



# ЦЕЛОСТНОСТЬ БАЗ ДАННЫХ

*Кто отказывается от многого,  
может многое себе позволить.  
Ж. Шардон*

Технологии баз данных

---

---

---

---

---

---

---

## Содержание

2

- Понятие целостности данных
- Средства контроля целостности данных
  - Ограничения целостности
  - Утверждения
  - Триггеры

Технологии баз данных

© М.Л. Цымбалер

---

---

---

---

---

---

---

## Целостность базы данных

3

- **Целостность (data integrity)** означает точность, корректность и непротиворечивость данных, хранящихся в базе данных. Целостность часто определяют как защиту данных от санкционированных пользователей.
- Примеры ограничений целостности
  - Ограничения первичных и внешних ключей
  - Ограничения доменов
    - Цена детали >0
    - Цвет детали – только "белый", "черный" или "серый"
  - Бизнес-правила
    - Суммарный вес деталей в поставке <1500
    - Поставщик с рейтингом <5 не может поставлять детали ценой >100
- Отсутствие поддержки целостности на уровне СУБД приводит к необходимости контролировать целостность на уровне приложения.

Технологии баз данных

© М.Л. Цымбалер

---

---

---

---

---

---

---

## Средства контроля целостности

- 4
  - *Ограничения целостности* – предикаты для проверки целостности.
    - ▣ Ограничения целостности ключей
    - ▣ Ограничения целостности атрибута
    - ▣ Ограничения целостности кортежа
    - ▣ Утверждения
  - *Триггеры* – хранимые процедуры, автоматически выполняемые СУБД при наступлении определенного пользователем события (удаление/вставка/обновление кортежа, создание пользователем сессии и др.).

Технологии баз данных © М.Л. Цымбалер

## Ограничения целостности

- 5
  - *Целостность (data integrity)* означает точность и корректность данных, хранящихся в базе данных. Целостность часто определяют как защиту данных от санкционированных пользователей.
  - Примеры ограничений целостности
    - ▣ Ограничения первичных и внешних ключей
    - ▣ Ограничения доменов и атрибутов
      - Цена детали > 0
      - Цвет детали – только "белый", "черный" или "серый"
    - ▣ Бизнес-правила
      - Суммарный вес деталей в поставке < 1500
      - Поставщик с рейтингом < 5 не может поставлять детали ценой > 100
  - Отсутствие поддержки целостности на уровне СУБД приводит к необходимости контролировать целостность на уровне приложения.

Технологии баз данных © М.Л. Цымбалер

## Целостность первичных ключей

- 6
  - create table S (
 

SID char(4) **primary key,**

 Name char(10),  
 City char(10),  
 Rating int);
  - create table S (
 SID char(4),  
 Name char(10),  
 City char(10),  
 Rating int  

**primary key (SID);**

Ограничение целостности  
первичного ключа

Технологии баз данных © М.Л. Цымбалер

Целостность потенциальных ключей

7

create table S (  
SID char(4),  
Name char(10),  
City char(10),  
Rating int,  
primary key (SID),  
**unique (Name);**

create table Emp (  
EmpID int,  
Name char(20),  
AreaCode char(4),  
PhoneNo char(7),  
**unique (AreaCode, PhoneNo));**

Ограничение целостности  
потенциального ключа

---

---

---

---

---

---

---

---

Целостность внешних ключей

8

create table SP (  
SID char(4),  
PID char(4),  
Qty int not null,  
primary key (SID, PID),  
**foreign key (SID) references S (SID)  
on delete cascade\*  
on update cascade\* ,**  
**foreign key (PID) references P (PID)  
on delete cascade\*  
on update cascade\*);**

Ограничение целостности  
внешнего ключа

\*Допустимы режимы **cascade**, **set null**, **set default** и **no action**.

---

---

---

---

---

---

---

---

Ограничения доменов

9

create domain EmpID as integer  
check (value between 1 and 100000);

create domain Salary as numeric(9,2)  
default 0,00  
check ((value between 0,00 and 100000,00) and  
(value is not null));

alter domain Salary set default 1000,00;

alter domain EmpID add check (value<>13 and value<>666);

drop domain Salary restrict;

drop domain EmpID cascade;

---

---

---

---

---

---

---

---

Ограничения атрибутов

10

```
create table S (  
  SID char(4) primary key,  
  Name char(10) default 'n/a',  
  City char(10) default 'n/a',  
  Rating int default 0);
```

