

# Имплементиране интегритета на данните (Data Integrity)

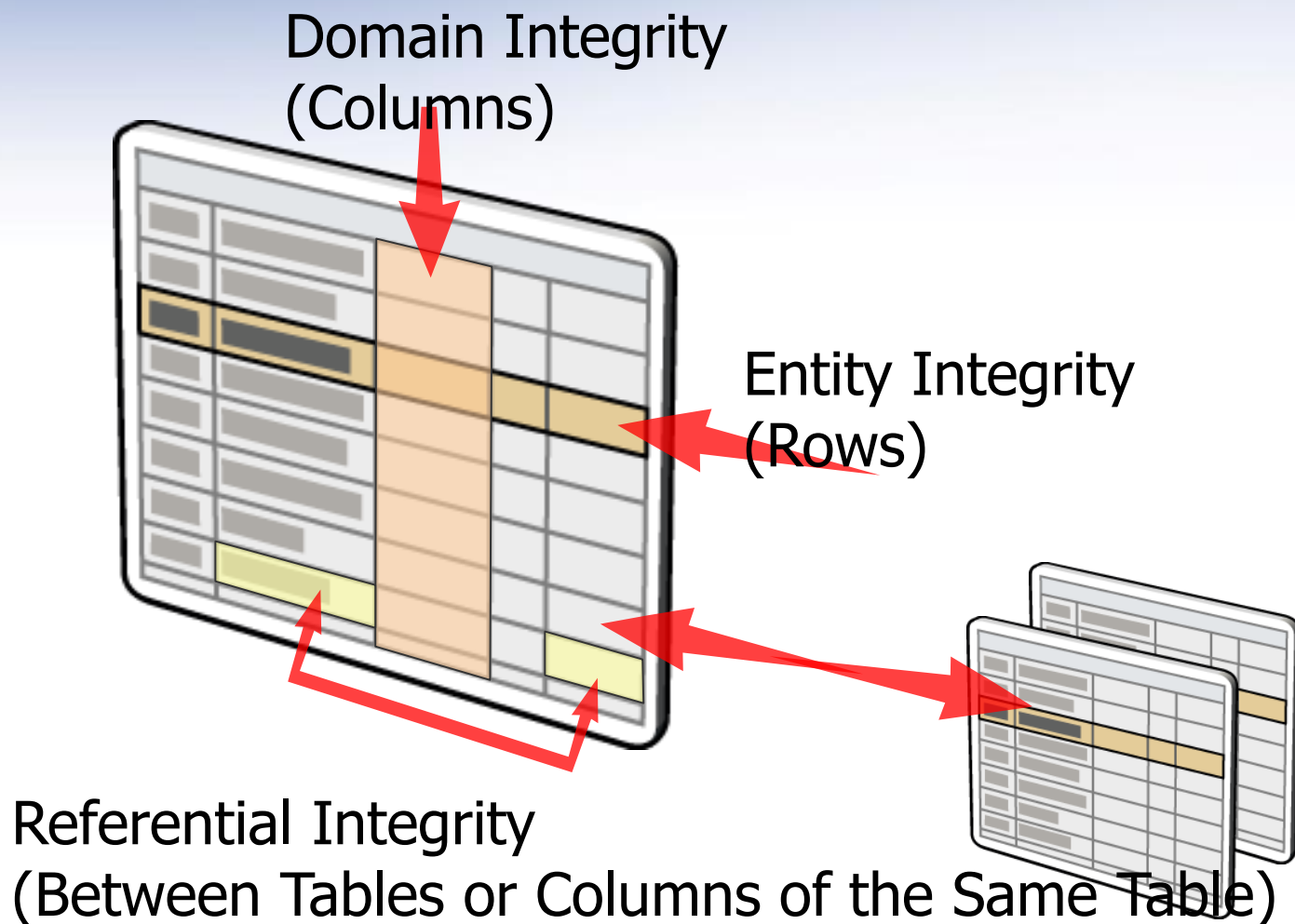
- Типове интегритет на данните
- Методи за осигуряване  
интегритет на данните
- Дефиниране на ограничения



# Типове интегритет на данните

- Интегритет на същността  
(Entity Integrity)
- Домейн интегритет  
(Domain Integrity)
- Интегритет на връзките  
(Referential Integrity)
- Потребителски дефиниран интегритет  
(User-Defined Integrity)

# Типове интегритет на данните



# Методи за осигуряване интегритет на данните

- **Декларативни методи**

Изискванията към данните се описват като свойства на колоните и таблиците при дефиниране на таблиците или се добавят при модификацията им.

