Local & Cascaded Check

This lesson looks at the two types of rule checking that can be used with the CHECK option clause.

LOCAL AND CASCADED CHECK

Local and cascaded check clauses are used to determine the scope of rule testing when a view is created based on another view. To summarize, **Local** check option restricts the rule checking to only the view being defined whereas the **Cascaded** check option checks the rules of all underlying views. In the absence of these keywords, cascaded check is used as default.

Syntax

> CREATE [OR REPLACE] VIEW **view_name** AS
>
> **select_statement**
>
> WITH [LOCAL | CASCADED] CHECK OPTION;

> Connect to the terminal below by clicking in the widget. Once connected, the command line prompt will show up. Enter or copy and paste the command **./DataJek/Lessons/44lesson.sh** and wait for the MySQL prompt to start-up.

```
-- The lesson queries are reproduced below for convenient copy/paste into the terminal.

-- CASCADED CHECK
-- Query 1
CREATE VIEW ActorsView1 AS
SELECT *
```

```
SELECT *
FROM Actors
WHERE TIMESTAMPDIFF(YEAR, DoB, CURDATE()) > 40;

-- Query 2
INSERT INTO ActorsView1
VALUES (DEFAULT, 'Young', 'Actress', '2000-01-01', 'Female', 'Single', 000.00);

-- Query 3
CREATE OR REPLACE VIEW ActorsView2 AS
SELECT *
FROM ActorsView1
WITH CASCADED CHECK OPTION;

-- Query 4
INSERT INTO ActorsView2
VALUES (DEFAULT, 'Young', 'Actor', '2000-01-01', 'Male', 'Single', DEFAULT);

-- Query 5
CREATE OR REPLACE VIEW ActorsView3 AS
SELECT *
FROM ActorsView2
WHERE TIMESTAMPDIFF(YEAR, DoB, CURDATE()) < 50;

-- Query 6
INSERT INTO ActorsView3
VALUES (DEFAULT, 'Young', 'Actor', '2000-01-01', 'Male', 'Single', DEFAULT);

-- Query 7
INSERT INTO ActorsView3
VALUES (DEFAULT, 'Old', 'Actor', '1960-01-01', 'Male', 'Single', DEFAULT);

-- LOCAL CHECK
-- Query 8
ALTER VIEW ActorsView2 AS
SELECT *
FROM ActorsView1
WITH LOCAL CHECK OPTION;

-- Query 9
INSERT INTO ActorsView2
VALUES (DEFAULT, 'Young', 'Actor', '2000-01-01', 'Male', 'Single', DEFAULT);

-- Query 10
INSERT INTO ActorsView3
VALUES (DEFAULT, 'Young', 'Actor', '2000-01-01', 'Male', 'Single', DEFAULT);
```

