

Ограничения на ниво таблица

Ограничения на ниво таблица

- Ограниченията са обекти в базите от данни, които налагат определени рестрикции на данните в дадена таблица.
- Ограниченията могат да бъдат зададени при дефиниране на релационната схема или да се добавят след това.

Видове ограничения

■ Първичен ключ

- ▶ Еднозначно определя всеки ред в таблицата.
- ▶ НЕ може да съдържа NULL стойности.
- ▶ Всяка таблица може да има само един първичен ключ.

■ Ограничение за уникалност

- ▶ Еднозначно определя всеки ред в таблицата.
- ▶ Колоната, ограничена с UNIQUE, може да съдържа NULL стойности (*само една за MS SQL Server*).
- ▶ Таблицата може да има повече от един уникален ключ.

Видове ограничения (continued)

■ Външен ключ

- ▶ Атрибут/и от една таблица може да реферира/т атрибут/и от друга таблица. Допуска се първата и втората таблица да съвпадат.
- ▶ Реферираните атрибути от втората таблица трябва да бъдат декларирани като UNIQUE или PRIMARY KEY ограничения.
- ▶ Стойностите на атрибутите, които са декларирани като FOREIGN KEY в първата таблица:
 - трябва да се появяват като стойности на реферираните атрибути в кортежите на втората таблица или
 - могат да приемат NULL стойност (*ако няма NOT NULL ограничение*).

Деклариране на първичен ключ

```
CREATE TABLE MovieStar(  
name CHAR(30) CONSTRAINT pk_ms PRIMARY KEY,  
address VARCHAR(255),  
gender CHAR(1),  
birthdate DATETIME  
);
```

```
CREATE TABLE MovieStar(  
name CHAR(30),  
address VARCHAR(255),  
gender CHAR(1),  
birthdate DATETIME,  
PRIMARY KEY (name)  
);
```

- За всички ограничения **CONSTRAINT** <име_на_ограничение> има незадължителен характер. Ако се използва, на ограничението се задава името <име_на_ограничение> . Ако не се използва, на ограничението се поставя служебно име.

Деклариране на първичен ключ (continued)

```
CREATE TABLE Movie(  
  title VARCHAR(50),  
  year INT,  
  length INT,  
  inColor BIT,  
  studioName VARCHAR(50),  
  producerC# INT,  
  CONSTRAINT pk_m PRIMARY KEY (title,year)  
);
```

Деклариране на UNIQUE ключ

```
CREATE TABLE Movie(  
  title VARCHAR(50),  
  year INT,  
  length INT,  
  inColor BIT,  
  studioName VARCHAR(50),  
  producerC# INT,  
  CONSTRAINT uk_ms UNIQUE (title,year)  
);
```

Деклариране на външен ключ

- Ако външният ключ се състои от един атрибут, може да добавим след името и типа му декларацията:

REFERENCES <parent_table> (<parent_table_attribute>)

- Алтернативно можем да добавим към списъка от атрибути в CREATE TABLE една или повече декларации, указващи че множество от атрибути е външен ключ:

FOREIGN KEY (<child_table_attributes>) **REFERENCES**
<parent_table> (<parent_table_attributes>)

Деклариране на външен ключ (continued)

```
CREATE TABLE Studio (  
name CHAR(30) PRIMARY KEY,  
address VARCHAR(255) ,  
presC# INT CONSTRAINT fk_me REFERENCES  
    MovieExec(cert#)  
);
```

```
CREATE TABLE Studio (  
name CHAR(30) PRIMARY KEY,  
address VARCHAR(255) ,  
presC# INT,  
CONSTRAINT fk_me FOREIGN KEY (presC#) REFERENCES  
    MovieExec(cert#)  
);
```

Добавяне на ограничение

- Общ синтаксис:

ALTER TABLE <име_на_таблица>

ADD CONSTRAINT <име_на_ограничение>
<тип_на_ограничение>;

- **CONSTRAINT** <име_на_ограничение> не е задължително.

Добавяне на ограничение (continued)

- Нека разгледаме таблиците my_table1 и my_table2

```
CREATE TABLE my_table1 (  
  col1 INT NOT NULL,  
  col2 CHAR(2) NOT NULL  
);
```

```
CREATE TABLE my_table2 (  
  col1 CHAR(10) NOT NULL,  
  col2 INT  
);
```

Добавяне на ограничение (continued)

- Нека добавим първичен ключ за колоната col1 на таблицата my_table1. Типът на ограничението е **PRIMARY KEY**.

```
ALTER TABLE my_table1  
ADD CONSTRAINT pk_my_table1 PRIMARY KEY(col1);
```

- Нека добавим ограничение за уникалност за колоната col1 на таблицата my_table2. Типът на ограничението е **UNIQUE**.

```
ALTER TABLE my_table2  
ADD CONSTRAINT uq_my_table2 UNIQUE(col1);
```

Добавяне на ограничение (continued)

- Нека да добавим към таблицата my_table2 външен ключ, който да реферира таблицата my_table1. Типът на ограничението е **FOREIGN KEY**. Използва се и ключовата дума REFERENCES.

```
ALTER TABLE my_table2  
ADD CONSTRAINT fk_to_my_table1 FOREIGN KEY(col2)  
REFERENCES my_table1(col1);
```

