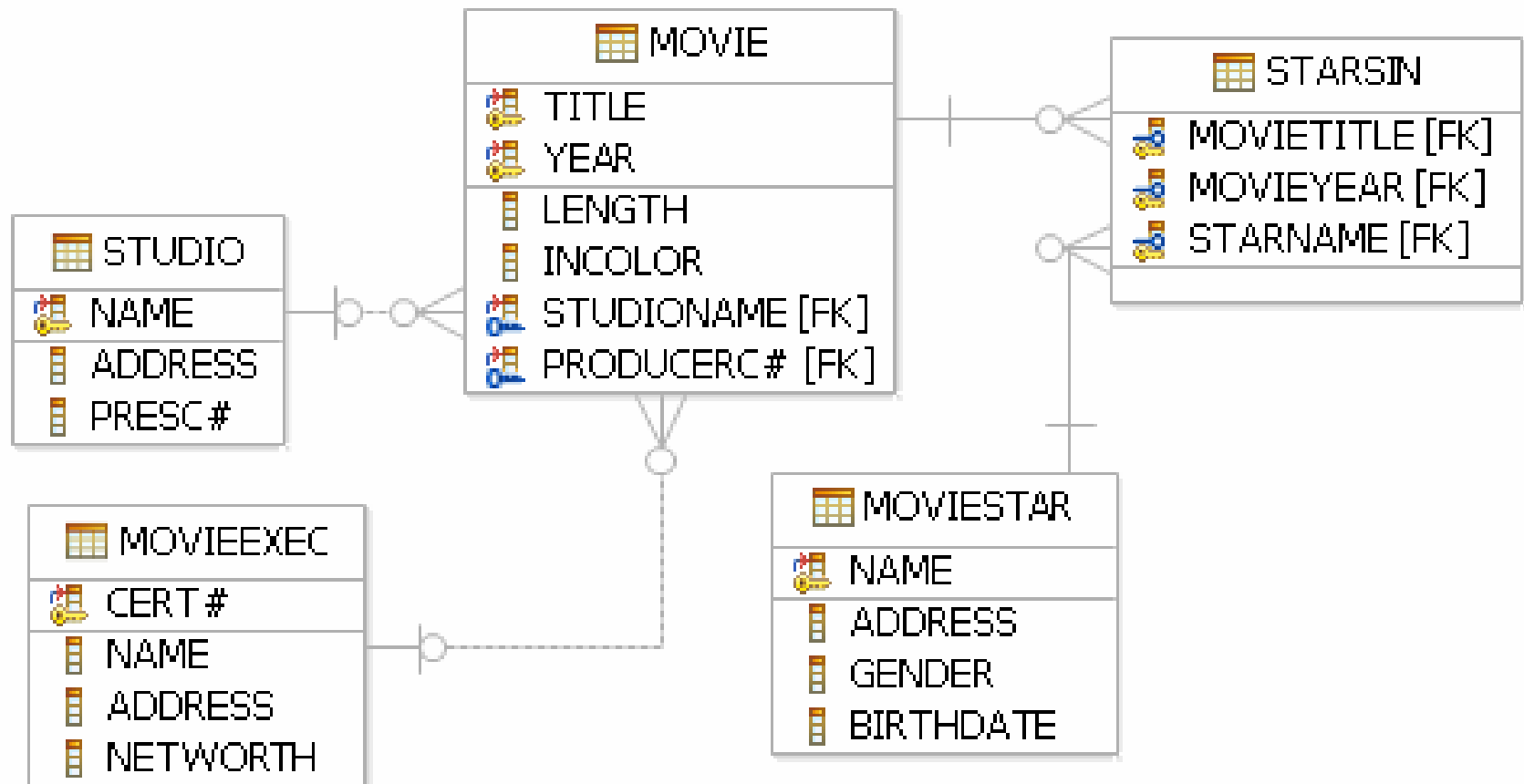




Тригери

База от данни - Movies



Тригери

- ▶ Тригерите са обекти в базите от данни, които се различават от другите ограничения по това, че се изпълняват при настъпването на дадено събитие, в следствие на което ако поставеното условие в тригера е удовлетворено се изпълнява определено действие.
- ▶ По тази причина, тригерите често са наричани и правила от тип “събитие-условие-действие”.
- ▶ Събитията, които могат да задействат тригер са INSERT, UPDATE и DELETE.
- ▶ Тригерите се обекти в базата от данни, които се дефинират за конкретна таблица или изглед и веднъж създадени се задействат автоматично.