Terminal

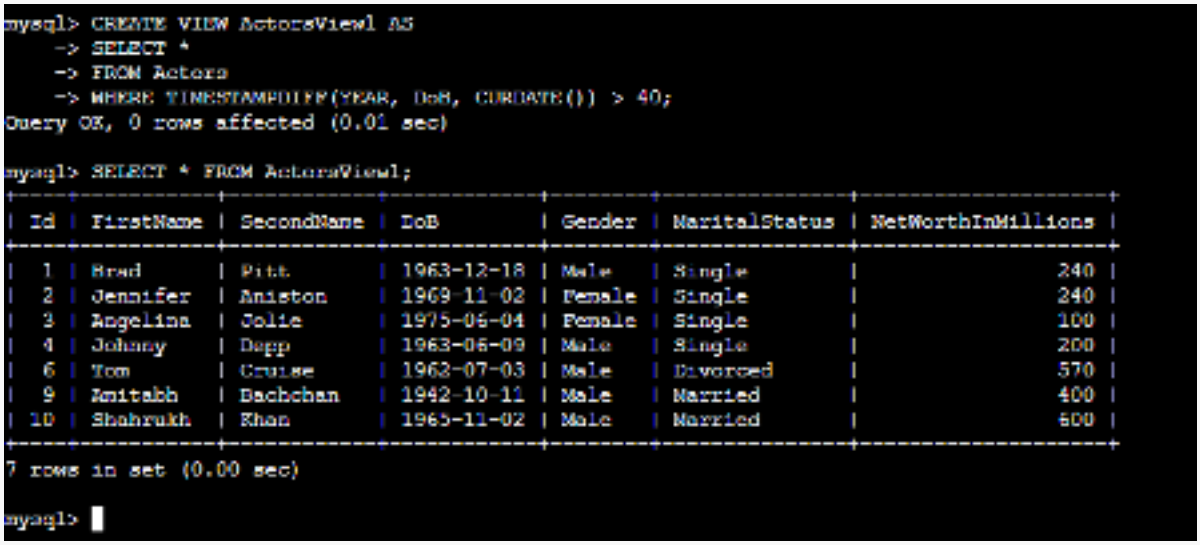
Cascaded Check

1. We will start by creating a view **ActorsView1** which shows all actors who are older than 40.

```
CREATE VIEW ActorsView1 AS
```

```
SELECT *
FROM Actors
WHERE TIMESTAMPDIFF(YEAR, DoB, CURDATE()) > 40;
```

Seven rows from the **Actors** table satisfy the WHERE clause.

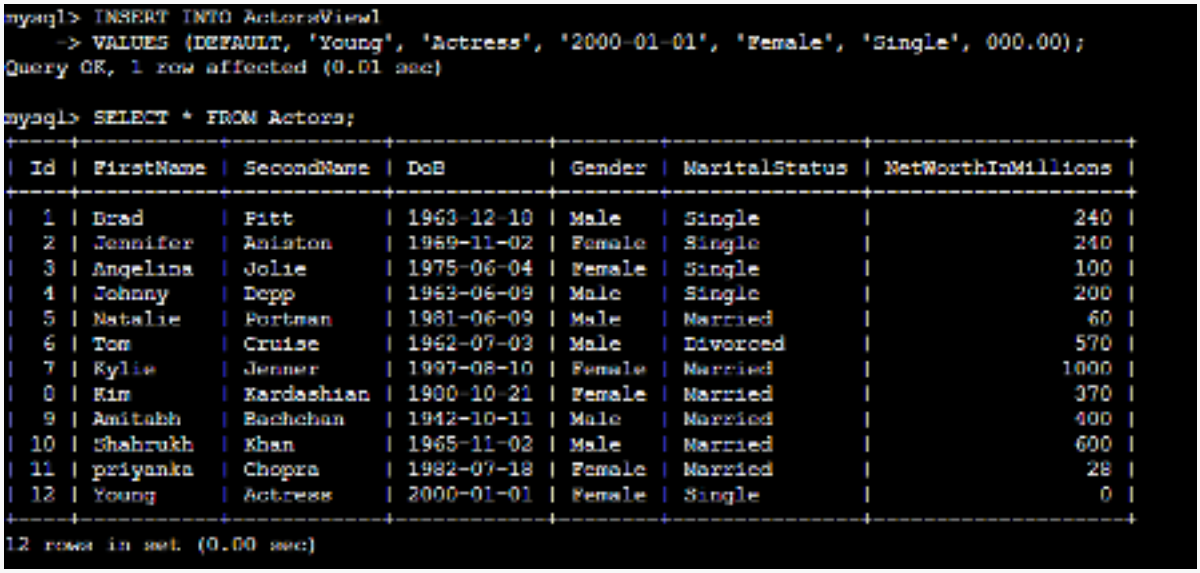


```
mysql> CREATE VIEW ActorsView1 AS
    -> SELECT *
    -> FROM Actors
    -> WHERE TIMESTAMPDIFF(YEAR, DoB, CURDATE()) > 40;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT * FROM ActorsView1;
+----+-----------+------------+------------+--------+---------------+--------------------+
| Id | FirstName | SecondName | DoB        | Gender | MaritalStatus | NetWorthInMillions |
+----+-----------+------------+------------+--------+---------------+--------------------+
|  1 | Brad      | Pitt       | 1963-12-18 | Male   | Single        |                240 |
|  2 | Jennifer  | Aniston    | 1969-11-02 | Female | Single        |                240 |
|  3 | Angelina  | Jolie      | 1975-06-04 | Female | Single        |                100 |
|  4 | Johnny    | Depp       | 1963-06-09 | Male   | Single        |                200 |
|  6 | Tom       | Cruise     | 1962-07-03 | Male   | Divorced      |                570 |
|  9 | Amitabh   | Bachchan   | 1942-10-11 | Male   | Married       |                400 |
| 10 | Shahrukh  | Khan       | 1965-11-02 | Male   | Married       |                600 |
+----+-----------+------------+------------+--------+---------------+--------------------+
7 rows in set (0.00 sec)

mysql>
```

