DB PROGRAMMING
RED TEAM

PHASE 2 – POPULATE, TEST, AND USE OPERATIONAL DATABASE

Akshay Jadhav

Jaymeen Gandhi

Tejas Joshi

Anmol Kansara

Rashmi Jain

Mayuri Khade

Agenda

- Project Requirements

- Approach Used

- Group Summary and Conclusion

- Suggestions

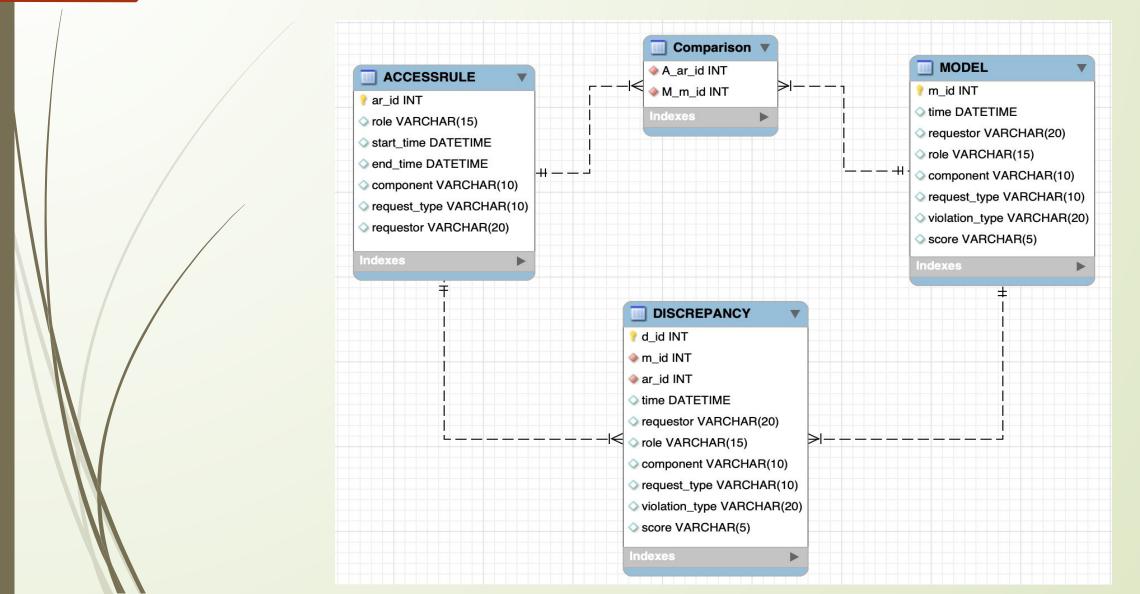# Project Requirements

Jaymeen

- To populate the Model, Access Rule tables with given data sets

- Implement at least two approaches to compare pertinent attributes between MODEL and ACCESS RULE tables and to gather mismatched records from MODEL table called 'evidence of discrepancies' into DISCREPANCY table for better evaluation of discrepancies.

- To select best approach to implement the comparison operation in the given system.