Тригери

- ▶ Действията INSERT, UPDATE и DELETE за съответната таблица, водят до задействане на тригера и изпълняване на съответното действие дефинирано в тригера.
- ▶ Тригерите се задействат автоматично при споменатите команди.
- ▶ Те не могат да бъдат задействани по друг начин, освен посредством модификация на таблицата или изгледа за които са дефинирани.

Синтаксис

- Обобщеният синтаксис за създаване на тригер е следният:

```
CREATE TRIGGER <име_на_тригер>
{AFTER | INSTEAD OF | BEFORE} {INSERT |
UPDATE | DELETE} ON my_table
REFERENCING NEW ROW | TABLE AS N OLD ROW
| TABLE AS O
FOR EACH ROW | STATEMENT
WHEN (<условие>)
        <действие>;
```

Обяснение

- ▶ При създаването на един тригер, може да бъде указано, кога да се изпълни действието на тригера: преди настъпване на събитието (BEFORE), след настъпване на събитието (AFTER) или вместо самото действие (INSTEAD OF).
- ▶ Също така в дефиницията на тригера се указва и кое е събитието, което задейства тригера (INSERT, UPDATE или DELETE).
- ▶ Възможно е тригерът да бъде задействан и при трите действия, за които да се изпълнява едно и също действие. В този случай при дефиниране на тригера се указват и трите команди.

Обяснение

- ▶ Ако събитието което задейства тригерът е UPDATE, освен името на таблицата (или изгледа), може да бъде указана и конкретна колона от таблицата, чиято промяна да води до задействане на тригера.
- ▶ В дефиницията на тригера в клаузата REFERENCING се пази и информация за стойностите на кортежите преди и след настъпване на събитието (при действие UPDATE), за новите стойности които ще бъдат вмъкнати (при действие INSERT) и за стойностите, които са били изтрети (при действие DELETE).
- ▶ Не всички СУБД поддържат REFERENCING клауза. При някои от тях, достъпът до стойностите на кортежите преди и след настъпване на събитието се осъществява посредством специални таблици.
- ▶ Например за MySQL и Oracle таблицата NEW съдържа новите стойности, а таблицата OLD съдържа старите стойности. За MSSQL това са специалните таблици INSERTED и DELETED, съответно за новите и старите стойности.

Обяснение

- ▶ Дали действието от тригера да бъде изпълнено за всеки кортеж, засегнат от събитието или еднократно за съответното събитие се указва в клаузите (FOR EACH ROW или FOR EACH STATEMENT).
- ▶ Ако тригерът е FOR EACH ROW тогава се използва (REFERENCING NEW ROW AS N OLD ROW AS O). Ако тригерът е FOR EACH STATEMENT тогава се използва (REFERENCING NEW TABLE AS N OLD TABLE AS O).
- ▶ Отново не всички СУБД поддържат клаузата FOR EACH STATEMENT. При MSSQL дори тази клауза не се използва.
- ▶ И на последно място в дефиницията на тригера се задава условие, което да бъде проверено преди да се изпълни действието на тригера. То се указва в скоби след WHEN клаузата.
- ▶ В тези случаи тялото на тригера ще се изпълни само ако е изпълнено условието в WHEN клаузата. WHEN клаузата не е задължителна, може да имаме действие на тригера и без да има WHEN клауза.
- ▶ Не всички СУБД поддържат WHEN клауза.