Ограничение целостности  
на значение по умолчанию

---

---

---

---

---

---

---

---

Ограничения атрибутов

11

```
create table S (  
  SID char(4) primary key,  
  Name char(10) check (Name not like 'Mr.%' and Name  
    not like 'Ms.%'),  
  City char(10) check (City in ('Москва', 'Челябинск',  
    'Одесса')),  
  Rating int check (Rating >= 0));
```

Ограничение целостности  
CHECK

---

---

---

---

---

---

---

---

Ограничения атрибутов

12

```
create table S (  
  SID char(4) primary key,  
  Name char(10) not null,  
  City char(10) not null,  
  Rating int not null);
```

Ограничение целостности  
not NULL

---

---

---

---

---

---

---

---

Ограничения UNIQUE и not NULL

13

- ❑ create table Emp (  
EmpID int,  
Name char(20),  
AreaCode char(4) **not null**,  
PhoneNo char(7) **not null**,  
**unique** (AreaCode, PhoneNo));

❑ Вставка кортежей  
✓❑ (1, 'Иванов', '351', '2420409')  
✗❑ (2, 'Петров', '351', NULL)  
✗❑ (21, 'Петрова', '351', NULL)  
✗❑ (3, 'Сидоров', NULL, '2420409')  
✗❑ (31, 'Сидорова', NULL, '2420409')  
✗❑ (4, 'Коньков', NULL, NULL)  
✗❑ (41, 'Конькова', NULL, NULL)  
✗❑ (5, 'Егоров', '351', '2420409')
- ❑ create table Emp (  
EmpID int,  
Name char(20),  
AreaCode char(4),  
PhoneNo char(7),  
**unique** (AreaCode, PhoneNo));

❑ Вставка кортежей  
✓❑ (1, 'Иванов', '351', '2420409')  
✓❑ (2, 'Петров', '351', NULL)  
✓❑ (21, 'Петрова', '351', NULL)  
✓❑ (3, 'Сидоров', NULL, '2420409')  
✓❑ (31, 'Сидорова', NULL, '2420409')  
✓❑ (4, 'Коньков', NULL, NULL)  
✓❑ (41, 'Конькова', NULL, NULL)  
✗❑ (5, 'Егоров', '351', '2420409')

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

---

Ограничения кортежей

14

- ❑ create table S (  
SID char(4) primary key,  
Name char(10) not null default 'n/a',  
City char(10) not null default 'n/a',  
Rating int default 0,  
**check** (City in ('Москва', 'Челябинск', 'Одесса', 'n/a')  
**and** (Rating between 0 and 30)  
**and** (Name not like 'Mr.%' or Name not like 'Ms.%'));

Ограничение целостности кортежа

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

---

Ограничения кортежей

15

- ❑ create table SP (  
SID char(4),  
PID char(4),  
Qty int not null,  
primary key (SID, PID),  
**foreign key** (SID)  
references S (SID)  
on delete cascade  
on update cascade,  
**foreign key** (PID)  
references P (PID)  
on delete cascade  
on update cascade);
- ❑ create table SP (  
SID char(4),  
PID char(4),  
Qty int not null,  
primary key (SID, PID),  
**check** (  
SID in (select SID from S) and  
PID in (select PID from P))  
);

Ограничение целостности кортежа  
НЕ может заменить ограничение  
целостности внешнего ключа

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

---

## Именованние ограничений

16

- create table S (  
SID char(4) **constraint** pk\_SID primary key,  
Name char(10) **constraint** unq\_Name unique,  
City char(10) **constraint** chk\_City check (City in  
(‘Москва’, ‘Одесса’, ‘Челябинск’, ‘n/a’)),  
Rating int **constraint** chk\_Rating (check (Rating>=0));
- alter table S **drop constraint** unq\_Name;
- alter table S **add constraint** unq\_Name unique (Name)  
exceptions into S\_badrecords;

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

## Проверка ограничений целостности

17

1. Выполнить оператор SQL\*.
2. Проверить результат выполнения на соответствие  
всем ограничениям целостности.  
Если хотя бы одно из них не выполняется,  
произвести откат.

\* При вставке строк пропущенные поля заменяются препроцессором  
на DEFAULT-значения.  
Каскадные действия рассматриваются как неотъемлемая часть выполняемого  
оператора.

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

## Утверждения целостности

18

- *Утверждение (assertion)* – задаваемый отдельной  
командой предикат целостности  
атрибута/кортежа/отношения/базы данных.
- **create assertion** asrt\_BadSuppliers **check** (  
not exists (select \* from S where S.Rating<5 and  
S.SID in (select SP.SID from SP)));
- **create assertion** asrt\_BadSuppliers2 **check** (  
not exists (select \* from S, SP  
where S.Rating<10 and S.SID=SP.SID  
and SP.Qty > 500));

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

## Немедленная и отложенная проверка ограничений

19

- create table Emp (
  - EmpID int primary key,
  - Name char(20),
  - Dept int foreign key references Dept (DeptID));
- create table Dept (
  - DeptID int primary key,
  - Name char(15),
  - EmpQty int,
  - check (EmpQty=(select count(\*) from Emp)
 where DeptID=Emp.Dept));
- Как вставить новую запись в Emp, не нарушив ограничений целостности?