- **Ограничения (Constraints )** – определят правила относно допустимите стойности в колоните и редовете на таблицата.

- **Индекси (Indexes)**

# Методи за осигуряване интегритет на данните

- Процедурни методи за осигуряване интегритет на данните
  - Скриптове и процедури
  - Тригери (Triggers) -  
специален клас съхранени процедури, които се изпълняват автоматично, когато за една таблица или изглед се зададе конструкция UPDATE, INSERT или DELETE

# Ограничения (constraints) за осигуряване интегритет на данните

- Ограниченията могат да се прилагат:
  - на ниво колона;
  - на ниво таблица.
- Задават се при:
  - при създаване (дефиниране) на таблицата (CREATE TABLE);
  - при промяна дефиницията на таблицата (ALTER TABLE).
- На ограничението се задава уникално име.

# Дефиниране на Entity Integrity

Дефинира се чрез

- PRIMARY KEY – първичен ключ.
- UNIQUE
  - осигурява уникалност на стойностите, приемани от колоната;
  - допуска колоната да приема неопределени стойности Null само в един ред от таблицата.

# Ограничения PRIMARY KEY

- PRIMARY KEY – задава се за една или няколко колони в таблицата и указва:
  - стойностите на първичния ключ да бъдат уникални;
  - не се разрешава Null стойности в колоните, съставлящи първичния ключ.
- В таблицата може да има само едно ограничение PRIMARY KEY.
- По подразбиране се създава клъстърен индекс, ако няма създаден по друга колона



# Пример за създаване на ограничение PRIMARY KEY

- При създаване на таблицата
  - *На ниво колона (само за прости PK)*

```
CREATE TABLE Products  
(ProductId INT PRIMARY KEY ,  
  ProductName VARCHAR(30)  
  , UnitPrice Money,....)
```

# Ограничение Съставен първичен ключ

**CONSTRAINT***constraint\_name*  
**PRIMARY KEY ( column [ ,...n ] )**

***! Задава се на ниво таблица***

Пример:

```
CREATE TABLE [Order Details]
( OrderID INT NOT NULL,
  ProductID CHAR(6) NOT NULL,
  UnitPrice SMALLMONEY,
  Quantity DECIMAL,
  Discount DECIMAL,
CONSTRAINT order_details_key
PRIMARY KEY (OrderID, ProductID) )
```

# Ограничения

## UNIQUE

- Осигурява уникалност на стойностите, приемани от колоната
- Допуска колоната да приема неопределени стойности Null (само в един ред от таблицата)
- Към колона, за която е зададено ограничение UNIQUE може да сочи ограничение FOREIGN KEY

# Ограничения UNIQUE

## Пример:

```
USE Northwind
```

```
GO
```

```
CREATE TABLE Employee1
```

```
    (EmployeeID INT IDENTITY(1,1) PRIMARY KEY
```

```
    , EGN NCHAR(10) UNIQUE NOT NULL
```

```
    , FIRSTNAME VARCHAR(15)
```

```
    , LASTNAME VARCHAR(30)
```

```
    , BIRTHDAY DATETIME
```

```
    , REPORTTO NCHAR(10))
```

# Дефиниране на Domain Integrity

- DEFAULT – стойност по подразбиране.
- NOT NULL – колоната не може да приема неопределени стойности.
- CHECK - чрез логически израз се задават валидните стойности, които може да приема колоната.

# Ограничение

## DEFAULT constant\_expression

- Дефинира стойност, която колоната приема по подразбиране, ако не се зададе стойност
- За една колона може да се зададе само едно ограничение DEFAULT
- Прилага се само в операторите INSERT
- Пример: CREATE TABLE Orders  
(OrderId INT PRIMARY KEY  
, OrderDate DATETIME  
                    DEFAULT (getdate()),...)

