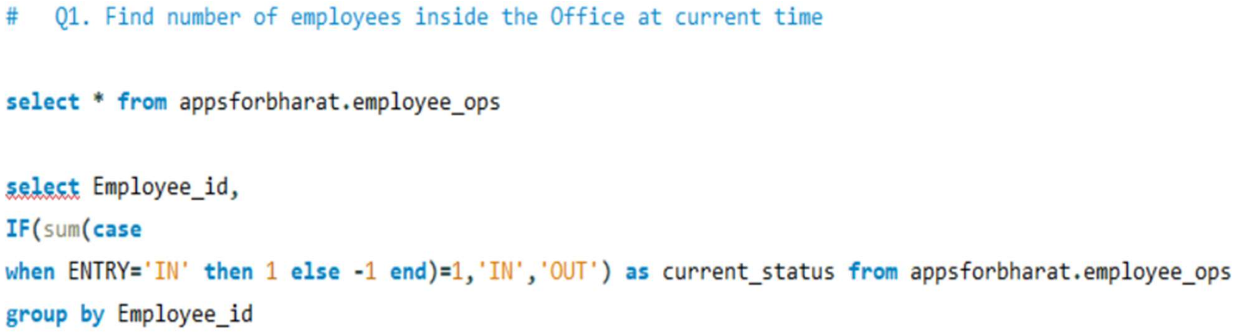


```
use appsforbharat
CREATE TABLE employee_ops(
    Employee_id  INT ,
    ENTRY  ENUM('In', 'Out'),
    Created  TIMESTAMP NOT NULL);

INSERT INTO appsforbharat.employee_ops (Employee_id, ENTRY, Created)
VALUES
    (1,'In','2019-04-01 12:00:00'),
    (1,'Out','2019-04-01 15:00:00'),
    (1,'In','2019-04-01 17:00:00'),
    (1,'Out','2019-04-01 21:00:00'),
    (2,'In','2019-04-01 15:00:00'),
    (2,'Out','2019-04-01 17:00:00'),
    (3,'In','2019-04-01 21:00:00'),
    (4,'In','2019-04-01 15:00:00'),
    (5,'In','2019-04-01 17:00:00'),
    (5,'Out','2019-04-01 21:00:00'),
    (5,'In','2019-04-02 08:00:00'),
    (6,'In','2019-04-02 10:00:00');
```

##########################################################################################

# Q1. Find number of employees inside the Office at current time

```
select Employee_id,
IF(sum(case
when ENTRY='IN' then 1 else -1 end)=1,'IN','OUT') as current_status from appsforbharat.employee_ops
group by Employee_id
```

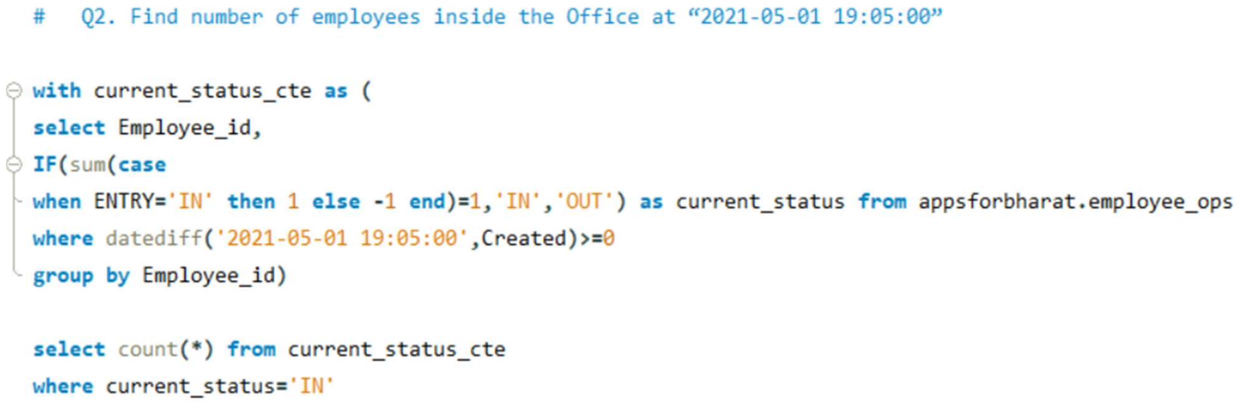


```
#   Q1. Find number of employees inside the Office at current time


select * from appsforbharat.employee_ops


select Employee_id,
IF(sum(case
when ENTRY='IN' then 1 else -1 end)=1,'IN','OUT') as current_status from appsforbharat.employee_ops
group by Employee_id
```

#########################################################################################