Approach Used

- **LEFT JOIN** Approach

- **CURSOR** Approach
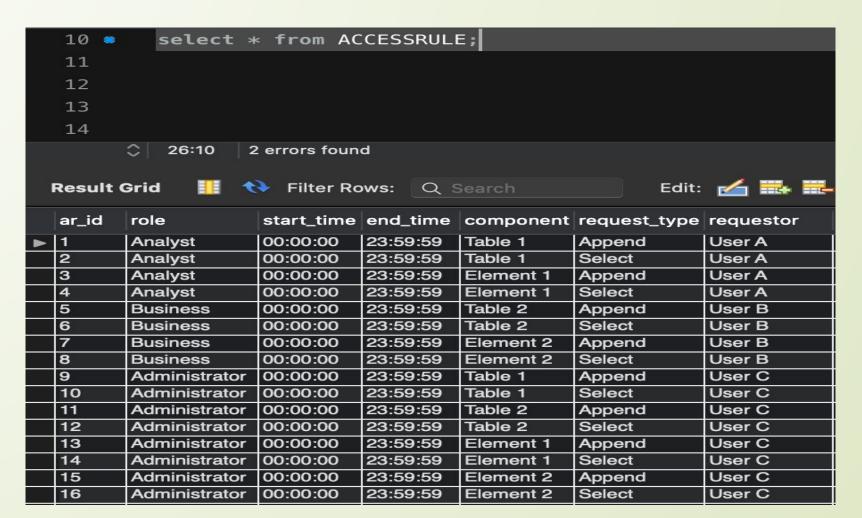
LEFT JOIN (Updated ERD)
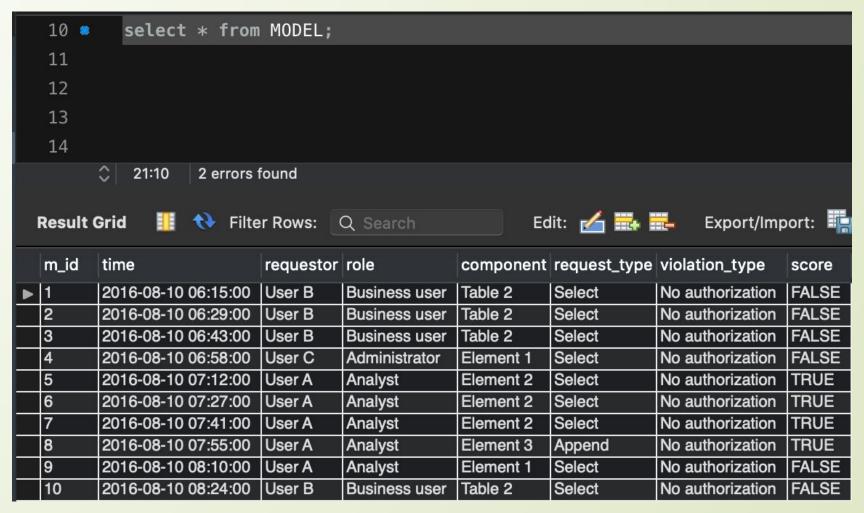
LEFT JOIN (Pseudo Code)

Akshay

- Fetch all MODEL records from MODEL table for score is TRUE.

- Using JOIN approach, performed comparison of pertinent attributes by comparing each row of MODEL with each row of ACCESS RULE table, to get all mismatched rows(called "Evidences of Discrepancy)

- Inserted mismatched rows between MODEL and ACCESS RULE table into DISCREPANCY table after comparison to gather the evidences of discrepancy.

- Created a stored procedure to gather evidence of discrepancies happened during comparison between MODEL and ACCESS RULE table.

Population of Tables

- **ACCESSRULE**
  - Total rows: 16

```
10 *     select * from ACCESSRULE;
11
12
13
14
```

26:10 | 2 errors found

Result Grid | Filter Rows: | Q Search | Edit:

ar_id	role	start_time	end_time	component	request_type	requestor
1	Analyst	00:00:00	23:59:59	Table 1	Append	User A
2	Analyst	00:00:00	23:59:59	Table 1	Select	User A
3	Analyst	00:00:00	23:59:59	Element 1	Append	User A
4	Analyst	00:00:00	23:59:59	Element 1	Select	User A
5	Business	00:00:00	23:59:59	Table 2	Append	User B
6	Business	00:00:00	23:59:59	Table 2	Select	User B
7	Business	00:00:00	23:59:59	Element 2	Append	User B
8	Business	00:00:00	23:59:59	Element 2	Select	User B
9	Administrator	00:00:00	23:59:59	Table 1	Append	User C
10	Administrator	00:00:00	23:59:59	Table 1	Select	User C
11	Administrator	00:00:00	23:59:59	Table 2	Append	User C
12	Administrator	00:00:00	23:59:59	Table 2	Select	User C
13	Administrator	00:00:00	23:59:59	Element 1	Append	User C
14	Administrator	00:00:00	23:59:59	Element 1	Select	User C
15	Administrator	00:00:00	23:59:59	Element 2	Append	User C
16	Administrator	00:00:00	23:59:59	Element 2	Select	User C

Population of Tables

- **MODEL**
  - **Total rows: 20266**

LEFT JOIN (Implementation)