Политики при външен ключ

- При създаване на външен ключ могат да бъдат зададени политики за налагане на ограничение по референтна цялостност. Тези политики определят поведението на СУБД при обновяване (ON UPDATE) или изтриване (ON DELETE) на реферирания запис.
- Политиките са:
 - ▶ **NO ACTION** - всяко действие, което нарушава референтна цялостност, се отхвърля.
 - ▶ **CASCADE** – при изтриване/ промяна на стойност в реферираната колона се изтрива/ обновява съответната стойност в рефериращите записи.
 - ▶ **SET NULL** - при изтриване/ промяна на стойност в реферираната колона се поставя стойност NULL в съответните рефериращи записи (*ако няма NOT NULL ограничение*).
 - ▶ **SET DEFAULT** – всички кортежи, които сочат към кортежи, които ще бъдат изтрети/ променени, получават стойността си по подразбиране. Тази стойност трябва да съществува в реферираната колона.

Политики при външен ключ (continued)

- Политиките при външни ключове могат да бъдат указани при дефиниране на релационната схема или да се добавят след това. Общ синтаксис:

```
REFERENCES <table_name>(<col_name>)  
[ON DELETE { CASCADE | NO ACTION | SET DEFAULT | SET NULL}]  
[ON UPDATE { CASCADE | NO ACTION | SET DEFAULT | SET NULL}]
```

- Не е задължително изрично да се укажат политики за външните ключове. Може да се посочи само политика при DELETE или само политика при UPDATE или и за двете действия.
- Ако не укажем изрично политика, политиката по подразбиране е NO ACTION.
- За един и същ външен ключ политиката при DELETE може да е една (напр. CASCADE), а политиката при UPDATE да е различна (напр. NO ACTION).

Добавяне на външен ключ с политика

- Нека добавим политики към външния ключ на таблицата my_table2.

```
ALTER TABLE my_table2  
ADD FOREIGN KEY(col2)  
REFERENCES my_table1(col1)  
ON DELETE CASCADE  
ON UPDATE SET NULL;
```


CHECK ограничение

- Ограничението CHECK проверява дали записите, които се вмъкват или обновяват, отговарят на предварително зададено условие.
- Като условие може да се използва всяко условие, което може да се постави и в WHERE клаузата на една SQL заявка
 - ▶ *MS SQL Server* - условията трябва да са прости и да съдържат логически изрази свързани с AND и/или OR, т.е. не могат да съдържат подзаявки.
- Ограничението CHECK, може да бъде създадено на ниво таблица и на ниво кортеж.

Добавяне на CHECK ограничение

(при дефиниране на релационната схема)

- Нека дефинираме CHECK ограничение за таблицата my_table1, което да не позволява вмъкването на други стойности в колоната col2 освен 'BG' и 'FR'

▶ Ограничение CHECK на ниво кортеж

```
CREATE TABLE my_table1 (  
  col1 INT NOT NULL,  
  col2 CHAR(2) NOT NULL CHECK (col2 IN ('BG', 'FR'))  
);
```

▶ Ограничение CHECK на ниво таблица

```
CREATE TABLE my_table1 (  
  col1 INT NOT NULL,  
  col2 CHAR(2) NOT NULL,  
  CHECK (col2 IN ('BG', 'FR'))  
);
```

Добавяне на CHECK ограничение (continued)

(при дефиниране на релационната схема)

- Предимството на добавянето на CHECK ограничение чрез втория вариант (на ниво таблица) е, че условието може да съдържа логически израз, включващ няколко колони от таблицата. Чрез първия вариант (на ниво кортеж) това е невъзможно. Напр.:

```
CREATE TABLE my_table1 (  
  col1 INT NOT NULL,  
  col2 CHAR(2) NOT NULL,  
  CHECK (col2 IN ('BG', 'FR') AND col1 > 1  
    AND col1 < 10)  
);
```

Добавяне на CHECK ограничение

(след като таблицата е създадена)

```
ALTER TABLE my_table1  
ADD CONSTRAINT ck_col2  
CHECK(col2 IN ('BG', 'FR'));
```

- За col2 може да се добави повече от едно CHECK ограничение. В този случай всички условия от дефинираните CHECK ограничения за една и съща колона се свързват логически с оператор AND. Напр.:

```
ALTER TABLE my_table1  
ADD CONSTRAINT ck1_col2  
CHECK(col2 IN ('BG', 'FR'));
```

```
ALTER TABLE my_table1  
ADD CONSTRAINT ck2_col2  
CHECK(col2 IN ('GR', 'RU'));
```

- *Проблеми с двете ограничения по-горе?*

Изтриване на ограничение

- Общ синтаксис:

```
ALTER TABLE <име_на_таблица>
```

```
DROP CONSTRAINT <име_на_ограничение>;
```

- Примери:

- ▶ Първичен ключ:

```
ALTER TABLE my_table1
```

```
DROP CONSTRAINT pk_my_table1;
```

- ▶ Външен ключ:

```
ALTER TABLE my_table2
```

```
DROP CONSTRAINT fk_to_my_table1;
```

- ▶ UNIQUE ограничение:

```
ALTER TABLE my_table2
```

```
DROP CONSTRAINT uq_my_table2;
```

- ▶ CHECK ограничение:

```
ALTER TABLE my_table1
```

```
DROP CONSTRAINT ck_col2;
```

Задачи