In the absence of the WITH CHECK OPTION clause there is no restriction on updates through **ActorsView1**. We can insert a 20-year-old actor to the **Actors** table using this view as follows:

```
INSERT INTO ActorsView1
VALUES (DEFAULT, 'Young', 'Actress', '2000-01-01', 'Female', 'Sin
gle', 000.00);
```

The record is inserted in the table even though it does not satisfy the condition of the view (age > 40 years).



```
mysql> INSERT INTO ActorsView1
    -> VALUES (DEFAULT, 'Young', 'Actress', '2000-01-01', 'Female', 'Single', 000.00);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Actors;
+----+-----------+------------+------------+--------+---------------+--------------------+
| Id | FirstName | SecondName | DoB        | Gender | MaritalStatus | NetWorthInMillions |
+----+-----------+------------+------------+--------+---------------+--------------------+
|  1 | Brad      | Pitt       | 1963-12-18 | Male   | Single        |                240 |
|  2 | Jennifer  | Aniston    | 1969-11-02 | Female | Single        |                240 |
|  3 | Angelina  | Jolie      | 1975-06-04 | Female | Single        |                100 |
|  4 | Johnny    | Depp       | 1963-06-09 | Male   | Single        |                200 |
|  5 | Natalie   | Portman    | 1981-06-09 | Male   | Married       |                 60 |
|  6 | Tom       | Cruise     | 1962-07-03 | Male   | Divorced      |                570 |
|  7 | Kylie     | Jenner     | 1997-08-10 | Female | Married       |               1000 |
|  8 | Kim       | Kardashian | 1900-10-21 | Female | Married       |                370 |
|  9 | Amitabh   | Bachchan   | 1942-10-11 | Male   | Married       |                400 |
| 10 | Shahrukh  | Khan       | 1965-11-02 | Male   | Married       |                600 |
| 11 | priyanka  | Chopra     | 1982-07-18 | Female | Married       |                 28 |
| 12 | Young     | Actress    | 2000-01-01 | Female | Single        |                  0 |
+----+-----------+------------+------------+--------+---------------+--------------------+
12 rows in set (0.00 sec)
```

The record appears in the table but not in the view.

```
mysql> SELECT * FROM ActorsView1;
+----+-----------+------------+------------+--------+---------------+---------------------+
| Id | FirstName | SecondName | DoB        | Gender | MaritalStatus | NetWorthInMillions |
+----+-----------+------------+------------+--------+---------------+---------------------+
|  1 | Brad      | Pitt       | 1963-12-18 | Male   | Single        |                 240 |
|  2 | Jennifer  | Aniston    | 1969-11-02 | Female | Single        |                 240 |
|  3 | Angelina  | Jolie      | 1975-06-04 | Female | Single        |                 100 |
|  4 | Johnny    | Depp       | 1963-06-09 | Male   | Single        |                 200 |
|  6 | Tom       | Cruise     | 1962-07-03 | Male   | Divorced      |                 570 |
|  9 | Amitabh   | Bachchan   | 1942-10-11 | Male   | Married       |                 400 |
| 10 | Shahrukh  | Khan       | 1965-11-02 | Male   | Married       |                 600 |
+----+-----------+------------+------------+--------+---------------+---------------------+
7 rows in set (0.00 sec)
```