- For each row from MODEL table having score TRUE, fetch each row from ACCESS RULE table and insert ar_id, m_id into Comparison table.
  - **Row Count: 38080**

```
38    insert into Comparison (A_ar_id,M_m_id)
39    select ar_id, m_id
40    from ACCESSRULE, MODEL
41    where score = 'TRUE';
42
43    select * from Comparison;
```
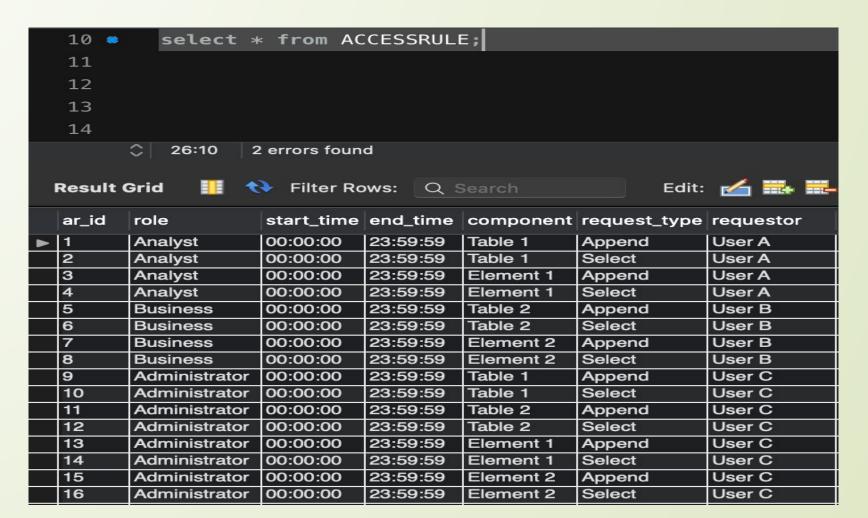
100%   1:44

Result Grid   Filter Rows: Search   Export

A_ar_id	M_m_id
1	5
2	5
3	5
4	5
5	5
6	5
7	5

LEFT JOIN (Implementation)

Akshay

- Created a stored procedure insertComparison() for the same to insert records into Comparison table from both MODEL and ACCESS RULE table.

```
31      DELIMITER //
32
33 •    create procedure insertComparison()
34 ⊖ BEGIN
35
36      insert into Comparison (A_ar_id,M_m_id)
37      select ar_id, m_id
38      from ACCESSRULE, MODEL
39      where score = 'TRUE';
40
41    END //
42
43      DELIMITER ;
```

LEFT JOIN (Implementation)

- Fetched all mismatched rows (evidences of discrepancy) from MODEL table after comparison of pertinent attributes from both MODEL table and ACCESS RULE table.
  - **Mismatched count (Evidence of Discrepancy): 9860**

```
36    select m.m_id,c.A_ar_id,m.time,m.requestor,m.role,m.component,m.request_type,m.violation_type,m.score
37    from
38    MODEL m
39    left join Comparison c
40    on m.m_id = c.M_m_id
41    left join ACCESSRULE ar
42    on ar.ar_id = c.A_ar_id
43    where
44    m.role <> ar.role
45    and m.requestor <> ar.requestor
46    and m.request_type <> ar.request_type
47    and m.time NOT between ar.start_time and ar.end_time
48    and m.component <> ar.component
49    order by m.m_id, ar.ar_id;
```

27:49

Result Grid | Filter Rows: | Search | Export: | Fetch rows:

m_id	A_ar_id	time	requestor	role	component	request_type	violation_type	score
5	5	2016-08-10 07:12:00	User A	Analyst	Element 2	Select	No authorization	TRUE
5	9	2016-08-10 07:12:00	User A	Analyst	Element 2	Select	No authorization	TRUE
5	11	2016-08-10 07:12:00	User A	Analyst	Element 2	Select	No authorization	TRUE
5	13	2016-08-10 07:12:00	User A	Analyst	Element 2	Select	No authorization	TRUE
6	5	2016-08-10 07:27:00	User A	Analyst	Element 2	Select	No authorization	TRUE
6	9	2016-08-10 07:27:00	User A	Analyst	Element 2	Select	No authorization	TRUE
6	11	2016-08-10 07:27:00	User A	Analyst	Element 2	Select	No authorization	TRUE