#   Q2. Find number of employees inside the Office at “2021-05-01 19:05:00”

```
with current_status_cte as (
select Employee_id,
IF(sum(case
when ENTRY='IN' then 1 else -1 end)=1,'IN','OUT') as current_status from appsforbharat.employee_ops
where datediff('2021-05-01 19:05:00',Created)>=0
group by Employee_id)
select count(*) from current_status_cte
where current_status='IN'
```

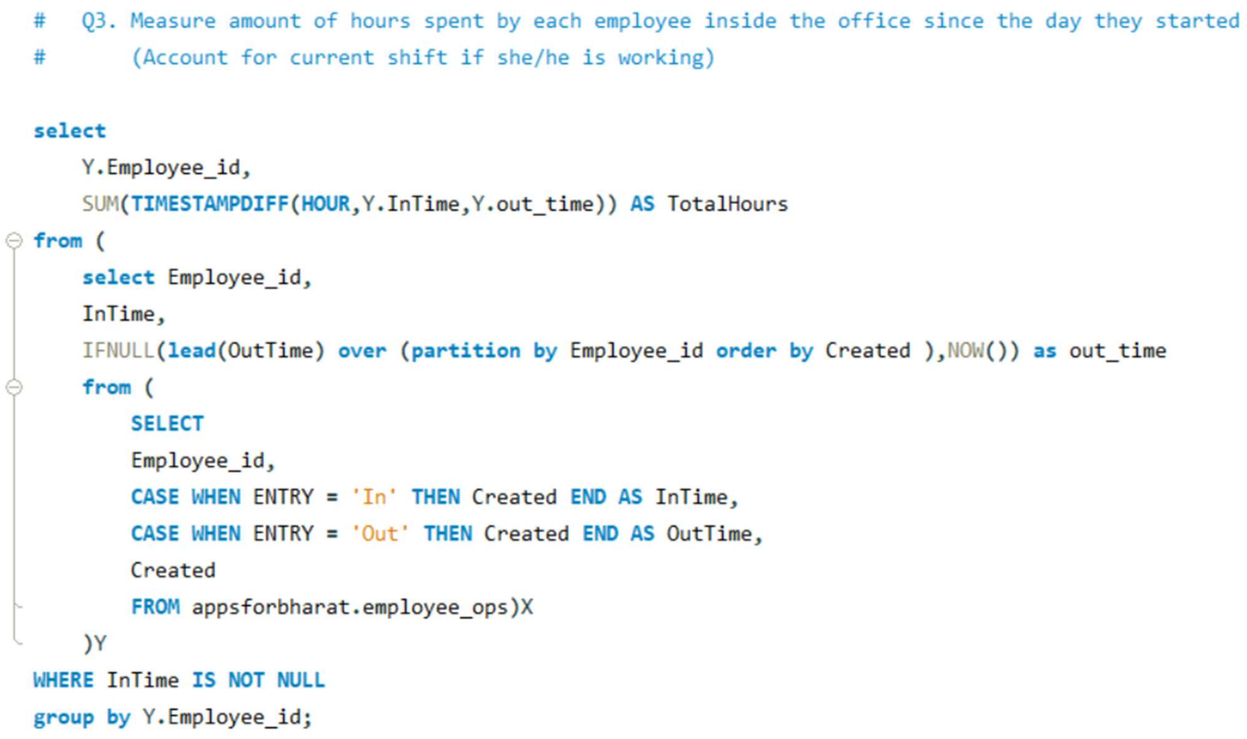


```
#   Q2. Find number of employees inside the Office at “2021-05-01 19:05:00”

with current_status_cte as (
select Employee_id,
IF(sum(case
when ENTRY='IN' then 1 else -1 end)=1,'IN','OUT') as current_status from appsforbharat.employee_ops
where datediff('2021-05-01 19:05:00',Created)>=0
group by Employee_id)

select count(*) from current_status_cte
where current_status='IN'
```

#########################################################################################

#   Q3. Measure amount of hours spent by each employee inside the office since the day they started (Account for current shift if she/he is working)

```
select
        Y.Employee_id,
        SUM(TIMESTAMPDIFF(HOUR,Y.InTime,Y.out_time)) AS TotalHours
from (
        select Employee_id,InTime,
  IFNULL(lead(OutTime) over (partition by Employee_id order by Created ),NOW()) as out_time
        from (
                SELECT
                Employee_id,
                CASE WHEN ENTRY = 'In' THEN Created END AS InTime,
                CASE WHEN ENTRY = 'Out' THEN Created END AS OutTime,
    Created
                FROM appsforbharat.employee_ops)X
        )Y
WHERE InTime IS NOT NULL
group by Y.Employee_id;
```



