

Бази от данни

Упражнение 8:

**Промяна на данните
в таблиците**

Димитър Димитров

Въведение

- Досега само извличахме информация от съществуваща БД
- Днес ще разгледаме как да:
 - Добавяме нови редове в таблица
 - Изтриваме редове
 - Променяме стойностите на клетки

Предупреждение №1



- Тъй като вече ще пишем заявки, които променят данните, ще се налага често базите да бъдат възстановявани в първоначалния си вид
 - Например след заявка за изтриване на редове
- Използвайте SQL скриптовете, качени в moodle

Добавяне на редове (1)

- Основен начин – посочваме:
 - Името на таблицата
 - Списък на колони
 - Списък на конкретни стойности, по една за всяка колона
- **INSERT INTO T (c_1, \dots, c_n)
VALUES (v_1, \dots, v_n);**
- Пример: да добавим информация за битката Denmark Strait, която е проведена на 24.05.1941:
- **INSERT INTO Battles (name, date)
VALUES ('Denmark Strait', '1941-05-24');**
- Демонстрация

Добавяне на редове (2)

- Ако не посочим дадена колона (и съответно стойност за нея), стойността ще бъде NULL*

```
INSERT INTO MovieStar (name, gender)  
VALUES ('Jennifer Aniston', 'F');
```

- *По-късно ще видим, че това е стойността по подразбиране за съответната колона, а тя може и да не е NULL

Добавяне на редове (3)

- Можем да пропуснем списъка с колони, ако:
 - Изброим всички колони,
 - Спазвайки техния ред

```
INSERT INTO Battles  
VALUES ( 'Denmark Strait', '1941-05-24' );
```

- Допълнителен материал: SQL Server позволява добавянето на няколко реда с една заявка

```
INSERT INTO T (C1, . . . , Cn)  
VALUES (v1,1, . . . , v1,n),  
        (v2,1, . . . , v2,n),  
        (v3,1, . . . , v3,n);
```

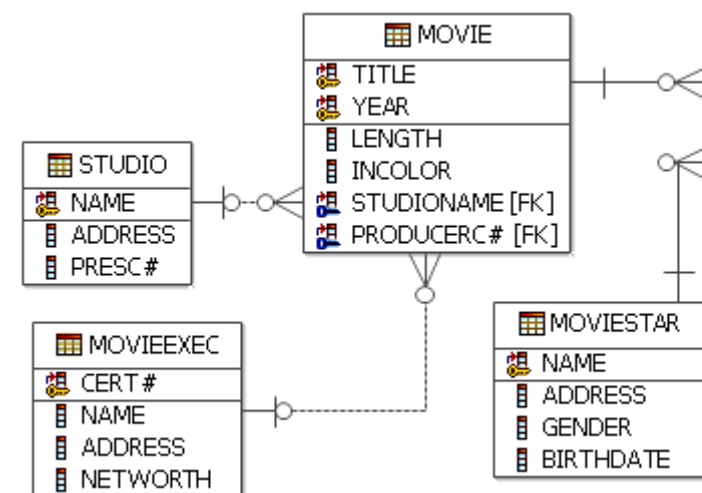
Добавяне на редове (4)

- Можем да добавим редове, получени от съществуващи данни в базата

```
INSERT INTO T (C1, ..., Cn)  
<subquery>;
```

- Пример: да добавим в таблицата Studio всички студиа, споменати в Movie, но липсващи в Studio:

```
INSERT INTO Studio(name)  
SELECT DISTINCT studioName  
FROM Movie  
WHERE studioName  
NOT IN (SELECT name  
FROM Studio);
```



Изтриване на редове (1)

- **DELETE FROM *T***
WHERE <condition>;
- Изтрива всички редове в посочената таблица, удовлетворяващи даденото условие
- Условието е като при SELECT заявките
- WHERE клаузата не е задължителна
- Практичен съвет: първо да напишем SELECT заявка със същата WHERE клауза, за да се уверим, че е вярна; след това да я сменим с DELETE

Изтриване на редове (2)

- Да изтрием току-що добавените редове:

```
DELETE FROM Battles  
WHERE name = 'Denmark Strait';
```

```
DELETE FROM MovieStar  
WHERE name LIKE 'Jen%';
```

Промяна на стойности

- **UPDATE T**
SET $C_1 = \langle \text{expression}_1 \rangle, \dots$
WHERE $\langle \text{condition} \rangle$;
- Отново WHERE клаузата може да се пропусне и отново това е опасно
- Пример: да намалим с 10% цената на всички цветни принтери:

```
UPDATE Printer  
SET price = price * 0.9  
WHERE color = 'y';
```

Предупреждение №2



- Във всички таблици от примерните БД са заложиени ограничения, които не позволяват някои промени, например:
 - Не можем да имаме две студиа с еднакво име
 - Не можем да имаме филм от несъществуващо студио
 - И т.н., ще ги разгледаме в Упражнение №10
- Заявки, коректни според днешния материал, няма да се изпълнят
- Затова ще използваме редуцираните скриптове, в които няма ограничения

Практичен съвет

- Какво е транзакция?
 - (Не се включва в материала за контролното)

- Можем да се възползваме от тях:

