

# **БЕЗОПАСНОСТЬ СИСТЕМ БАЗ ДАННЫХ**

## **6. Обеспечение целостности**

# Целостность данных

- ▣ Понятие *целостности* данных является фундаментальным в области ИБ и включает ряд специфических свойств, связанных с объектом «данные».
- ▣ В информатике "данные" понимаются как информация, представленная в формализованном виде, что обеспечивает возможность ее хранения, обработки и передачи.
- ▣ Информация на стадии данных характеризуется определенной формой и структурой.
- ▣ Под целостностью информации понимается не искаженность, достоверность, полнота, адекватность, т.е. такое ее свойство, при котором ее содержание и структура определены и изменяются только уполномоченными лицами и процессами.

# Целостность данных

- С учетом специфики используемых механизмов обеспечения целостности данных можно выделить три стороны этого понятия в отношении соответствующих видов данных в КС.
- **Правильность.** Заключается в отсутствии логических ошибок в структуре и ошибок в содержании (в значениях) данных при их обработке.
- **Неискаженность.** Заключается в отсутствии подделки данных или возникновения ошибок в данных при их передаче в линиях связи, а также при хранении в КС.
- **Неизменность.** Заключается в тождественности данных определенному эталону.

# Целостность данных

- В руководящих документах ФСТЭК России целостность информации определяется следующим образом:
- ***Целостность информации*** – это способность средства вычислительной техники или автоматизированной системы обеспечивать неизменность информации в условиях случайного и (или) преднамеренного искажения (разрушения).
- Под ***угрозой нарушения целостности*** понимается любое умышленное изменение информации, хранящейся в КС или передаваемой из одной системы в другую. Целостность также будет нарушена, если к несанкционированному изменению приводит случайная ошибка программного или аппаратного обеспечения.

# Целостность данных

- В зависимости от превалирования того или иного аспекта целостности и области использования данных выделяют:
  - специальные технологии разграничения доступа данных, обеспечивающие их целостность в моделях целостности данных;
  - криптографические технологии ЭЦП, обеспечивающие целостность данных в смысле их нескаженности при передаче в линиях связи и хранении в КС;
  - технологии параллельного выполнения транзакций в клиент-серверных системах (СУБД).

# Модели обеспечения целостности

- Формальные модели безопасности, ориентированные на обеспечения целостности, также как и формальные модели обеспечения конфиденциальности строятся на основе субъектно-объектной модели КС.
- Осуществление субъектом разрешенного доступа к объекту переводит систему в следующий момент времени в другое состояние, рассматриваемое как совокупность состояний субъектов и объектов системы.
- Проблема безопасности КС рассматривается с точки зрения разрешений запросов на доступ, при которых КС, изначально находясь в безопасном состоянии, за конечное число переходов перейдет также в безопасное состояние в смысле сохранения целостности.

# Дискреционная модель Кларка-Вильсона

Основные положения модели.

1. Все множество объектов  $D$  разделяется на объекты  $CDI$ , требующие контроля целостности (constrained data items), и объекты  $UDI$ , не требующие контроля целостности (unconstrained data items).

$$D = CDI \cup UDI, CDI \cap UDI = \emptyset.$$

2. На множестве элементарных операций над объектами выделяются совокупности (последовательности), обособляющиеся в логически самостоятельные сущности, называемые процедурами преобразования ТР (transformation procedures).

# Дискреционная модель Кларка-Вильсона

3. Дополнительно вводится особый класс процедур *IVP* над данными, которые обеспечивают проверку целостности контролируемых данных (integrity verification procedures).
  4. Те процедуры преобразования данных *TP*, применение к результатам которых процедур проверки целостности *IVP* дает положительный результат, называются «корректно сформированными транзакциями».
- Устанавливаются правила функционирования системы.

# Мандатная модель Кена Биба

Основные положения модели.

1. Система защиты представляет совокупность

- множества субъектов  $S$
- множества объектов  $O$
- множества операций над объектами доступа  $R$
- решетки уровней безопасности  $\Lambda$  субъектов и объектов
- функции  $F$ , отображающей элементы множеств  $S$  и  $O$  в  $\Lambda$
- множества состояний системы  $V$
- начального состояния  $v_0$
- набора запросов  $Q$
- функции переходов  $T: (V \times Q) \rightarrow V$ , которая переводит систему из одного состояния в другое при выполнении запросов субъектов на доступ к объектам

# Мандатная модель Кена Биба

2. Критерий безопасности (обеспечения целостности) состоит в недопустимости опасных потоков «снизу вверх».