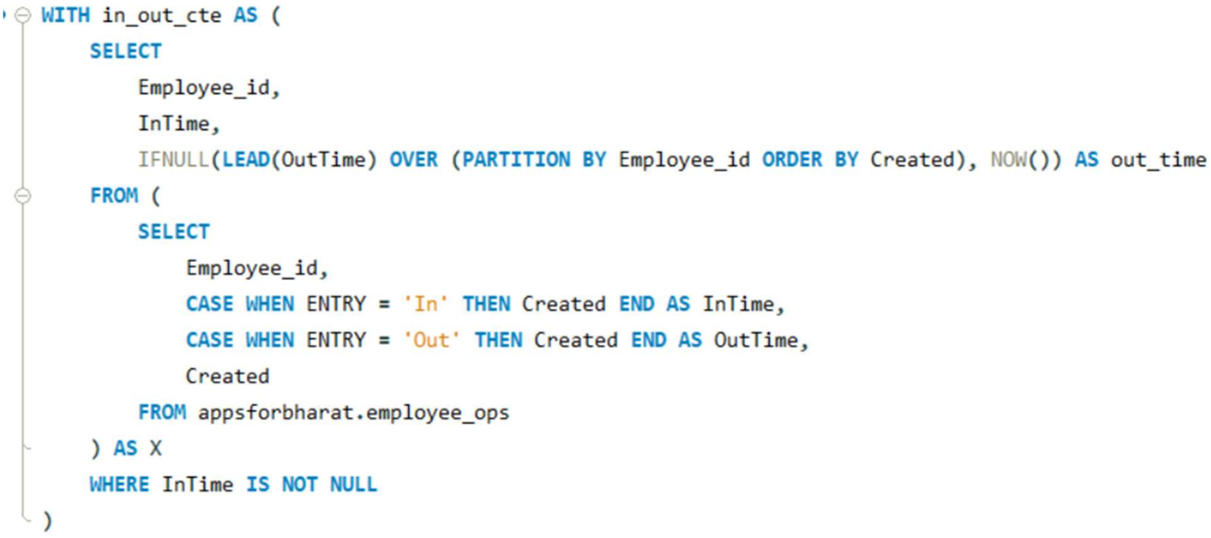
```
#   Q3. Measure amount of hours spent by each employee inside the office since the day they started
#       (Account for current shift if she/he is working)

select
    Y.Employee_id,
    SUM(TIMESTAMPDIFF(HOUR,Y.InTime,Y.out_time)) AS TotalHours
from (
    select Employee_id,
    InTime,
    IFNULL(lead(OutTime) over (partition by Employee_id order by Created ),NOW()) as out_time
    from (
        SELECT
        Employee_id,
        CASE WHEN ENTRY = 'In' THEN Created END AS InTime,
        CASE WHEN ENTRY = 'Out' THEN Created END AS OutTime,
        Created
        FROM appsforbharat.employee_ops)X
    )Y
WHERE InTime IS NOT NULL
group by Y.Employee_id;
```

##########################################################################################