Population of Tables
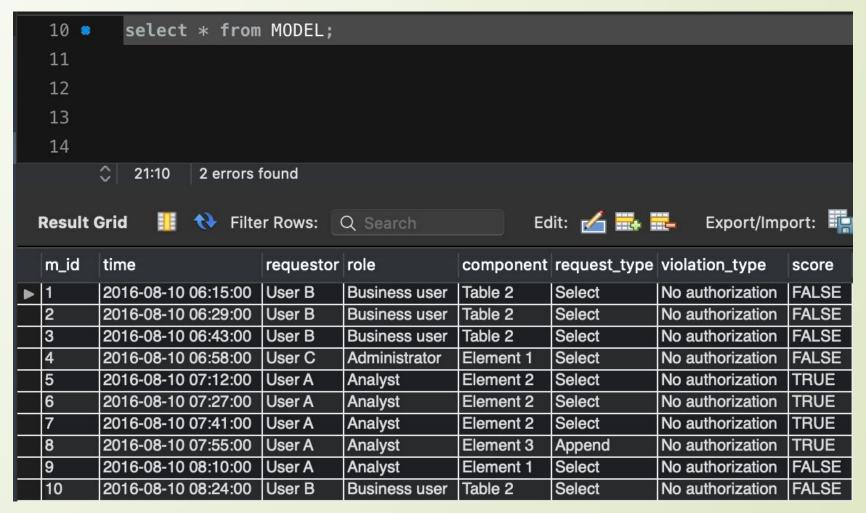
- **ACCESSRULE**
  - Total rows: 16

LEFT JOIN (Implementation)

- Created a stored procedure i.e. getEvidence() to get all evidence of discrepancies happened during comparison between MODEL and ACCESS RULE table.

```
DELIMITER //

create procedure getEvidence()
BEGIN

call insertComparison();

insert into DISCREPANCY (m_id,ar_id,time,requestor,role,component,request_type,violation_type,score)
select m.m_id,c.A_ar_id,m.time,m.requestor,m.role,m.component,m.request_type,m.violation_type,m.score
from
MODEL m
left join Comparison c
on m.m_id = c.M_m_id
left join ACCESSRULE ar
on ar.ar_id = c.A_ar_id
where
m.role <> ar.role
and m.requestor <> ar.requestor
and m.request_type <> ar.request_type
and m.time NOT between ar.start_time and ar.end_time
and m.component <> ar.component
order by m.m_id, ar.ar_id;

END //

DELIMITER ;
```

Population of Tables

- **MODEL**
  - **Total rows: 20266**

```
10 ● 	select * from MODEL;
11
12
13
14
      ⌄    21:10    2 errors found
```

Result Grid		Filter Rows:	Q Search		Edit:	Export/Import:	

m_id	time	requestor	role	component	request_type	violation_type	score
▶ 1	2016-08-10 06:15:00	User B	Business user	Table 2	Select	No authorization	FALSE
2	2016-08-10 06:29:00	User B	Business user	Table 2	Select	No authorization	FALSE
3	2016-08-10 06:43:00	User B	Business user	Table 2	Select	No authorization	FALSE
4	2016-08-10 06:58:00	User C	Administrator	Element 1	Select	No authorization	FALSE
5	2016-08-10 07:12:00	User A	Analyst	Element 2	Select	No authorization	TRUE
6	2016-08-10 07:27:00	User A	Analyst	Element 2	Select	No authorization	TRUE
7	2016-08-10 07:41:00	User A	Analyst	Element 2	Select	No authorization	TRUE
8	2016-08-10 07:55:00	User A	Analyst	Element 3	Append	No authorization	TRUE
9	2016-08-10 08:10:00	User A	Analyst	Element 1	Select	No authorization	FALSE
10	2016-08-10 08:24:00	User B	Business user	Table 2	Select	No authorization	FALSE

LEFT JOIN (Implementation)

Akshay

- For each row from MODEL table having score TRUE, fetch each row from ACCESS RULE table and insert ar_id, m_id into Comparison table.
  - **Row Count: 38080**

LEFT JOIN (Implementation)

Akshay

- Created a stored procedure insertComparison() for the same to insert records into Comparison table from both MODEL and ACCESS RULE table.

```
31    DELIMITER //
32
33 •  create procedure insertComparison()
34 ⊖ BEGIN
35
36    insert into Comparison (A_ar_id,M_m_id)
37    select ar_id, m_id
38    from ACCESSRULE, MODEL
39    where score = 'TRUE';
40
41 └ END //
42
43    DELIMITER ;
```

LEFT JOIN (Implementation)