This is a disparity that can be handled using the **WITH CHECK OPTION** clause as discussed at length in the previous lesson.

2. Next, create a view **ActorsView2** based on **ActorsView1** as follows:

```
CREATE OR REPLACE VIEW ActorsView2 AS
SELECT *
FROM ActorsView1
WITH CASCADED CHECK OPTION;
```

Since **ActorsView2** is based on **ActorsView1**, it also has seven rows. The view has a **CASCADED** check option which means that insert or update through **ActorsView2** should not only be compatible with this view but also the underlying view, which is, **ActorsView1**. To see how it works, insert an actor to the **Actors** table using **ActorsView2** whose age is 20 years.

```
INSERT INTO ActorsView2
VALUES (DEFAULT, 'Young', 'Actor', '2000-01-01', 'Male', 'Single'
, DEFAULT);
```

```
mysql> CREATE OR REPLACE VIEW ActorsView2 AS
    -> SELECT *
    -> FROM ActorsView1
    -> WITH CASCADED CHECK OPTION;
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO ActorsView2
    -> VALUES (DEFAULT, 'Young', 'Actor', '2000-01-01', 'Male', 'Single', DEFAULT);
ERROR 1369 (HY000): CHECK OPTION failed 'MovieIndustry.ActorsView2'
mysql>
```

We encounter an error message, **CHECK OPTION failed 'MovieIndustry.ActorsView2'**. The record is not inserted into the **Actors** table even though **ActorsView2** did not impose any age restrictions. This is because the **CASCADED CHECK OPTION** clause

also tests the rules of the underlying view, **ActorsView1**. Since **ActorsView1** only allowed actors who are older than 40 years and

ActorsView2 is based on **ActorsView1** so we were unable to add a 20 year old actor.

3. Now we are going to demonstrate the scope of rule testing of the **CASCADED** check option. For this purpose, create a view **ActorsView3** based on **ActorsView2** which should only display actors who are younger than 50.

```
CREATE OR REPLACE VIEW ActorsView3 AS
SELECT *
FROM ActorsView2
WHERE TIMESTAMPDIFF(YEAR, DoB, CURDATE()) < 50;
```

```
mysql> CREATE OR REPLACE VIEW ActorsView3 AS
    -> SELECT *
    -> FROM ActorsView2
    -> WHERE TIMESTAMPDIFF(YEAR, DoB, CURDATE()) < 50;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT * FROM ActorsView3;
+----+-----------+------------+------------+--------+---------------+--------------------+
| Id | FirstName | SecondName | DoB        | Gender | MaritalStatus | NetWorthInMillions |
+----+-----------+------------+------------+--------+---------------+--------------------+
|  3 | Angelina  | Jolie      | 1975-06-04 | Female | Single        |                100 |
+----+-----------+------------+------------+--------+---------------+--------------------+
1 row in set (0.00 sec)
```

There is only one row in this view. We should be able to insert a 20-year-old actor through this view as it satisfies the age < 50 rule. Try the following query:

```
INSERT INTO ActorsView3
VALUES (DEFAULT, 'Young', 'Actor', '2000-01-01', 'Male', 'Single'
, DEFAULT);
```

```
mysql> INSERT INTO ActorsView3
    -> VALUES (DEFAULT, 'Young', 'Actor', '2000-01-01', 'Male', 'Single', DEFAULT);
ERROR 1369 (HY000): CHECK OPTION failed 'MovieIndustry.ActorsView3'
mysql>
```