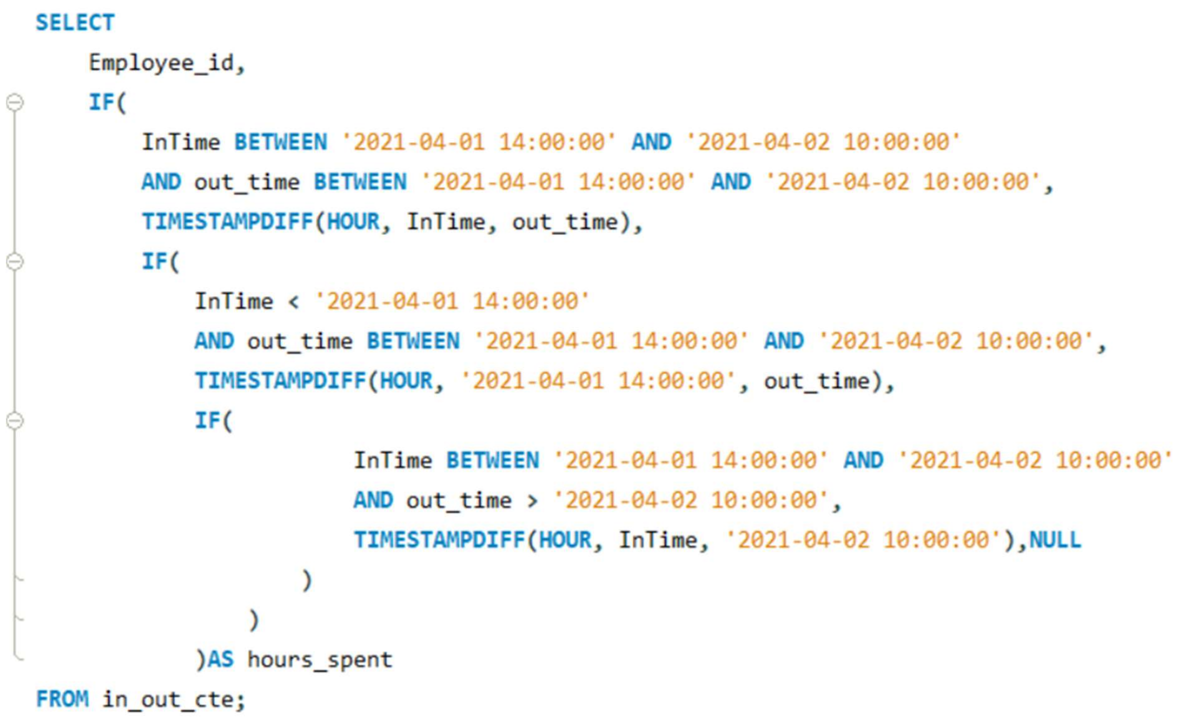
# Q4. Measure amount of hours spent by each employee inside the office between “2021-04-01 14:00:00” and “2021-04-02 10:00:00”

```
WITH in_out_cte AS (
  SELECT
    Employee_id,
    InTime,
    IFNULL(LEAD(OutTime) OVER (PARTITION BY Employee_id ORDER BY Created), NOW()) AS out_time
  FROM (
    SELECT
      Employee_id,
      CASE WHEN ENTRY = 'In' THEN Created END AS InTime,
      CASE WHEN ENTRY = 'Out' THEN Created END AS OutTime,
      Created
    FROM appsforbharat.employee_ops
  ) AS X
  WHERE InTime IS NOT NULL
)
```

```
SELECT
    Employee_id,
    IF(
        InTime BETWEEN '2021-04-01 14:00:00' AND '2021-04-02 10:00:00'
        AND out_time BETWEEN '2021-04-01 14:00:00' AND '2021-04-02 10:00:00',
        TIMESTAMPDIFF(HOUR, InTime, out_time),
        IF(
            InTime < '2021-04-01 14:00:00'
            AND out_time BETWEEN '2021-04-01 14:00:00' AND '2021-04-02 10:00:00',
            TIMESTAMPDIFF(HOUR, '2021-04-01 14:00:00', out_time),
            IF(
                        InTime BETWEEN '2021-04-01 14:00:00' AND '2021-04-02 10:00:00'
                        AND out_time > '2021-04-02 10:00:00',
                        TIMESTAMPDIFF(HOUR, InTime, '2021-04-02 10:00:00'),NULL
                    )
                )
            )AS hours_spent
FROM in_out_cte;
```



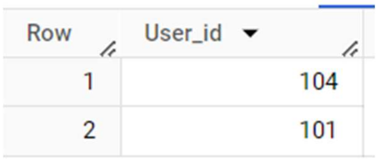
```
WITH in_out_cte AS (
    SELECT
        Employee_id,
        InTime,
        IFNULL(LEAD(OutTime) OVER (PARTITION BY Employee_id ORDER BY Created), NOW()) AS out_time
    FROM (
        SELECT
            Employee_id,
            CASE WHEN ENTRY = 'In' THEN Created END AS InTime,
            CASE WHEN ENTRY = 'Out' THEN Created END AS OutTime,
            Created
        FROM appsforbharat.employee_ops
    ) AS X
    WHERE InTime IS NOT NULL
)
```



```
SELECT
    Employee_id,
    IF(
        InTime BETWEEN '2021-04-01 14:00:00' AND '2021-04-02 10:00:00'
        AND out_time BETWEEN '2021-04-01 14:00:00' AND '2021-04-02 10:00:00',
        TIMESTAMPDIFF(HOUR, InTime, out_time),
        IF(
            InTime < '2021-04-01 14:00:00'
            AND out_time BETWEEN '2021-04-01 14:00:00' AND '2021-04-02 10:00:00',
            TIMESTAMPDIFF(HOUR, '2021-04-01 14:00:00', out_time),
            IF(
                        InTime BETWEEN '2021-04-01 14:00:00' AND '2021-04-02 10:00:00'
                        AND out_time > '2021-04-02 10:00:00',
                        TIMESTAMPDIFF(HOUR, InTime, '2021-04-02 10:00:00'),NULL
                    )
                )
            )AS hours_spent
FROM in_out_cte;
```