- Fetched all mismatched rows (evidences of discrepancy) from MODEL table after comparison of pertinent attributes from both MODEL table and ACCESS RULE table.
  - **Mismatched count (Evidence of Discrepancy): 9860**

```sql
36   select m.m_id,c.A_ar_id,m.time,m.requestor,m.role,m.component,m.request_type,m.violation_type,m.score
37   from
38   MODEL m
39   left join Comparison c
40   on m.m_id = c.M_m_id
41   left join ACCESSRULE ar
42   on ar.ar_id = c.A_ar_id
43   where
44   m.role <> ar.role
45   and m.requestor <> ar.requestor
46   and m.request_type <> ar.request_type
47   and m.time NOT between ar.start_time and ar.end_time
48   and m.component <> ar.component
49   order by m.m_id, ar.ar_id;
```

27:49

Result Grid | Filter Rows: | Search | Export: | Fetch rows:

m_id	A_ar_id	time	requestor	role	component	request_type	violation_type	score
5	5	2016-08-10 07:12:00	User A	Analyst	Element 2	Select	No authorization	TRUE
5	9	2016-08-10 07:12:00	User A	Analyst	Element 2	Select	No authorization	TRUE
5	11	2016-08-10 07:12:00	User A	Analyst	Element 2	Select	No authorization	TRUE
5	13	2016-08-10 07:12:00	User A	Analyst	Element 2	Select	No authorization	TRUE
6	5	2016-08-10 07:27:00	User A	Analyst	Element 2	Select	No authorization	TRUE
6	9	2016-08-10 07:27:00	User A	Analyst	Element 2	Select	No authorization	TRUE
6	11	2016-08-10 07:27:00	User A	Analyst	Element 2	Select	No authorization	TRUE

LEFT JOIN (Implementation)
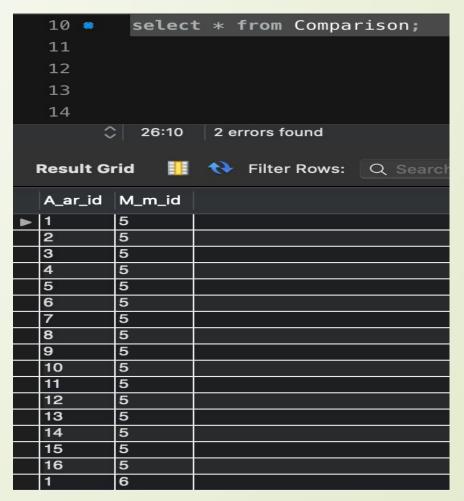
- Created a stored procedure i.e. getEvidence() to get all evidence of discrepancies happened during comparison between MODEL and ACCESS RULE table.
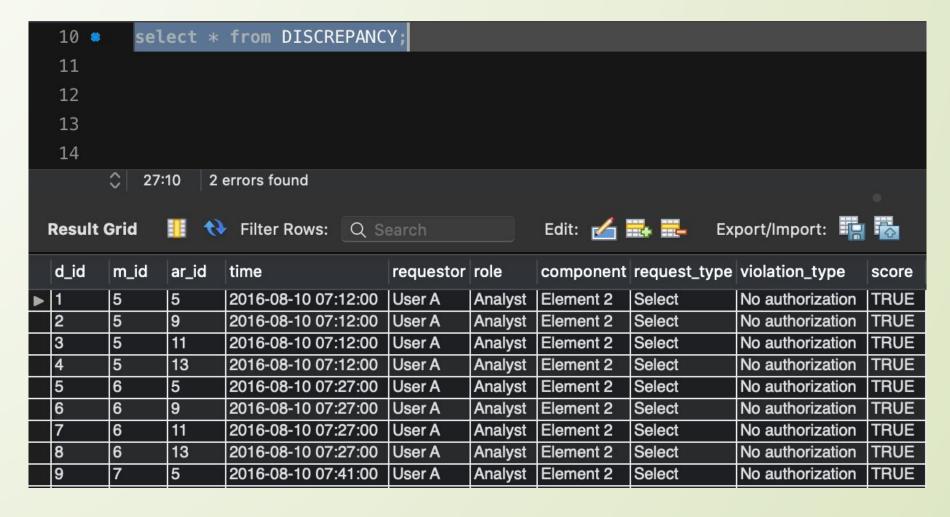
```
DELIMITER //

create procedure getEvidence()
BEGIN

call insertComparison();

insert into DISCREPANCY (m_id,ar_id,time,requestor,role,component,request_type,violation_type,score)
select m.m_id,c.A_ar_id,m.time,m.requestor,m.role,m.component,m.request_type,m.violation_type,m.score
from
MODEL m
left join Comparison c
on m.m_id = c.M_m_id
left join ACCESSRULE ar
on ar.ar_id = c.A_ar_id
where
m.role <> ar.role
and m.requestor <> ar.requestor
and m.request_type <> ar.request_type
and m.time NOT between ar.start_time and ar.end_time
and m.component <> ar.component
order by m.m_id, ar.ar_id;

END //

DELIMITER ;
```