3. Правила безопасности доступов:

- запись данных субъектом  $s \in S$  в объект  $o \in O$  возможна тогда и только тогда, когда  $F(s) \geq F(o)$ . Т.е. класс целостности субъекта должен доминировать над классом целостности объекта;
- чтение данных субъектом  $s \in S$  из объекта  $o \in O$  возможно тогда и только тогда, когда  $F(o) \geq F(s)$ . Т.е. класс целостности объекта должен доминировать над классом целостности субъекта.

# Объединение моделей Белла-ЛаПадуды и К. Биба

- В практическом плане для создания защищенных компьютерных систем (и в смысле обеспечения конфиденциальности и в смысле обеспечения целостности данных) целесообразно объединение моделей Белла-ЛаПадуды и К. Биба.
- Такое объединение может осуществляться в одном из трех вариантов.

# Объединение моделей Белла-ЛаПадуды и К. Биба

1. На основе двух различных решеток. В таких системах у субъектов и объектов доступа две метки доступа – уровень конфиденциальности, и уровень целостности.
2. На основе одной общей решетки уровней безопасности (конфиденциальности / целостности).
3. На основе одной общей решетки, но с двумя метками безопасности – по конфиденциальности и по целостности с противоположным характером их назначения / присвоения.

Третий вариант наиболее сложен, но именно он находит применение в современных ВС, в частности, в СУБД, где реализуется мандатная политика безопасности.

# Поддержание целостности данных в реляционных СУБД

- ▣ Статистика свидетельствует о том, что главными врагами БД являются не внешние злоумышленники, а ошибки оборудования, администраторов, прикладных программ и пользователей. Защиту от подобных ошибок обеспечивают механизмы поддержания целостности данных.
- ▣ Как отдельные таблицы, так и вся их совокупность (БД), обладает некоторой целостностью, которая выражается через различного рода ограничения, накладываемые на значения столбцов и связи между ними.
- ▣ При проектировании БД таблицы объединяются в целостную систему, содержащих непротиворечивые данные.

# Поддержание целостности данных в реляционных СУБД

- ▣ Применительно к реляционной БД понятие целостности включает в себя 4 аспекта:
  - семантическую целостность,
  - структурную целостность,
  - языковую целостность,
  - ссылочную целостность.
- ▣ *Семантическая целостность* БД обеспечивается на уровне ее концептуальной модели. Требования семантической целостности вытекают из понятийной модели предметной области, а именно из той ее части, которая описывает бизнес-логику или прецеденты.

# Поддержание целостности данных в реляционных СУБД

- ▣ *Структурная целостность* реляционной БД трактуется следующим образом: реляционная СУБД должна допускать работу только с однородными структурами данных типа «реляционное отношение» (реляционная таблица).
- ▣ При этом понятие «реляционного отношения» должно удовлетворять всем ограничениям, накладываемым на него в классической реляционной теории БД:
  - ▣ отсутствие множественности значений атрибутов;
  - ▣ отсутствие дубликатов и упорядоченности кортежей;
  - ▣ обязательное наличие первичного ключа.

# Поддержание целостности данных в реляционных СУБД

- ▣ *Языковая целостность* реляционной БД означает, что реляционная СУБД должна обеспечивать языки описания и манипулирования данными не ниже стандарта SQL.
- ▣ Не должны быть доступны иные низкоуровневые средства манипулирования данными, не соответствующие стандарту.
- ▣ Именно поэтому доступ к информации, хранимой в БД, и любые изменения этой информации могут быть выполнены только с использованием операторов языка SQL.

# Поддержание целостности данных в реляционных СУБД

- ▣ *Ссылочная целостность* – это ограничение БД, гарантирующее, что ссылки между данными разных таблиц являются правомерными и неповрежденными.
- ▣ Ссылочная целостность является фундаментальным принципом теории БД и проистекает из той идеи, что БД должна не только сохранить данные, но и активно содействовать обеспечению их качества.
- ▣ Ссылочная целостность в реляционной БД – это согласованность между связанными таблицами, которая поддерживается путем комбинирования первичных и внешних ключей по правилу: каждый внешний ключ должен соответствовать первичному ключу.

