



Национална програма
"Обучение за ИТ умения и кариера"
<https://it-kariera.mon.bg>

Министерството на
образованието и науката
<https://www.mon.bg>



Ограничения (Constraints)

Бази данни

Ограничения [1/2]

- Ограниченията задават правила за данните в дадена таблица
- Ето примери за често срещани ограничения:
 - **NOT NULL** – не позволява в клетка да седи празна стойност
 - **UNIQUE** – гарантира, че всички данни в колоната са различни
 - **PRIMARY KEY** – комбинация от **NOT NULL** и **UNIQUE**
 - Уникално идентифицира всеки запис в таблицата
 - **FOREIGN KEY** – уникално идентифицира запис от друга таблица

Ограничения [1/2]

- Ограниченията задават правила за данните в дадена таблица
- Ето примери за често срещани ограничения:
 - **CHECK** – Гарантира, че всички данни в колоната удовлетворяват дадено условие
 - **DEFAULT** – Задава стойност по подразбиране за колоната, когато не е зададена такава при вмъкване на записа
 - **INDEX** – Използва се, за да се създава и извлича информация от таблицата много бързо

NOT NULL

- По подразбиране колоните могат да имат празна **NULL** стойност. За да се забрани това се налага **NOT NULL** ограничение
 - Ограничението се добавя при създаване или чрез ALTER TABLE

```
CREATE TABLE Persons(  
    PersonID int NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    LastName varchar(255) NOT NULL,  
    Age int  
);
```

UNIQUE

- Гарантира уникалност на всички данни в колоната
 - Ограничението се добавя при създаване или чрез **ALTER TABLE**

```
CREATE TABLE Persons(  
    PersonID int NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    LastName varchar(255) NOT NULL,  
    Age int,  
    UNIQUE(PersonID)  
);
```

PRIMARY KEY [1/2]

- Първичният (основен) ключ идентифицира уникнално запис в таблицата
 - Създаване на първичен ключ на базата на **една** колона:

```
CREATE TABLE Persons(  
    PersonID int NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    LastName varchar(255) NOT NULL,  
    Age int,  
    PRIMARY KEY(PersonID)  
);
```

PRIMARY KEY [2/2]

- Създаване на първичен ключ на базата на **gbe или повече** колони и именуване на ограничението:

```
CREATE TABLE Persons(  
    PersonID int NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    LastName varchar(255) NOT NULL,  
    Age int,  
    CONSTRAINT PK_Person PRIMARY KEY(PersonID, LastName)  
);
```

FOREIGN KEY

- Външният ключ се използва за свързване на две таблици. Външният ключ е колона/и, която сочи към първичния ключ в друга таблица.

```
CREATE TABLE Orders(  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY(OrderID),  
    FOREIGN KEY(PersonID) REFERENCES Persons(PersonID)  
);
```


CHECK [1/2]

- **CHECK** ограничението се използва за допускане на стойности в колоната, които отговарят само на дадено условие

```
CREATE TABLE Persons(  
    PersonID int NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    LastName varchar(255) NOT NULL,  
    Age int,  
    CHECK (Age >= 18)  
);
```

CHECK [2/2]

- CHECK ограничението може да се наложи и за няколко колони:

```
CREATE TABLE Persons(  
    PersonID int NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    LastName varchar(255) NOT NULL,  
    Age int,  
    City varchar(255),  
    CONSTRAINT CHK_Person CHECK (Age >= 18 AND City='Burgas')  
);
```

DEFAULT

- DEFAULT ограничението предоставя стойност по подразбиране за колона:

```
CREATE TABLE Persons(  
    PersonID int NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    LastName varchar(255) NOT NULL,  
    Age int,  
    City varchar(255) DEFAULT 'Plovdiv'  
);
```

INDEX

- INDEX ограничението създава индекси. Индексите могат да се използват, за да се повиши скоростта на извличане на данни:

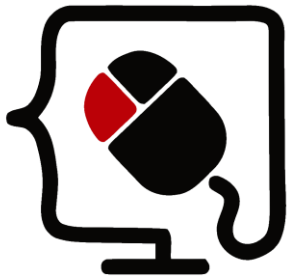
```
CREATE INDEX idx_lastname ON Persons (LastName);
```

- Създавайте индекси само за често претърсвани колони. Обновяването на таблица с индекси, отнема повече време отколкото за същата таблица без индекси, понеже индексите също имат нужда от обновяване.

Auto Increment

- Auto Increment генерира ново число всеки път, когато се добави нов запис. Auto Increment се използва към първичния ключ:

```
CREATE TABLE Persons(  
    PersonID int NOT NULL AUTO_INCREMENT,  
    FirstName varchar(255) NOT NULL,  
    LastName varchar(255) NOT NULL,  
    Age int,  
    PRIMARY KEY(PersonID)  
);
```



Национална програма
"Обучение за ИТ умения и кариера"
<https://it-kariera.mon.bg>

Министерството на
образованието и науката
<https://www.mon.bg>



Документът е разработен за нуждите на Национална програма "Обучение за ИТ умения и кариера" на Министерството на образованието и науката (МОН) и се разпространява под свободен лиценз CC-BY-NC-SA (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).