Дефиниране на схеми на релации. Ограничения.

SQL: CREATE, ALTER, DROP

Дефиниране на релационна схема

Атрибутите имат типове

- CHAR(n) –низ с фиксирана дължина
- VARCHAR(n) –низ с дължина до n символа
- INT / INTEGER –цяло число, 32-bit, със знак
- FLOAT / REAL, DOUBLE
- DECIMAL(n, d) –дробно число с n цифри, d от тях след десетичната запетая (има разлика от FLOAT)
- DATE, TIME

Създаване на релация

MovieStar (name, address, gender, birthdate)

```
CREATE TABLE MovieStar
(name CHAR(30),
 address VARCHAR(255),
 gender CHAR(1),
 birthdate DATE DEFAULT CURRENT_DATE
);
```

Обновяване на релация

- Добавяне на атрибут в релация

```
ALTER TABLE MovieStar
ADD phone CHAR(16);
```

- Изтриване на атрибут

```
ALTER TABLE MovieStar
DROP birthdate;
```

Ограничения

- Ограниченията са обекти в базите от данни, които налагат определени рестрикции на данните в дадена таблица.
- Пример за ограничение е първичен ключ на дадена таблица, външен ключ на дадена таблица и други.
- Според това, колко таблици обхваща едно ограничение, ограниченията могат да бъдат - ограничения на ниво таблица и ограничения на ниво схема на базата от данни.

Ограничения на ниво таблица

- Ограниченията на ниво таблица могат да бъдат зададени при създаването на таблицата или след като таблицата е вече създадена.

Ограничение	Кратко име	Описание	Кога се проверява?
PRIMARY KEY	Първичен ключ	Ограничава стойностите да са уникални и да са различни от NULL.	При INSERT, при UPDATE на стойност от колоната, при DELETE на първичния ключ
UNIQUE	Ограничение за уникалност	Ограничава стойностите да са уникални	При INSERT, при UPDATE на стойност от колоната, при DELETE
FOREIGN KEY	Външен ключ	Ограничава стойностите да са от определено множество от стойности	При INSERT, при UPDATE на стойност от колоната, при DELETE на стойност от колоната на реферираната таблица
CHECK	Ограничение за проверка	Ограничава стойностите да отговарят на дадено условие	При INSERT, при UPDATE на стойност от колоната
NOT NULL	Ограничение за не NULL стойности		При INSERT, при UPDATE на стойност от колоната

Видове ограничения

- NOT NULL ограничения
- Първичен ключ (Primary Key) - PK
  - **Първичният ключ** притежава следните свойства:
    - Еднозначно определя всеки ред в таблицата;
    - НЕ може да съдържа NULL стойности;
    - Всяка таблица може да има само един първичен ключ;
- UNIQUE ограничение - UK
  - **Ограничението за уникалност** притежава следните свойства:
    - Еднозначно определя всеки ред в таблицата;
    - Колоната ограничена с UNIQUE може да съдържа NULL стойности.
- Външен ключ (Foreign Key) - FK
- CHECK ограничения - CK

NOT NULL ограничение

- Не се допуска даден атрибут да приема NULL стойности;
- Декларира се САМО на ниво атрибут с ключовите думи NOT NULL

```
CREATE TABLE MovieStar
(name CHAR(30) not null,
 address VARCHAR(255)
);
```

- След като таблицата е създадена, се декларира като промяна на дефиницията на колоната

```
ALTER TABLE T
ALTER COLUMN col1 col1_type NOT NULL
```

Деклариране на първичен ключ върху един атрибут

- Декларация на ниво атрибут

```
CREATE TABLE MovieStar(
  name CHAR(30)NOT NULL PRIMARY KEY,
  address VARCHAR(255),
  gender CHAR(1),
  birthdate DATE
);
```

- На ограничението се поставя служебно име.

Деклариране на първичен ключ върху един атрибут

- Декларация на ниво таблица

```
CREATE TABLE MovieStar(
  name CHAR(30) NOT NULL,
  address VARCHAR(255),
  gender CHAR(1),
  birthdate DATE ,
  PRIMARY KEY (name)
);
```

- На ограничението се поставя служебно име.