Population of Tables

- **Comparison**
  - Total rows: 38080

Population of Tables

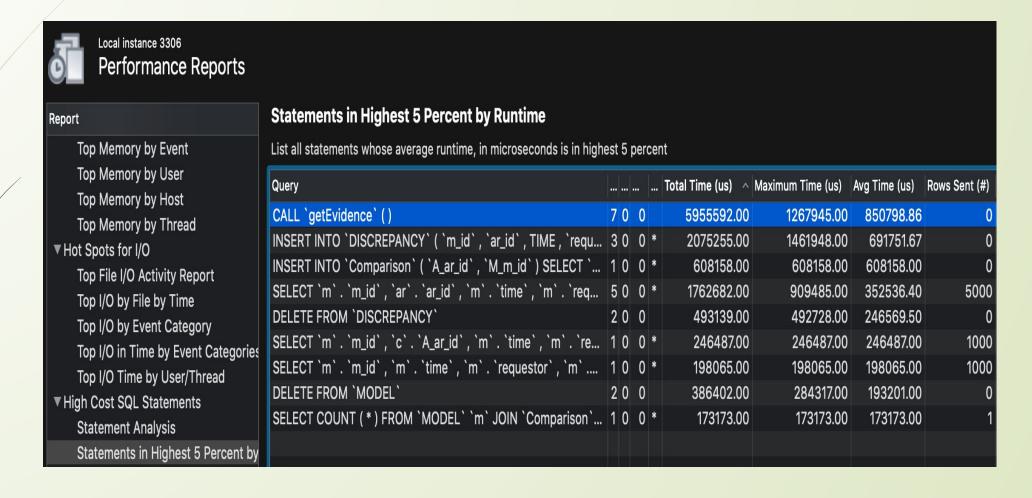- **DISCREPANCY**
  - Total rows: 9860

LEFT JOIN (Performance Report)
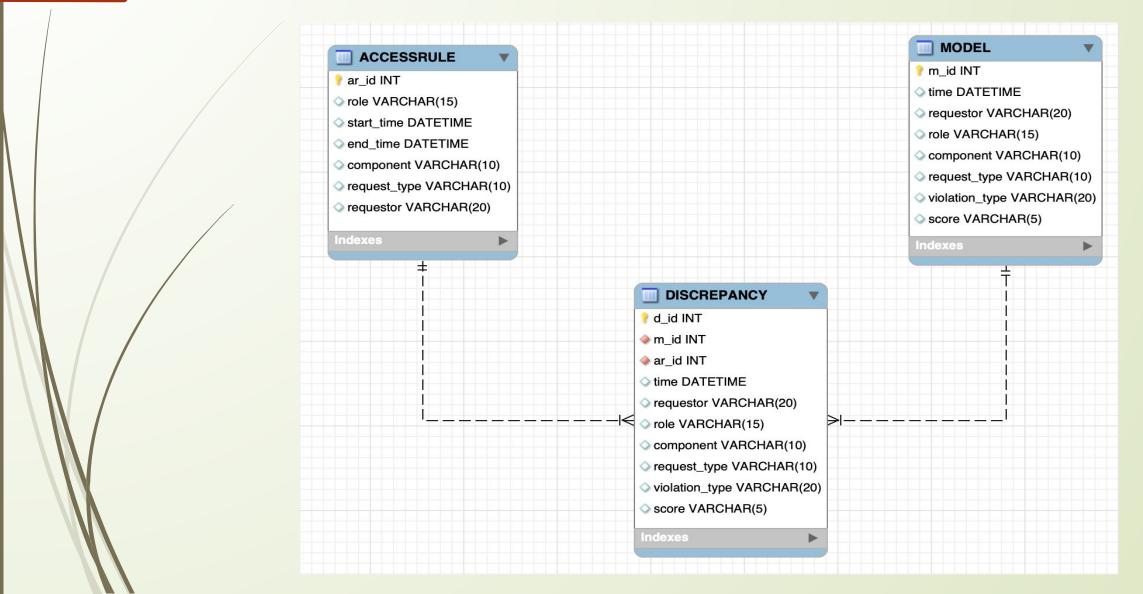
Akshay

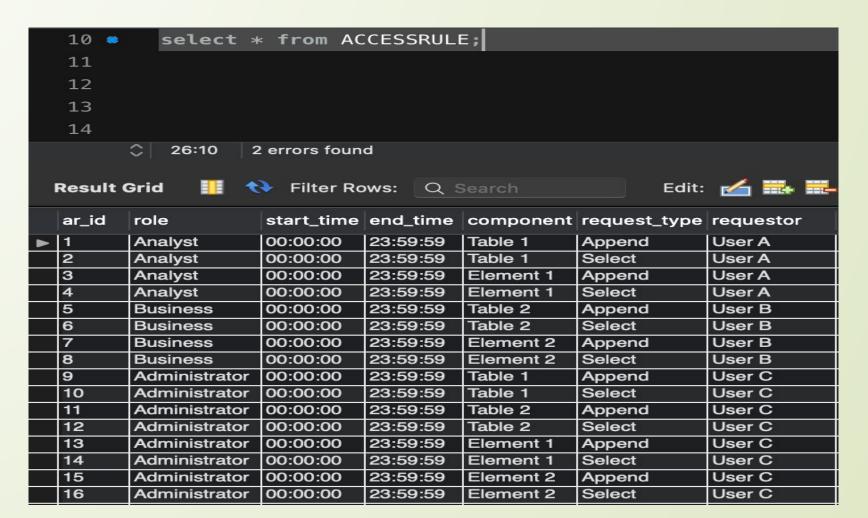**Total Time: 5955592.00 microsec → 5.95 seconds**

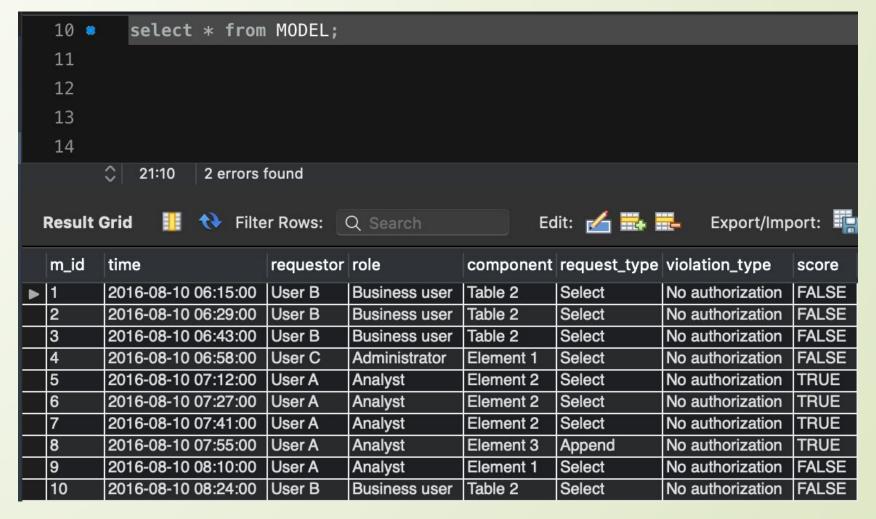# Cursors (Updated ERD)

Cursors (Pseudo Code)

- We have created a stored procedure called getEvidenceByCursor().

- Declared all variables which are required.

- Uses two cursors, for MODEL and ACCESS RULE table respectively.

- Uses IF loop to compare all pertinent attributes from MODEL and ACCESS RULE table.

- Based on comparison, mismatched rows between MODEL and ACCESS RULE will be inserted into DISCREPANCY table.