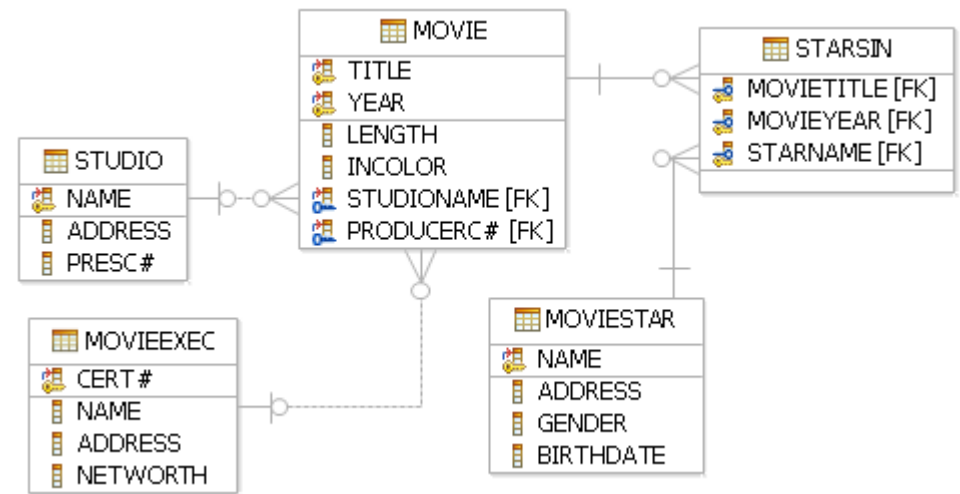
```
begin transaction; -- изпълняваме този ред  
-- пишем си заявки и ги изпълняваме по познатия начин:  
delete from starsin;  
select * from starsin; -- редовете са изтрети  
  
-- накрая пишем:  
rollback transaction;  
-- всичко се връща в първоначалното си състояние  
select * from starsin;
```

Въпроси?

Следват задачи

1. Задачи - Movies

1. Да се добави информация за актрисата Nicole Kidman. За нея знаем само, че е родена на 20-и юни 1967.
2. Да се изтрият всички продуценти с печалба (networth) под 10 милиона.
3. Да се изтрие информацията за всички филмови звезди, за които не се знае адресът.
4. Да се добави титлата „Pres.“ пред името на всеки продуцент, който е и президент на студио.

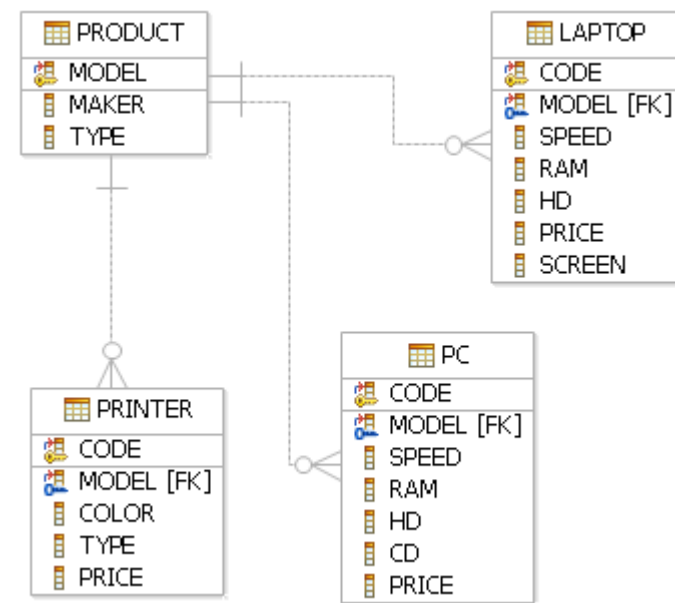


2. Задачи - PC

1. Използвайки две INSERT заявки, съхранете в базата от данни факта, че персонален компютър модел 1100 е направен от производителя C, има процесор 2400 MHz, RAM 2048 MB, твърд диск 500 GB, 52x DVD устройство и струва \$299. Нека новият компютър има код 12. Забележка: моделът и CD са от тип низ.

Упътване: самото вмъкване на данни е очевидно как ще стане, помислете в какъв ред е по-логично да са двете заявки.

2. Да се изтрие всичката налична информация за компютри модел 1100.
3. За всеки персонален компютър се продава и 15-инчов лаптоп със същите параметри, но с \$500 по-скъп. Кодът на такъв лаптоп е със 100 по-голям от кода на съответния компютър. Добавете тази информация в базата.
4. Да се изтрият всички лаптопи, направени от производител, който не произвежда принтери.
5. Производител A купува производител B. На всички продукти на B променете производителя да бъде A.
6. Да се намали два пъти цената на всеки компютър и да се добавят по 20 GB към всеки твърд диск. Упътване: няма нужда от две отделни заявки.
7. За всеки лаптоп от производител B добавете по един инч към диагонала на екрана.



3. Задачи - Ships

1. Два британски бойни кораба (type = 'bb') от класа Nelson - Nelson и Rodney - са били пуснати на вода едновременно през 1927 г. Имали са девет 16-инчови оръдия (bore) и водоизместимост от 34 000 тона (displacement). Добавете тези факти към базата от данни.
2. Изтрийте от Ships всички кораби, които са потънали в битка.
3. Променете данните в релацията Classes така, че калибърът (bore) да се измерва в сантиметри (в момента е в инчове, 1 инч ~ 2.54 см) и водоизместимостта да се измерва в метрични тонове (1 м.т. = 1.1 т.)
4. Изтрийте всички класове, от които има по-малко от три кораба.
5. Променете калибъра на оръдията и водоизместимостта на класа Iowa, така че да са същите като тези на класа Bismarck.