When executed, we encounter the error, **CHECK OPTION failed 'MovieIndustry.ActorsView3'**. This is because the **CASCADED** check option checks the rules of all underlying views before an update is allowed. **ActorsView3** is based on **ActorsView2** and **ActorsView2** is based on **ActorsView1** which only allows actors older than 40 so we were unable to add a 20-year-old actor through **ActorsView3.**

4. Let's see if we can insert a 60 year old actor with this view. Execute

4. Let's see if we can insert a 60-year-old actor with this view. Execute the query given below:

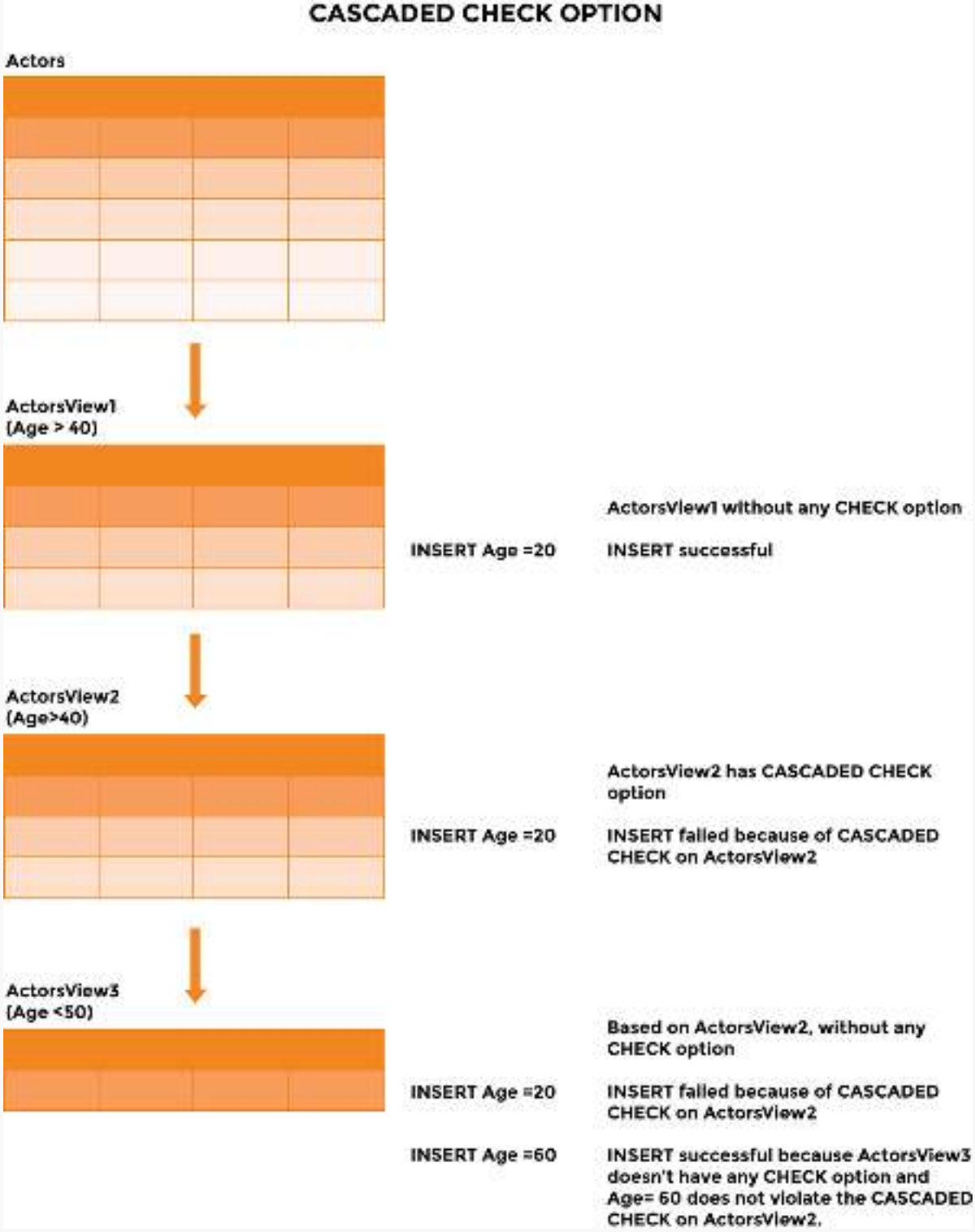
```
INSERT INTO ActorsView3
VALUES (DEFAULT, 'Old', 'Actor', '1960-01-01', 'Male', 'Single',
DEFAULT);
```

```
mysql> INSERT INTO ActorsView3
    -> VALUES (DEFAULT, 'Old', 'Actor', '1960-01-01', 'Male', 'Single', DEFAULT);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM ActorsView3;
+----+-----------+------------+------------+--------+---------------+--------------------+
| Id | FirstName | SecondName | DoB        | Gender | MaritalStatus | NetWorthInMillions |
+----+-----------+------------+------------+--------+---------------+--------------------+
|  3 | Angelina  | Jolie      | 1975-06-04 | Female | Single        |                100 |
+----+-----------+------------+------------+--------+---------------+--------------------+
1 row in set (0.00 sec)
```