Population of Tables

- **ACCESSRULE**
  - Total rows: 16

Population of Tables

- **MODEL**
  - **Total rows: 20266**

Cursors (Implementation)

Anmol

```sql
use sync;
DELIMITER //
CREATE PROCEDURE getEvidenceByCursor()
BEGIN
            DECLARE done INT DEFAULT FALSE;
            DECLARE m_id INT;
            DECLARE ar_id INT;
            DECLARE time datetime;
            DECLARE start_time time;
            DECLARE end_time time;
            DECLARE m_requestor varchar(20);
            DECLARE ar_requestor varchar(20);
            DECLARE m_role varchar(15);
            DECLARE ar_role varchar(15);
            DECLARE m_component varchar(10);
            DECLARE ar_component varchar(10);
            DECLARE m_request_type varchar(10);
            DECLARE ar_request_type varchar(10);
            DECLARE violation_type varchar(30);
            DECLARE score varchar(5);

    DECLARE cur1 CURSOR FOR
        SELECT m_id, time, requestor, role, component, request_type, violation_type, score
        FROM MODEL WHERE score = 'TRUE';

    DECLARE cur2 CURSOR FOR
        SELECT ar_id,role,start_time,end_time,component,request_type,requestor
        FROM ACCESSRULE;
```
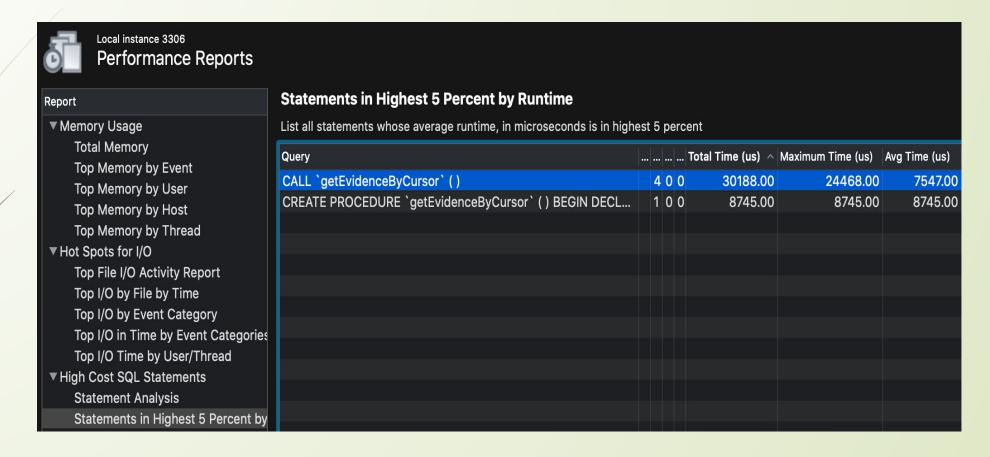
Cursors (Implementation)

Anmol

```sql
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur1;
OPEN cur2;


read_loop: LOOP
    FETCH cur1 INTO m_id, time, m_requestor, m_role, m_component, m_request_type, violation_type, score;
    FETCH cur2 INTO ar_id, ar_role, start_time,end_time,ar_component,ar_request_type,ar_requestor;

    IF done THEN
        LEAVE read_loop;
    END IF;
    IF (m_role <> ar_role AND (time NOT BETWEEN start_time AND end_time) AND m_component <> ar_component
        AND m_request_type <> ar_request_type AND m_requestor <> ar_requestor)
    THEN
        INSERT INTO DISCREPANCY VALUES (m_id, ar_id, time, requestor, role, component, request_type, violation_type, score);
    END IF;
END LOOP;

CLOSE cur1;
CLOSE cur2;
END //

DELIMITER ;

call getEvidenceByCursor();
```

Cursors (Performance Report)

Rashmi

NOTE: Unfortunately, procedure is not working properly with Cursor approach to perform comparison operation. (So Total Time in performance report will be different in working scenario)

Group Summary & Conclusion   **Akshay**

- We have populated the MODEL and ACCESS RULE tables with given data set.

- We have performed two approaches to do a comparison operation between pertinent attributes of MODEL and ACCESS RULE table i.e. LEFT JOIN, and CURSOR.

- With LEFT JOIN approach, we have got 9860 mismatched records (called 'evidence of discrepancy') between MODEL and ACCESS RULE table after comparison operation. (Total time: 5.95 seconds)

- Unfortunately, with CURSOR approach, we have not able to fetch the expected mismatched records between MODEL and ACCESS RULE table and unable to gather evidences in DISCREPANCY table.

Suggestions

- According to real time use of Cursors, the approach uses more network bandwidth which ultimately degrade the speed of the operations.

- Due to such poor performance, cursors are not recommended approach.

- We recommend the JOIN approach over CURSOR approach, to get overall good performance for the implemented system with much less execution time.

Thank You