# Поддержание целостности данных в реляционных СУБД

- ▣ При обеспечении целостности в реляционных БД выделяют три большие категории целостности:
  - **доменная целостность** отвечают за то, чтобы в соответствующем поле БД были допустимые значения. Обеспечивается условиями на значения, запретом пустых значений, триггерами и хранимыми процедурами, а также ключами;
  - **сущностная целостность** отвечают за то, чтобы данные об одной сущности не попали в БД два раза. Обеспечивается ограничением уникальности и первичным ключом;
  - **ссылочная целостность** обеспечивается системой первичных и внешних ключей.

# Поддержание целостности данных в реляционных СУБД

- Еще две большие категории, на которые делятся средства обеспечения целостности в реляционных СУБД – это средства декларативного и процедурного характера.
- *Средства декларативного характера* создаются как составные части объектов при их определении в БД (например, условие на значение при определении таблицы в БД).
- *Средства процедурного характера* (триггеры и хранимые процедуры) реализуются как отдельные программные модули. В общем случае декларативные ограничения менее функциональны, но более экономны с точки зрения ресурсов и наоборот.

# Доменная целостность

- ▣ Ограничения на допустимые значения для столбца таблицы предназначены для поддержания *доменной целостности*.
- ▣ Они применяются для того, чтобы гарантировать, что вводимые величины попадают в допустимый интервал значений или отвечают заданному шаблону.
- ▣ В языке SQL риск нарушить доменную целостность возникает при добавлении и обновлении записей с помощью операторов INSERT и UPDATE.
- ▣ Ограничения доменной целостности можно задать при создании таблицы с помощью оператора CREATE TABLE, а также предварительно, путем создания домена, оператором CREATE DOMAIN.

# Null-значения

- Для того чтобы обойти проблему неполных или неизвестных данных в БД Эдвард Кодд предложил ввести понятие *null-значения*. Null-значение – это, собственно, не значение, а некий маркер, показывающий, что значение неизвестно.
- Если возможно появление неизвестных или неполных данных, разработчик имеет выбор из двух вариантов.
- **Первый вариант** - ограничиться использованием обычных типов данных, а вместо неизвестных данных вводить либо нулевые значения, либо значения специального вида.
- При этом не все данные становятся равноправны, и требуются дополнительный программный код, "отслеживающий" эту неравноправность.

# Null-значения

- **Второй вариант** - использовать null-значений вместо неизвестных данных. За кажущейся естественностью такого подхода скрываются глубокие проблемы.
- Наибольшей проблемой является необходимость использования трехзначной логики при оперировании с данными, которые могут содержать null-значения.
- В этом случае при неаккуратном формулировании запросов, они могут давать неправильные ответы.
- Есть и более фундаментальные проблемы, связанные с теоретическим обоснованием корректности введения null-значений, например, непонятно вообще, входят ли null-значения в домены или нет.

# Трехзначная логика (3VL)

- ▣ Так как null-значение обозначает тот факт, что значение неизвестно, то любые алгебраические операции (сложение, умножение, конкатенация строк и т.д.) должны давать также неизвестное значение, т. е. null.
- ▣ При сравнении выражений, содержащих null-значения, результат также может быть неизвестен.
- ▣ Например, значение истинности для выражения  $a = b$  есть null, если один или оба аргумента есть null.
- ▣ Таким образом, определение истинности логических выражений базируется на *трехзначной логике* (three-valued logic, 3VL), в которой кроме значений Т (true – истина), F (false – ложь), введено значение U (unknown – неизвестно).

# Трехзначная логика (3VL)

Трехзначные дизъюнкция и конъюнкция  
 $a \vee b = \max(a, b)$ ;  $a \& b = a \wedge b = \min(a, b)$

a	b	$a \vee b$	$a \& b$
0	0	0	0
0	1	1	0
0	u	u	0
1	0	1	0
1	1	1	1
1	u	1	u
u	0	u	0
u	1	1	u
u	u	u	u

# Сущностная целостность

- ▣ Требование *целостности сущностей*: каждый кортеж отношения должен отличаться от любого другого кортежа этого отношения.
- ▣ Поддержание целостности сущностей обеспечивается средствами СУБД с помощью двух ограничений:
  - при добавлении записей в таблицу проверяется уникальность их первичных ключей;
  - не допускается изменение значений атрибутов, входящих в первичный ключ.
- ▣ *Правило целостности сущностей*. Атрибуты, входящие в состав некоторого потенциального ключа не могут принимать null-значений.

