MySQL: Views

- 12 in Murach text (MySQL)
- Create and use views in query-writing

Database Views

- A View is a virtual table (also called a **logical table**), consisting of the rows and columns specified in the SELECT statement in its CREATE VIEW statement.
- The view does not store any data itself; it refers back to the base table(s). Thus, the view always reflects the most current data in the database.

Benefits to Views

- Simplified queries
- Updatability
- Data security, limited access for specified users
- Design independence

CREATE VIEW statement

```
CREATE [OR REPLACE] VIEW view_name
        [(column_alias_1[, column_alias_2]...)]
AS
        select_statement
        [WITH [CASCADED | LOCAL ] CHECK OPTION]
;
```

```
-- create view
DROP VIEW last_training;
CREATE OR REPLACE VIEW last_training AS
    SELECT  Division, locations.Location_ID,
        MAX(STR_TO_DATE(Training_Date, "%m/%d/%Y")) AS Last_Training,
        DATEDIFF("2018-01-01", MAX(STR_TO_DATE(Training_Date, "%m/%d/%Y"))) AS Days_Since_Training
    FROM trainings
    JOIN locations ON trainings.location_ID = locations.location_ID
    GROUP BY Location_ID
;
```

```
14    -- use view in SELECT statement
15 •  SELECT * FROM last_training;
16 •  SELECT * FROM last_training WHERE Days_Since_Training > 500;
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Division	Location_ID	Last_Training	Days_Since_Training
CF	2453	2016-06-24	556
PK	2468	2016-06-17	563
FD	2505	2016-07-02	548
KJ	2534	2016-03-18	654
AE	2554	2016-02-18	683
SM	2612	2016-05-14	597
KJ	2639	2016-02-13	688

View vs Table

```
19  -- create table instead of view
20  DROP TABLE IF EXISTS last_training_table;
21  CREATE TABLE last_training_table AS
22      SELECT  Division, locations.Location_ID,
23              MAX(STR_TO_DATE(Training_Date, "%m/%d/%Y")) AS Last_Training,
24              DATEDIFF("2018-01-01", MAX(STR_TO_DATE(Training_Date, "%m/%d/%Y"))) AS Days_Since_Training
25      FROM trainings
26      JOIN locations ON trainings.location_ID = locations.location_ID
27      GROUP BY Location_ID
28  ;
29
30  SELECT * FROM last_training_table WHERE Days_Since_Training > 500;
```

Division	Location_ID	Last_Training	Days_Since_Training
CF	2453	2016-06-24	556
PK	2468	2016-06-17	563
FD	2505	2016-07-02	548
KJ	2534	2016-03-18	654
AE	2554	2016-02-18	683
SM	2612	2016-05-14	597
KJ	2639	2016-02-13	688

```
14  -- use view in SELECT statement
15  SELECT * FROM last_training;
16  SELECT * FROM last_training WHERE Days_Since_Training > 500;
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Division	Location_ID	Last_Training	Days_Since_Training
CF	2453	2016-06-24	556
PK	2468	2016-06-17	563
FD	2505	2016-07-02	548
KJ	2534	2016-03-18	654
AE	2554	2016-02-18	683
SM	2612	2016-05-14	597
KJ	2639	2016-02-13	688

View vs Table

trainings (underlying table)
BEFORE inserting new record

Location_ID	Training_Date	Training_Location
2453	8/17/2014	Onsite
2453	12/4/2014	Onsite
2453	6/20/2015	Onsite
2453	8/16/2015	Onsite
2453	11/7/2015	Offsite
2453	6/24/2016	Onsite

```
32  -- note virtual nature of view vs. table
33  SELECT * FROM trainings
34  WHERE Location_ID = 2453
35  ORDER BY STR_TO_DATE(Training_Date, "%m/%d/%Y");
36
37  INSERT INTO trainings VALUES
38  (2453, "08/01/2016", "Offsite");
39
40  SELECT * FROM last_training_table WHERE Days_Since_Training > 500;
41  SELECT * FROM last_training WHERE Days_Since_Training > 500;
```

trainings (underlying table)
AFTER inserting new record

Location_ID	Training_Date	Training_Location
2453	8/17/2014	Onsite
2453	12/4/2014	Onsite
2453	6/20/2015	Onsite
2453	8/16/2015	Onsite
2453	11/7/2015	Offsite
2453	6/24/2016	Onsite
2453	8/1/2016	Offsite

last_training_table (derived table)

Division	Location_ID	Last_Training	Days_Since_Training
CF	2453	2016-06-24	556
PK	2468	2016-06-17	563
FD	2505	2016-07-02	548
KJ	2534	2016-03-18	654
AE	2554	2016-02-18	683
SM	2612	2016-05-14	597
KJ	2639	2016-02-13	688

last_training (view)

Division	Location_ID	Last_Training	Days_Since_Training
CF	2453	2016-08-01	518
PK	2468	2016-06-17	563
FD	2505	2016-07-02	548
KJ	2534	2016-03-18	654
AE	2554	2016-02-18	683
SM	2612	2016-05-14	597
KJ	2639	2016-02-13	688

Example: Safety Training View

- Create query that shows the location in each division which has gone the longest time since a safety training.