Insert operation is successful even though an actor whose age is more than 50 years does not comply with the age restriction of **ActorsView3**. Since we did not mention the **WITH CHECK OPTION** clause when creating **ActorsView3**, the above insert operation was successful. Here the rules for **ActorsView2** and **ActorsView1** were checked because of the **CASCADED** check option and the insert was made as the row conforms with the rules of the underlying view (age should be more than 40 years).

Note that if **ActorsView3** was created using the **WITH CHECK OPTION** clause then the above insert operation would fail.

The following image gives a pictorial explanation of the effects of using cascaded check option:



Local Check

5. To limit the scope of rule checking let's redefine **ActorsView2** with a **LOCAL** check option as follows:

```
ALTER VIEW ActorsView2 AS
SELECT *
FROM ActorsView1
WITH LOCAL CHECK OPTION;
```

To recap, this view is based on **ActorsView1** which shows actors who

are older than 40 as shown:

```
mysql> ALTER VIEW ActorsView2 AS
    -> SELECT *
    -> FROM ActorsView1
    -> WITH LOCAL CHECK OPTION;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT * FROM ActorsView2;
+----+-----------+------------+------------+--------+---------------+--------------------+
| Id | FirstName | SecondName | DoB        | Gender | MaritalStatus | NetWorthInMillions |
+----+-----------+------------+------------+--------+---------------+--------------------+
|  1 | Brad      | Pitt       | 1963-12-18 | Male   | Single        |                240 |
|  2 | Jennifer  | Aniston    | 1969-11-02 | Female | Single        |                240 |
|  3 | Angelina  | Jolie      | 1975-06-04 | Female | Single        |                100 |
|  4 | Johnny    | Depp       | 1963-06-09 | Male   | Single        |                200 |
|  6 | Tom       | Cruise     | 1962-07-03 | Male   | Divorced      |                570 |
|  9 | Amitabh   | Bachchan   | 1942-10-11 | Male   | Married       |                400 |
| 10 | Shahrukh  | Khan       | 1965-11-02 | Male   | Married       |                600 |
+----+-----------+------------+------------+--------+---------------+--------------------+
7 rows in set (0.00 sec)
```

Since **ActorsView2** does not specify any age criterion, we can try inserting a 20-year-old actor using this view:

```
INSERT INTO ActorsView2
VALUES (DEFAULT, 'Young', 'Actor', '2000-01-01', 'Male', 'Single'
, DEFAULT);
```

```
mysql> INSERT INTO ActorsView2
    -> VALUES (DEFAULT, 'Young', 'Actor', '2000-01-01', 'Male', 'Single', DEFAULT);
Query OK, 1 row affected (0.01 sec)

mysql>
```

The row is successfully inserted. **LOCAL** check option in **ActorsView2** means that insert operation should only conform to the age restriction of **ActorsView2** (none in our case). When we used the **CASCADED** check in step 2, we got an error message when the above query was executed, because the rule of the underlying **ActorsView1** was also checked.