Технологии баз данных © М.Л. Цымбалер

## Немедленная и отложенная проверка ограничений

20

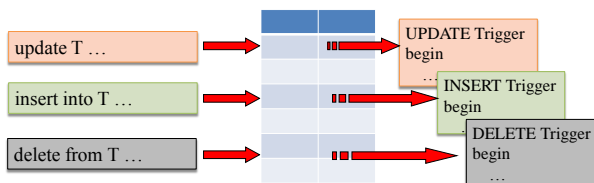
- *Немедленная проверка (immediate)* выполняется непосредственно после выполнения операции, изменяющей состояние базы данных.
- *Отложенная проверка (deferred)* выполняется после завершения *транзакции* (набора операций, которые рассматриваются как атомарная операция).
- Режимы проверки ограничения
  - initially immediate/deferred
  - initially immediate deferrable/not deferrable
- Режимы проверки ограничений в составе транзакции
  - set constraints список/all deferred/immediate

Технологии баз данных © М.Л. Цымбалер

## Триггеры

21

- *Триггеры* – хранимые процедуры, которые СУБД автоматически выполняет при наступлении события, определенного пользователем.
- Особенности:
  - Триггер не может быть выполнен явно (как хранимая процедура).
  - Триггер может быть реализован на языках SQL/PSM, Java, C.



Технологии баз данных © М.Л. Цымбалер

# Типы триггеров

23

- *Триггер строки* запускается каждый раз при наступлении указанного события.
- *Триггер события* запускается один раз при наступлении указанного события.
- *Триггер BEFORE/AFTER* запускается до/после наступления события триггера.
- *Триггер комбинированного типа* – триггер строки BEFORE, триггер события AFTER и др.
- *Триггер INSTEAD-OF* запускается вместо стандартной операции обработки указанного события.

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

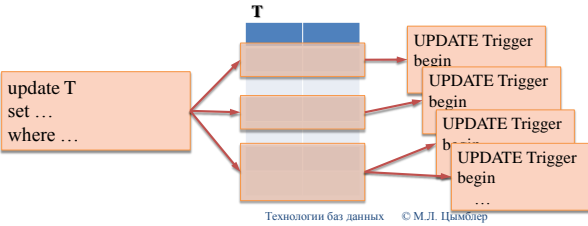
---

---

# Триггеры строки и события

23

- *Триггер строки* запускается *каждый раз* при наступлении указанного события.



Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

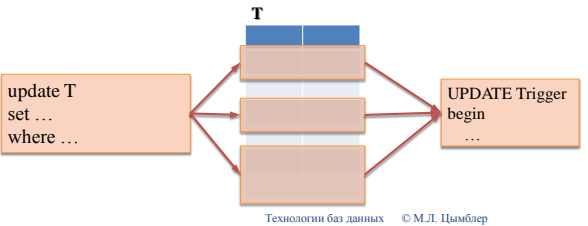
---

---

# Триггеры строки и события

24

- *Триггер события* запускается *один раз* при наступлении указанного события.



Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

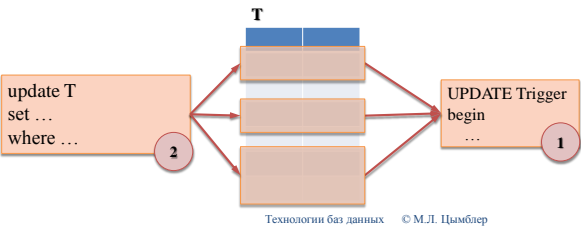
---



# Триггеры BEFORE и AFTER

25

□ Триггер *BEFORE* запускается *до* наступления события триггера.



---

---

---

---

---

---

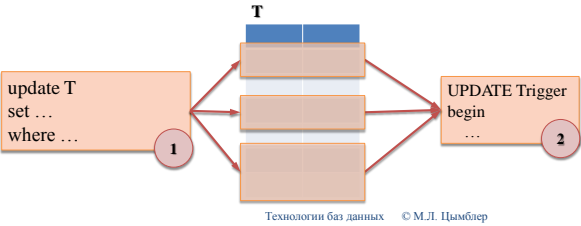
---

---

# Триггеры BEFORE и AFTER

26

□ Триггер *AFTER* запускается *после* наступления события триггера.