Division	Location_ID	Last_Training	Days_Since_Training
AE	2554	2016-02-18	683
CF	4083	2016-01-12	720
FD	3435	2016-02-05	696
KJ	2910	2016-02-07	694
PK	4800	2016-01-23	709
SM	4264	2016-01-23	709

Two queries that accomplish this query, using last_training VIEW

```sql
-- correlated subquery
SELECT Division, Location_ID, Last_Training, Days_Since_Training
FROM last_training lt
WHERE Days_Since_Training = (   SELECT MAX(Days_Since_Training)
                                FROM last_training
                                WHERE Division = lt.Division)
ORDER BY Division
;
```

```sql
-- join
SELECT lt_detail.Division, Location_ID, Last_Training, Days_Since_Training
FROM last_training lt_detail
    JOIN (  SELECT Division, MAX(Days_Since_Training) AS Days
            FROM last_training
            GROUP BY Division) AS lt_max
        ON lt_detail.Division = lt_max.Division
            AND lt_detail.Days_Since_Training = lt_max.Days
ORDER BY lt_detail.Division
;
```

MySQL: Views

- 7 in Lemahieu text (note SQL), 12 in Murach text (MySQL)
- Create and use views
  - Read-only views
  - Views for updating and deleting data

Updatable Views

Requirements for creating updatable views

- The select list can't include a DISTINCT clause.
- The select list can't include aggregate functions.
- The SELECT statement can't include a GROUP BY or HAVING clause.
- The view can't include the UNION operator.

Example 1: last_training view

```
12
13  CREATE OR REPLACE VIEW last_training AS
14  SELECT Division, locations.Location_ID,
15  MAX(STR_TO_DATE(Training_Date, "%m/%d/%Y")) AS Last_Training,
16  DATEDIFF("2018-01-01", MAX(STR_TO_DATE(Training_Date, "%m/%d/%Y"))) AS Days_Since_Training
17  FROM trainings
18  JOIN locations ON trainings.location_ID = locations.location_ID
19  GROUP BY Location_ID;
20
21
```

Example 1: last_training view

```
89 • SELECT * FROM last_training;
90
91 • UPDATE last_training SET Days_Since_Training = 181
92   WHERE Location_ID=2408;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Division	Location_ID	Last_Training	Days_Since_Training
FD	2440	2017-12-27	5
CF	2453	2016-08-01	518
SM	2464	2016-09-04	484
PK	2468	2016-06-17	563
CF	2493	2016-10-06	452
FD	2505	2016-07-02	548
AE	2528	2017-11-04	58
KJ	2534	2016-03-18	654
FD	2551	2017-08-04	150
AE	2554	2016-02-18	683
CF	2572	2017-06-02	213
KJ	2594	2017-04-10	266
SM	2612	2016-05-14	597
AE	2616	2017-12-25	7
FD	2623	2017-11-21	41
KJ	2639	2016-02-13	688
SM	2643	2016-12-04	393
AE	2656	2016-03-28	644
KJ	2664	2017-02-28	307
FD	2671	2017-09-10	113
AE	2689	2016-07-17	533
KJ	2707	2017-12-23	9
KJ	2709	2017-04-24	252
CF	2728	2016-11-18	409

Result Grid | Form Editor | Field Types | Query Stats | Execution Plan

last_training 1 | Read Only

Output

Action Output

#	Time	Action	Message
22	12:33:16	SELECT * FROM last_training	179 row(s) returned
23	12:33:37	UPDATE last_training SET Days_Since_Training = 181 WHERE Location_ID=2408	Error Code: 1288. The target table last_training of the UPDATE is not updatable

Example 1: locations_headcount view

```
102 • CREATE OR REPLACE VIEW locations_headcount AS
103   SELECT locations.Location_ID, Headcount
104   FROM  locations;
105
106 • SELECT * FROM locations_headcount;
107
108 • UPDATE locations_headcount SET Headcount = 555
109   WHERE Location_ID=2408;
110
111 • UPDATE locations_headcount SET Headcount = 554
112   WHERE Location_ID=2408;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Location_ID	Headcount
2408	554
2415	181
2417	327
2440	273
2453	228
2464	183
2468	365

Result Grid | Filter Rows:

Location_ID	Headcount
2408	555
2415	181
2417	327
2440	273
2453	228
2464	183
2468	365
2493	430

130 | 15:39:23 | UPDATE reimb_summary SET reimbursement_amount = 750 WHERE employee = ... | 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0