# Ограничения

## CHECK (*search\_conditions* )

- Ограничава стойностите, които могат да бъдат въвеждани в колоната с операторите INSERT или UPDATE
- Валидните стойности се определят чрез логически израз
- Могат да се дефинират няколко CHECK ограничения за една колона
- Могат да се обръщат към колони от същата таблица
- Не могат да съдържат подзаявки

# Пример на ограничение CHECK

```
CREATE TABLE Products  
  (ProductId INT PRIMARY KEY,  
   ProductName NVARCHAR(30)  
  , UnitPrice Money  
   CHECK (UnitPrice BETWEEN 1 and 100))
```



# Ограничения FOREIGN KEY за референтен интегритет

- **FOREIGN KEY** ограниченията осигуряват референтен интегритет между колоните в същата или различни таблици. Задава изискването колоните да изпълняват изискванията за външния ключ

```
[CONSTRAINT constraint_name]
[ FOREIGN KEY ( column [ ,...n ] ) ]
REFERENCES ref_table [ ( ref_column ) ]
[ ON DELETE { CASCADE | NO ACTION } ]
[ ON UPDATE { CASCADE | NO ACTION } ]
```

-

# Ограничение

## FOREIGN KEY...REFERENCES

- FOREIGN KEY ( *column* [ ,...*n* ] ) – колоната (колоните), явяваща(и) се външен ключ
- *ref\_table* – име на таблицата, към която се обръща външният ключ (таблицата родител)
- (*ref\_column*[,...*n*]) – колона или списък от колони от таблицата, към която е обръщението
- FOREIGN KEY ограничението може да се обръща само към колони, за които
  - е зададено ограничение PRIMARY KEY
  - или е зададено ограничение UNIQUE
  - или за таблицата, към която се обръща външният ключ е създаден уникален индекс  
UNIQUE INDEX

# Пример: Дефиниране на ограничението FOREIGN KEY при създаване на таблицата

USE Northwind

GO

CREATE TABLE Payment

(PaymentId INT NOT NULL

, PaymentDate DATETIME NOT NULL

, CustomerId NCHAR(5) NOT NULL

, Total MONEY

CONSTRAINT Payment\_PK PRIMARY KEY

(PaymentId, PaymentDate)

CONSTRAINT Payment\_Cust FOREIGN KEY (CustomerId)

REFERENCES Customers (CustomerId)

)

# Добавяне на ограничения

- Оператор **ALTER TABLE table\_name  
ADD CONSTRAINT .....**
- Добавяните ограничения са на ниво таблица
- Пример:

```
ALTER TABLE Payment  
ADD CONSTRAINT PaymentDate_Def  
DEFAULT (getdate()) FOR PaymentDate
```

## Пример на рекурсивен референтен интегритет:

```
CREATE TABLE [dbo].[Employees1]
(EmployeeID [int] IDENTITY(1,1) PRIMARY KEY,
LastName nvarchar(20) NOT NULL,
FirstName nvarchar(10) NOT NULL,
EGN nchar(10) UNIQUE,
BirthDate datetime NULL,
Address nvarchar (60) NULL,
City nvarchar(15) NULL, Country nvarchar(15) NULL,
ReportsTo nchar(10) -- EGN на прекия ръководител
)
GO
```

## Пример: Добавяне на ограничение рекурсивен FOREIGN KEY

```
USE Northwind
```

```
GO
```

```
ALTER TABLE Employees1
```

```
    ADD CONSTRAINT REFERENCES_REPORTSTO
```

```
    FOREIGN KEY (Reportsto) REFERENCES
```

```
    Employees1 (EGN)
```

/\* колоната Reportsto е рекурсивен външен ключ,  
който се обръща към колоната EGN в същата  
таблица; колоната EGN е кандидат ключ и е  
дефинирана със свойство UNIQUE \*/

# Премахване на ограничение

- **ALTER TABLE table\_name  
DROP CONSTRAINT constraint\_name**

- *Пример:*

```
ALTER TABLE Orders  
DROP CONSTRAINT Orders_pk
```

- Особеност: ограничение PRIMARY KEY не може да се премахне, ако към него сочи ограничение FOREIGN KEY в друга таблица