# Ссылочная целостность

- ▣ *Ссылочная целостность*, синонимы: «целостность по ссылкам», «целостность связей», «целостности внешних ключей», «требование внешнего ключа».
- ▣ Сложные объекты представляются в реляционной БД в виде кортежей нескольких нормализованных отношений, связанных между собой. При этом:
  - Связи между данными отношениями описываются в терминах функциональных зависимостей.
  - Для отражения функциональных зависимостей используется дублирование первичного ключа одного отношения (родительского) в другое (дочернее). При этом копии ключей родительских отношений, называются внешними ключами.

# Ссылочная целостность

- ▣ **Правило ссылочной целостности:** Внешние ключи не должны быть несогласованными, т.е. для каждого значения внешнего ключа должно существовать соответствующее значение первичного ключа в родительском отношении.
- ▣ **Другая формулировка:** Правило ссылочной целостности состоит в том, что внешние ключи не должны ссылаться на отсутствующие в родительском отношении кортежи, т.е. внешние ключи должны быть корректны.
- ▣ Как правило, поддержание целостности ссылок также возлагается на СУБД. Например, она может не позволить пользователю добавить запись, содержащую внешний ключ с несуществующим (неопределенным) значением.

# Ссылочная целостность

- Ссылочную целостность могут нарушить операции, изменяющие состояние БД: вставки, обновления и удаления кортежей.
- Для двух отношений - родительского и дочернего и трех операций появляются шесть различных вариантов.
- Для родительского отношения:
  - *Вставка кортежа в родительское отношение.*
  - *Обновление кортежа в родительском отношении.*
  - *Удаление кортежа в родительском отношении.*
- Для дочернего отношения:
  - *Вставка кортежа в дочернее отношение.*
  - *Обновление кортежа в дочернем отношении.*
  - *Удаление кортежа в дочернем отношении.*

# Вставка кортежа в родительское отношение

Родительское отношение ЖЕНЩИНА

№ паспорта	ФИО	Дата рождения	Адрес
627519	...	...	...
627198	...	...	...
624566	...	...	...
623415	...	...	...

Вставка кортежа

Дочернее отношение БРАКИ

№ паспорта мужчины	№ паспорта женщины
...	627519
...	627198
...	624566

В родительском отношении допустимо существование кортежей, на которые нет ссылок из дочернего отношения.

**Нарушений целостности нет!**

# Обновление кортежа в родительском отношении

Родительское отношение ЖЕНЩИНА

№ паспорта	ФИО	Дата рождения	Адрес
627519	...	...	...
627198	...	...	...
624566 623793	...	...	...

Обновление кортежа

Дочернее отношение БРАКИ

№ паспорта мужчины	№ паспорта женщины
...	627519
...	627198
...	624566

При обновлении кортежа в родительском отношении может измениться значение ключа. Если в дочернем отношении есть кортежи, ссылающиеся на обновляемый кортеж, то значения их внешних ключей станут некорректными!

# Удаление кортежа в родительском отношении

Родительское отношение ЖЕНЩИНА

№ паспорта	ФИО	Дата рождения	Адрес
627519	...	...	...
627198	...	...	...
624566	...	...	...
623415	...	...	...

Удаление кортежа



Дочернее отношение БРАКИ

№ паспорта мужчины	№ паспорта женщины
...	627519
...	627198
...	624566



При удалении кортежа в родительском отношении удаляется значение ключа. Если в дочернем отношении есть кортежи, ссылающиеся на обновляемый кортеж, то значения их внешних ключей станут некорректными!

# Вставка кортежа в дочернее отношение

Нельзя вставить в дочернее отношение кортеж с некорректным значением ключа. Это может привести к нарушению ссылочной целостности!

Родительское отношение ЖЕНЩИНА

№ паспорта	ФИО	Дата рождения	Адрес
627519	...	...	...
627198	...	...	...
624566	...	...	...
623415	...	...	...

Дочернее отношение БРАКИ

№ паспорта мужчины	№ паспорта женщины
...	627519
...	627198
...	624566
...	626827

Вставка кортежа

# Обновление кортежа в дочернем отношении

Обновление кортежа в дочернем отношении с некорректным изменением ключа **может привести к нарушению ссылочной целостности!**

Родительское отношение ЖЕНЩИНА

№ паспорта	ФИО	Дата рождения	Адрес
627519	...	...	...
627198	...	...	...
624566	...	...	...
623415	...	...	...