Деклариране на съставен първичен ключ

- Възможно е САМО на ниво таблица.

```
CREATE TABLE Movie(
  title varchar(50) NOT NULL,
   year  integer NOT NULL,
  length integer,
  inColor char(1),
  studioName varchar(50),
  PRIMARY KEY (title,year)
);
```

Деклариране на първичен ключ след създаване на таблицата

```
ALTER TABLE Movie
ADD [ CONSTRAINT Movie_pk ]
PRIMARY KEY (title,year)

ALTER TABLE MovieStar
ADD [ CONSTRAINT MovieStar_pk ]
PRIMARY KEY (name)
```

Външен ключ (FOREIGN KEY)

- Ограничение за референтна цялостност
- Гарантира, че стойностите на атрибутите от релацията, в която е дефиниран външния ключ, се срещат в съответните атрибути от релацията, която се реферира от външния ключ.
- Реферираните атрибути от втората релация трябва да бъдат декларирани като UNIQUE или PRIMARY KEY.
- Връзка (Parent – Child). FOREIGN KEY се декларира в таблицата Child към първичния ключ на Parent

Деклариране на външен ключ

- Декларация на ниво атрибут

```
REFERENCES  <parent_table>
(<parent_table_attribute>)
```

- На ниво таблица

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

Деклариране на външен ключ

- Декларация на ниво атрибут

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

- Тъй като полето presC# допуска NULL стойности, разрешените стойности за него са NULL или реално съществуващ (регистриран в MovieExec) режисьор.

Деклариране на съставен външен ключ

- САМО на ниво таблица

```
CREATE TABLE StarsIn(
  movietitle varchar(50),
  movieyear integer,
  name char(30),
  FOREIGN KEY (movietitle, movieyear)
  REFERENCES Movie(title, year)
);
```

Деклариране на външен ключ след създаване на таблицата

```
ALTER TABLE StarsIn
ADD [ CONSTRAINT MovieStar_FK ]
  FOREIGN KEY (starname)
  REFERENCES MovieStar(name);

ALTER TABLE StarsIn
ADD [ CONSTRAINT MovieStar_FK ]
  FOREIGN KEY (movietitle, movieyear)
  REFERENCES Movie(title, year);
```

Добавяне на външен ключ с политика

```
ALTER TABLE StarsIn
ADD [ CONSTRAINT MovieStar_FK ]
FOREIGN KEY (movietitle, movieyear)
REFERENCES Movie(title, year)
ON DELETE CASCADE
ON UPDATE RESTRICT;
```

Добавяне на CHECK ограничение

- Друг тип ограничение е CHECK. То 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, само че условията трябва да са прости и да съдържат логически изрази свързани с AND и/или OR, т.е. не могат да съдържат подзаявки.
- Условието в CHECK ограничението се проверява всеки път, когато вмъкваме запис в таблицата или обновяваме запис от таблицата, при което атрибутът за когото е дефиниран CHECK ограничението получава нова стойност (в резултат на INSERT или UPDATE).
- Ограничението CHECK, може да бъде създадено на ниво таблица и на ниво кортеж.

Добавяне на CHECK ограничение

- Например нека да дефинираме CHECK ограничение за таблицата MOVIESTAR, което да не позволява вмъкването на други стойности в колоната gender от стойностите ‘M’ и ‘F’

- **Ограничение CHECK на ниво кортеж (при създаване на таблицата)**

```
CREATE TABLE MOVIESTAR
( ……,
 gender char(1) NOT NULL check (gender in ('M', 'F'))
);
```

- **Ограничение CHECK на ниво таблица (при създаване на таблицата)**

```
CREATE TABLE MOVIESTAR
( ……,
 gender char(1) NOT NULL,
 check (gender in ('M', 'F'))
);
```

Добавяне на CHECK ограничение

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

```
CREATE TABLE my_table1
( col1 int NOT NULL,
 col2 char(2) NOT NULL,
 check (col2 in ('BG', 'FR')
        and col1 > 1 and col1 < 10)
);
```

Изтриване на ограничение

- Синтаксисът за изтриване на ограничение е следният:

```
ALTER TABLE <име_на_таблица>
DROP CONSTRAINT <име_на_ограничение>;
```

- Например:

```
ALTER TABLE my_table1
DROP CONSTRAINT PK_MY_TABLE1;

ALTER TABLE my_table1
DROP CONSTRAINT ck_col2;
```

Премахване (изтриване) на релация

- Изтриване на релация

```
DROP TABLE  MovieStar;
```