# Промяна на ограничение

- Премахване на съществуващото ограничение с  
`ALTER TABLE table_name  
DROP CONSTRAINT constraint_name`
- Създаване на нова дефиниция -добавяне на ограничението с  
`ALTER TABLE table_name  
ADD CONSTRAINT constraint_name.....`



## Клаузи за каскаден интегритет в ограничението FOREIGN KEY...REFERENCES...

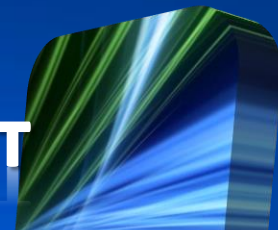
- *Действия при изтриване на редове в таблицата-родител*

[ ON DELETE { NO ACTION | CASCADE  
| SET NULL | SET DEFAULT } ]

- *Действия при промяна стойността на колоните в таблицата-родител, към които се обръща външният ключ*

[ ON UPDATE { NO ACTION | CASCADE  
| SET NULL | SET DEFAULT } ]

# Каскаден референтен интегритет



Контролира се от клаузата **CASCADE** в ограничението **FOREIGN KEY**

Cascade опция	UPDATE поведение	DELETE поведение
NO ACTION (Default)	Произвежда грешка; операцията се отменя	
CASCADE	Обновява външните ключове в референтната таблица	Изтрива редове от референтната таблица
SET NULL	Поставя NULL във външните ключове в референтната таблица	
SET DEFAULT	Поставя DEFAULT стойности във външните ключове в референтната таблица	

# Пример:

```
USE [Northwind]
```

```
GO
```

```
/* премахване на ограничението за FK */
```

```
ALTER TABLE [dbo].[Order Details]
```

```
DROP CONSTRAINT [FK_Order_Details_Orders]
```

```
/* добавяне на ограничението за FK с опция за каскадно  
изтриване*/
```

```
ALTER TABLE [dbo].[Order Details]
```

```
ADD CONSTRAINT [FK_Order_Details_Orders]
```

```
FOREIGN KEY([OrderID])
```

```
REFERENCES [dbo].[Orders] ([OrderID])
```

```
ON DELETE CASCADE
```

```
GO
```

# Анулиране на ограничения **CHECK** и **FOREIGN KEY** към съществуващите данни в таблицата

- Клаузата **WITH NOCHECK** задава при създаване на ограничението да не се прави проверка дали съществуващите данни в таблицата отговарят на изискванията.
- Проверка се прави при добавяне на нови данни и при промяна на съществуващите данни в бъдеще.
- По подразбиране клаузата е **WITH CHECK**.

# Пример:

```
USE [Northwind]
```

```
GO
```

```
ALTER TABLE [dbo].[Order Details]
```

```
WITH NOCHECK
```

```
ADD CONSTRAINT [DF_Order_Details_Quantity]
```

```
CHECK ([Quantity]<=10)
```

```
GO
```

Пример: Към таблицата Orders да се добави ограничение за датата, която не може да бъде 30 дни преди текущата дата без да се проверяват съществуващите данни

```
ALTER TABLE Orders
```

```
WITH NOCHECK
```

```
ADD CONSTRAINT Order_date
```

```
CHECK (OrderDate
```

```
    BETWEEN getdate() - 30 AND getdate())
```

# Какви ограничения и с какви опции са зададени в следващия пример?

```
USE Northwind
```

```
GO
```

```
ALTER TABLE Orders
```

```
    DROP CONSTRAINT FK_Orders_Customers
```

```
ALTER TABLE Orders
```

```
    ADD CONSTRAINT FK_Orders_Customers
```

```
    FOREIGN KEY (CustomerID)
```

```
    REFERENCES CUSTOMERS(CustomerID)
```

```
    ON DELETE NO ACTION
```

```
    ON UPDATE CASCADE
```

# Практически съвети:

- Ограничения FOREIGN KEY да се създават чрез ALTER TABLE .....ADD CONSTRAINT.... след като са създадени таблиците
- Ограничения CHECK, които задават нови правила за верността на данните и противоречат на зададени и действащи до момента други ограничения, се добавят с опция WITH NOCHECK в ALTER TABLE ....