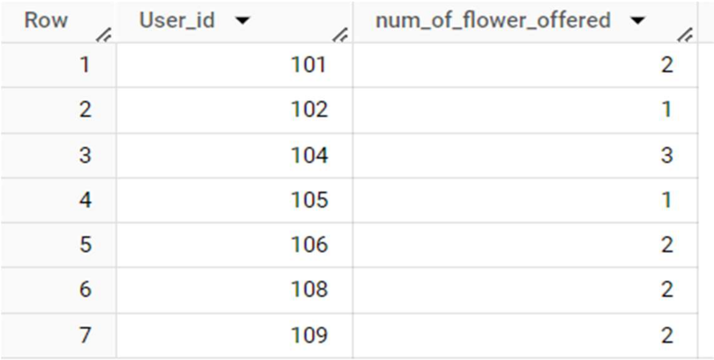
# Problem 3

```
# Q1. DAU who clicked on Flower Icon on Mandir Home %
#We need to identify users who clicked on the Flower Icon (mandir_flower) after visiting the
Mandir Home (mandir_home).

select distinct User_id
from (select *,lead(events)over (partition by User_id order by TimeStamp) as second_menu
from `sri_mandir_app.event_details` where events in ('mandir_home','mandir_flower'))
where second_menu='mandir_flower'
```



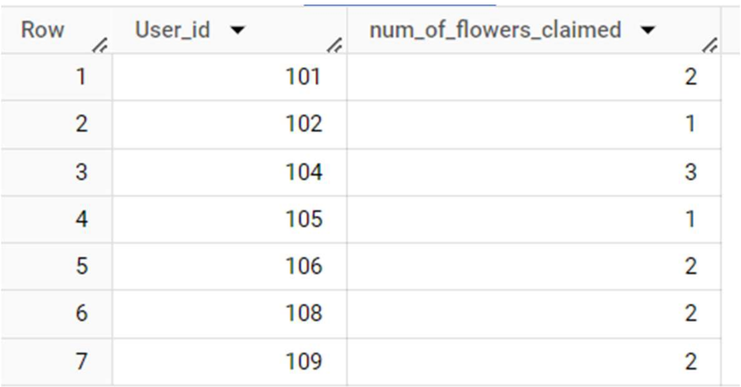
| Row | User_id |
|---|---|
| 1 | 104 |
| 2 | 101 |

```
###############################################################################################
# Q2. DAU who Offered at least 1 Flower %

select distinct User_id,count(Flower_Id) as num_of_flower_offered
from `sri_mandir_app.flowers_offered`
group by User_id
having count(Flower_Id)>=1
```



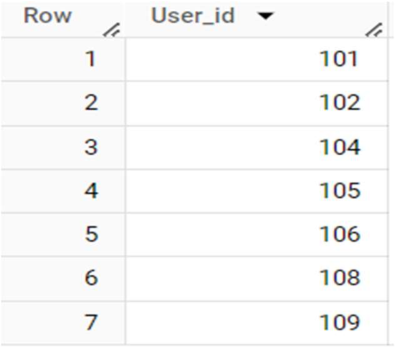
| Row | User_id | num_of_flower_offered |
|---|---|---|
| 1 | 101 | 2 |
| 2 | 102 | 1 |
| 3 | 104 | 3 |
| 4 | 105 | 1 |
| 5 | 106 | 2 |
| 6 | 108 | 2 |
| 7 | 109 | 2 |

```
##############################################################################################
Q3. DAU who claimed at least 1 Flower %

select  User_id,count(Flower_Id) as num_of_flowers_claimed
from `sri_mandir_app.flowers_claimed`
group by User_id
having count(Flower_Id)>=1
```



| Row | User_id | num_of_flowers_claimed |
|---|---|---|
| 1 | 101 | 2 |
| 2 | 102 | 1 |
| 3 | 104 | 3 |
| 4 | 105 | 1 |
| 5 | 106 | 2 |
| 6 | 108 | 2 |
| 7 | 109 | 2 |

```
#############################################################################################
#Q4. DAU who clicked on Locked Flowers %

select distinct User_id
from `sri_mandir_app.flowers_offered`
where condition='locked'
```



| Row | User_id |
|---|---|
| 1 | 101 |
| 2 | 102 |
| 3 | 104 |
| 4 | 105 |
| 5 | 106 |
| 6 | 108 |
| 7 | 109 |

## Additional Insights

Users with user_id 101 and 104 clicked on flower icon after visiting mandir home.

User 104 offered most number of flowers.

User 104 claimed most number of flowers.

All DAU have have clicked on locked condition