BEFORE Синтаксис за DB2

```
CREATE TRIGGER TRIG_MOVIE_LENGTH  
BEFORE UPDATE OF LENGTH ON MOVIE  
REFERENCING NEW AS N OLD AS O  
FOR EACH ROW  
WHEN (N.LENGTH < 0) SET N.LENGTH = O.LENGTH
```

- ▶ С горният синтаксис се създава тригер с име TRIG_MOVIE_LENGTH. Той се задейства при UPDATE на колоната LENGTH на таблицата Movie.
- ▶ Ако новата стойност е отрицателно число, действието на тригера я променя на старата дължина на филм.
- ▶ Тригерът се задейства за всеки ред засегнат от UPDATE.

Тестване на тригера

- ▶ За да тестваме как работи тригера, трябва да направим **UPDATE** за **Movie** на колоната **LENGTH**. Например:

```
SELECT * FROM MOVIE  
WHERE STUDIOName = 'MGM'
```

```
UPDATE MOVIE  
SET LENGTH = -1  
WHERE STUDIOName = 'MGM'
```

- ▶ Горният **UPDATE** би трябвало да промени дължините на всички филми на студио – **MGM**. Но заради дефинирания тригер дължините ще останат не променени.

AFTER Синтаксис за DB2

```
CREATE TRIGGER TRIG_MGM_AVG
AFTER INSERT ON MOVIE
REFERENCING NEW TABLE AS N
FOR EACH STATEMENT
WHEN ((SELECT AVG(LENGTH)
        FROM MOVIE
        WHERE STUDIOName='MGM') > 500)
DELETE FROM MOVIE
WHERE TITLE IN (SELECT TITLE FROM N)
AND YEAR IN (SELECT YEAR FROM N) @
```

- ▶ С горният синтаксис се създава тригер с име TRIG_MGM_AVG. Той се задейства при INSERT в таблицата MOVIE.
- ▶ Ако средната стойност на дължината за 'MGM' е по-голяма от 500, действието на тригерът изтрива редът, който последно сме вмъкнали и който е причината за това дължината да е по-висока от 500. Действието се извършва еднократно за конкретен INSERT.

Тестване на тригера

- ▶ За да тестваме как работи тригера, трябва да направим **INSERT** в таблицата **Movie**. Например:

```
SELECT * FROM MOVIE;
```

```
INSERT INTO MOVIE
```

```
VALUES ('TITLE1', 1999, 5000, 'Y', 'MGM', 199);
```

```
SELECT * FROM MOVIE;
```

Пример за INSTEAD OF тригер

- ▶ Заместващите тригери са тригери, чиито действия се изпълняват вместо самото събитие.
- ▶ Този вид тригери се дефинират за изгледи, тъй като вмъкването на ред, обновяването на ред и изтриването на ред за изглед не е смислена операция.
- ▶ Нека да създадем изглед V_STUDIO, който да ни дава следната информация за студията:

```
CREATE VIEW V_STUDIO
AS
SELECT NAME, ADDRESS, PRESC#
FROM STUDIO;
```

Пример за INSTEAD OF тригер

- ▶ Искаме да създадем тригер, който да се задейства при INSERT в изгледа V_STUDIO;
- ▶ Тъй като вмъкването в изгледа реално означава вмъкване в таблицата STUDIO, действието на тригерът ще реализира точно това.
- ▶ Синтаксис за DB2:

```
CREATE TRIGGER TRIG_V_STUDIO  
INSTEAD OF INSERT ON V_STUDIO  
REFERENCING NEW AS N  
FOR EACH ROW  
    INSERT INTO STUDIO VALUES (N.NAME,  
N.ADDRESS, N.PRESC#);
```

Тестване на тригера

- ▶ За да тестваме как работи тригера, трябва да направим **INSERT** в изгледът **V_AGENCIES**. Например:

```
SELECT *  
FROM STUDIO;
```

```
INSERT INTO V_STUDIO  
VALUES ('New Studio', 'Bulgaria', 199);
```

```
SELECT *  
FROM STUDIO;
```