# Определяне свойствата на таблица

- Използва се системната съхранена процедура

`sp_helpconstraint table_name`

- Пример:

USE Northwind

GO

EXEC sp\_helpconstraint [Order Details]

SQLQuery7.sql - WIN-6C20A8VN2U0.Northwind (WIN-6C20A8VN2U0\MK (59))\* - Microsoft SQL Server Management Studio

File Edit View Query Project Debug Tools Window Help

Northwind Execute Debug

Object Explorer

- Databases
  - System Databases
  - Database Snapshots
  - AdventureWorks2012
  - AdventureWorksDW2012
  - AdventureWorksLT2012
  - ContosoRetailDW
  - ContosoSalesSQL
  - Northwind
    - Database Diagrams
    - Tables
      - System Tables
      - FileTables
      - dbo.Categories
      - dbo.CustomerCustomerDemo
      - dbo.CustomerDemographics
      - dbo.Customers
      - dbo.Employees
      - dbo.Employees1
      - dbo.EmployeeTerritories
      - dbo.Order Details
      - dbo.Orders
        - Columns
        - Keys
          - PK\_Orders
          - FK\_Orders\_Customers
          - FK\_Orders\_Employees
          - FK\_Orders\_Shippers
        - Constraints

SQLQuery7.sql - W...0A8VN2U0\MK (59))\* SQLQuery2.sql - W...0A8VN2U0\MK (53))\* SQLQuery6.sql - W...0A8VN2U0\MK (52)) SQLQuery5.sql - W...0A8VN2U0\MK (57))\*

```
EXEC sp_helpconstraint [Order Details]
```

100 %

Results Messages

Object Name	constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
1 Order Details	CHECK on column Discount	CK_Discount	(n/a)	(n/a)	Enabled	Is_For_Replication	((Discount]>=0) AND [Discount]<=(1))
2	CHECK on column Quantity	CK_Quantity	(n/a)	(n/a)	Enabled	Is_For_Replication	((Quantity]>0))
3	CHECK on column UnitPrice	CK_UnitPrice	(n/a)	(n/a)	Enabled	Is_For_Replication	((UnitPrice]>=0))
4	DEFAULT on column Discount	DF_Order_Details_Discount	(n/a)	(n/a)	(n/a)	(n/a)	((0))
5	CHECK on column Quantity	DF_Order_Details_Quantity	(n/a)	(n/a)	Enabled	Is_For_Replication	((Quantity]<=(10))
6	DEFAULT on column UnitPrice	DF_Order_Details_UnitPrice	(n/a)	(n/a)	(n/a)	(n/a)	((0))
7	FOREIGN KEY	FK_Order_Details_Orders	No Action	No Action	Enabled	Is_For_Replication	OrderID
8							REFERENCES Northwind.dbo.Orde...
9	FOREIGN KEY	FK_Order_Details_Products	No Action	No Action	Enabled	Is_For_Replication	ProductID
10							REFERENCES Northwind.dbo.Prod...
11	PRIMARY KEY (clustered)	PK_Order_Details	(n/a)	(n/a)	(n/a)	(n/a)	OrderID, ProductID

Query executed successfully. WIN-6C20A8VN2U0 (12.0 RTM) WIN-6C20A8VN2U0\MK (59) Northwind 00:00:00 1 rows

This is not a valid location for a breakpoint. Ln 1 Col 1 INS

Тип интегритет	Тип ограничение	Пояснения
Domain	DEFAULT	Специфицира стойност по подразбиране
	CHECK	Специфицира допустими стойности за колона
	FOREIGN KEY	Специфицира колоната, в която стойностите трябва да съществуват
	NULL	Специфицира дали е разрешена стойност NULL
Entity	PRIMARY KEY	Идентифицира всеки ред уникално
	UNIQUE	Предпазва от дублиране на стойностите в дадена колона (алтернативен ключ)
Referential	FOREIGN KEY	Дефинира колони, чиято стойност трябва да среща първичния ключ на същата или друга таблица
	CHECK	Специфицира разрешените стойности за колона, основани на друга колона