---

---

---

---

---

---

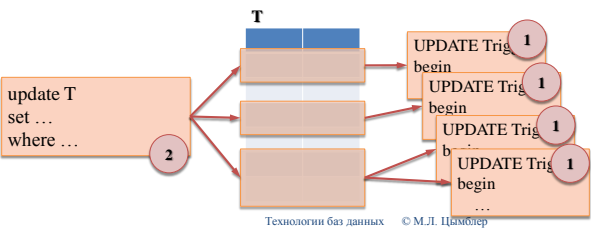
---

---

# Комбинированные триггеры

27

□ Комбинированный триггер – триггер BEFORE/AFTER строки/события.



---

---

---

---

---

---

---

---

### Триггер INSTEAD-OFF

28

□ Триггер *INSTEAD-OFF* запускается *вместо* стандартной операции обработки указанного события.

```
create view V as
select ...
from T
where ...

insert into V
values ...
```

V

T

INSTEAD-OFF Trigger  
begin  
...

Технологии баз данных

© М.Л. Цымбалер

---

---

---

---

---

---

---

---

### Триггеры

29

```
create trigger ChkWeight
before insert or update of Qty on SP
referencing new row as NewT, old row as OldT
when NewT.Qty > OldT.Qty
for each row
declare
W, Total number;
begin
select P.Weight into W
from P
where P.PID = NewT.PID;
Total = NewT.Qty * W;
if Total > 1500 then
raise_application_error('Превышен допустимый вес!');
end;
```

Событие триггера

Переменные для  
ссылки на "старый"  
и "новый" кортежи

Условие запуска  
триггера

Триггер  
строки/события

Локальные  
переменные триггера

Тело триггера

Технологии баз данных

© М.Л. Цымбалер

---

---

---

---

---

---

---

---

### Триггер INSTEAD-OF

30

□ create view ChelyabinskS as  
select SID, Name, Rating  
from S  
where City='Челябинск';

□ create trigger ChelInsert  
instead of insert on ChelyabinskS  
referencing new row as NewR  
for each row  
begin  
insert into S  
values (NewR.SID, NewR.Name, 'Челябинск',  
NewR.Rating);  
end;

Технологии баз данных

© М.Л. Цымбалер

---

---

---

---

---

---

---

---

10

## Применение триггеров

31

- Реализация сложных ограничений целостности (бизнес-правил) предметной области.
- Аудит, сбор и публикация статистики (о выполнении операторов SQL, доступе к таблицам и др.).
- Синхронная репликация таблиц.
- Модификация данных в базовых таблицах необновляемых представлений.

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

## Выполнение триггеров

32

1. Выполнить все BEFORE-триггеры события.
2. Для каждой изменяемой триггером строки
  1. Выполнить все BEFORE-триггеры строки.
  2. Заблокировать и изменить строку, выполнить проверку ограничений целостности.
  3. Выполнить все AFTER-триггеры строки.
3. Выполнить отложенную проверку декларативных ограничений целостности.
4. Выполнить все AFTER-триггеры события.

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

## Разработка триггеров

33

- Не создавать триггеры, которые
  - дублируют функциональность СУБД (декларативные ограничения целостности и др.).
  - являются рекурсивными (например, триггер AFTER INSERT выполняет оператор INSERT над той же таблицей).
  - относятся к одному событию одной и той же таблицы (порядок их выполнения неизвестен).
- Использовать хранимые процедуры для уменьшения размера триггеров.

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

Ограничения целостности vs триггеры

- ❑ Недостатки триггеров:
  - ❑ Не действует на данные, помещенные в таблицу до его создания
  - ❑ Может неявно запустить другие триггеры
    - Бесконечный вызов триггера: триггер AFTER INSERT выполняет оператор INSERT над той же таблицей.
- ❑ Недостатки ограничений целостности:
  - ❑ Не может реализовать сложные правила целостности.
- ❑ Резюме:  
Если правило целостности не удастся реализовать при помощи ограничений целостности, нужно реализовать его при помощи триггеров.

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

Задача



Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

Заключение

- ❑ Целостность – точность, корректность и непротиворечивость данных, хранящихся в базе данных.
- ❑ Средства СУБД для контроля целостности:
  - ❑ Ограничения целостности – предикаты для проверки целостности:
    - Ограничения целостности ключей
    - Ограничения целостности атрибута
    - Ограничения целостности кортежа
    - Утверждения.
  - ❑ Триггеры – хранимые процедуры, автоматически выполняемые СУБД при наступлении определенного пользователем события.

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---