However, the row just inserted into the **Actors** table is not visible through **ActorsView2** because this view only shows actors who are older than 40.

```
mysql> INSERT INTO ActorsView2
    -> VALUES (DEFAULT, 'Young', 'Actor', '2000-01-01', 'Male', 'Single', DEFAULT);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM ActorsView2;
+----+-----------+------------+------------+--------+---------------+--------------------+
| Id | FirstName | SecondName | DoB        | Gender | MaritalStatus | NetWorthInMillions |
+----+-----------+------------+------------+--------+---------------+--------------------+
|  1 | Brad      | Pitt       | 1963-12-18 | Male   | Single        |                240 |
|  2 | Jennifer  | Aniston    | 1969-11-02 | Female | Single        |                240 |
|  3 | Angelina  | Jolie      | 1975-06-04 | Female | Single        |                100 |
|  4 | Johnny    | Depp       | 1963-06-09 | Male   | Single        |                200 |
|  6 | Tom       | Cruise     | 1962-07-03 | Male   | Divorced      |                570 |
|  9 | Amitabh   | Bachchan   | 1942-10-11 | Male   | Married       |                400 |
| 10 | Shahrukh  | Khan       | 1965-11-02 | Male   | Married       |                600 |
+----+-----------+------------+------------+--------+---------------+--------------------+
```

```
7 rows in set (0.00 sec)
```

6. Now that we have changed the scope of the check option for **ActorsView2**, we can see the effects on **ActorsView3** as well (**ActorsView3** is based on **ActorsView2** and specifies a rule, age < 50). We can now insert a 20-year-old actor using **ActorsView3** with the following query:

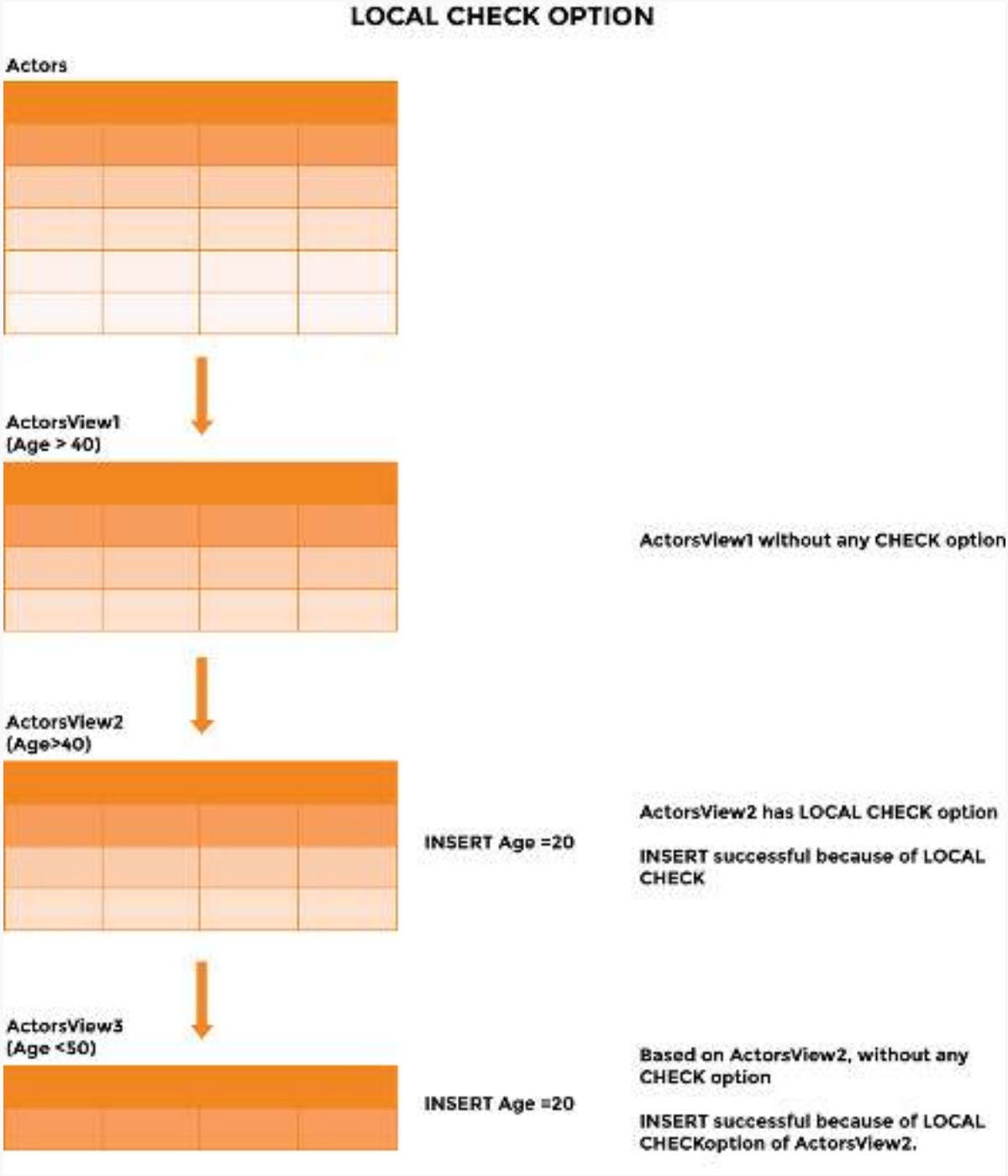
```
INSERT INTO ActorsView3
VALUES (DEFAULT, 'Young', 'Actor', '2000-01-01', 'Male', 'Single'
, DEFAULT);
```

Because of the **LOCAL** check, the insert operation is successful. Previously we had encountered an error in step 3 with the same query. At that time **ActorsView2** had a **CASCADED** check option and MYSQL checked the age restriction of underlying **ActorsView1**. Now **ActorsView2** has a **LOCAL** check option. Hence MYSQL inserts the record without checking the rule of **ActorsView1** (age>40).

```
mysql> INSERT INTO ActorsView3
    -> VALUES (DEFAULT, 'Young', 'Actor', '2000-01-01', 'Male', 'Single', DEFAULT);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM ActorsView3;
+----+-----------+------------+------------+--------+---------------+--------------------+
| Id | FirstName | SecondName | DoB        | Gender | MaritalStatus | NetWorthInMillions |
+----+-----------+------------+------------+--------+---------------+--------------------+
|  3 | Angelina  | Jolie      | 1975-06-04 | Female | Single        |                100 |
+----+-----------+------------+------------+--------+---------------+--------------------+
1 row in set (0.00 sec)

mysql>
```

As can be seen from the image above, the newly inserted row does not appear in the view because the actor we just inserted is 20-years-old while this view only shows actors who are older than 40 but less than 50 years old.

The following image shows the effects of using local check option instead of cascaded check:


LOCAL CHECK OPTION
Actors
ActorsView1
(Age > 40)
ActorsView1 without any CHECK option
ActorsView2
(Age>40)
INSERT Age =20
ActorsView2 has LOCAL CHECK option
INSERT successful because of LOCAL CHECK
ActorsView3
(Age <50)
INSERT Age =20
Based on ActorsView2, without any CHECK option
INSERT successful because of LOCAL CHECKoption of ActorsView2.