Дочернее отношение БРАКИ

№ паспорта мужчины	№ паспорта женщины
...	627519
...	627198
...	624566
...	628425

Обновление кортежа

# Удаление кортежа в дочернем отношении

При удалении кортежа в дочернем отношении  
ссылочная **целостность**  
не нарушается!

Родительское отношение ЖЕНЩИНА

№ паспорта	ФИО	Дата рождения	Адрес
627519	...	...	...
627198	...	...	...
624566	...	...	...
623415	...	...	...

Дочернее отношение БРАКИ

№ паспорта мужчины	№ паспорта женщины
...	627519
...	627198
...	624566

Удаление кортежа

# Ссылочная целостность

- Анализ всех шести вариантов показывает, что ссылочная целостность в принципе может быть нарушена при выполнении одной из четырех операций:
  - *обновление кортежа в родительском отношении;*
  - *удаление кортежа в родительском отношении;*
  - *вставка кортежа в дочернее отношение;*
  - *обновление кортежа в дочернем отношении.*

# Стратегии поддержания ссылочной целостности

- Для поддержания ссылочной целостности обычно используются две основные стратегии:
  - **RESTRICT (ОГРАНИЧИТЬ)** - не разрешать выполнение операции, приводящей к нарушению ссылочной целостности .
  - **CASCADE (КАСКАДИРОВАТЬ)** - разрешить выполнение требуемой операции, но внести каскадные изменения в другие отношения так, чтобы не допустить нарушения ссылочной целостности.

# Стратегии поддержания ссылочной целостности

- ▣ Дополнительные стратегиями являются:
  - **SET NULL (УСТАНОВИТЬ В NULL)** - все некорректные значения внешних ключей изменять на null-значения.
  - **SET DEFAULT (УСТАНОВИТЬ ПО УМОЛЧАНИЮ)** - все некорректные значения внешних ключей изменять на некоторое значение, принятое по умолчанию.
- ▣ Можно также отказаться от использования какой-либо стратегии поддержания ссылочной целостности:
  - **IGNORE (ИГНОРИРОВАТЬ)** – т.е. выполнять операции, не обращая внимания на нарушения ссылочной целостности.

# Способы поддержания целостности

- ▣ Способы (средства) поддержания целостности делятся на декларативные - *ограничения* и процедурные - *правила*.
- ▣ *Ограничения* обеспечивают проверку и выполнение ряда описательно заданных правил-ограничений, часто называемых «бизнес-правилами» (Business Rules).
- ▣ Ограничение представляет собой правило, которому должны подчиняться данные, вводимые пользователями в таблицу.
- ▣ Ограничения могут относиться к таблицам или столбцам и являются частью определения таблицы. Их можно накладывать при создании таблицы или же добавлять позднее. Можно удалять или приостанавливать действие ранее наложенных ограничений.

# Виды ограничений целостности

- ▣ Выделяются следующие виды декларативных ограничений :
  - ▣ ограничения целостности на уровне атрибута;
  - ▣ ограничения целостности на уровне домена;
  - ▣ ограничения целостности на уровне отношения;
  - ▣ ограничения целостности на уровне связи между отношениями.

# Ограничения целостности на уровне атрибута

- Применяются только к одному атрибуту (столбцу) и их можно вводить как часть определения столбца.
- Ограничения на атрибуты задаются при создании таблицы, в операторах `CREATE TABLE`.
- Ограничения на атрибуты могут быть выполнены следующими способами:
  - задание значения по умолчанию;
  - задание обязательности или необязательности значений (Null);
  - задание условий на значения атрибутов.

# Ограничения целостности на уровне домена

- ▣ Некоторые СУБД поддерживают доменную структуру, то есть разрешают определять отдельно домены, задавать тип данных для каждого домена и задавать соответственно ограничения в виде бизнес-правил для доменов.
- ▣ Ограничения целостности на уровне доменов удобны, если в БД присутствуют несколько столбцов разных отношений, которые принимают значения из одного и того же множества допустимых значений.
- ▣ В этом случае действительно удобно задать ограничение на значение прямо на уровне домена, тогда оно автоматически будет выполняться для всех атрибутов, которые принимают значения из этого домена

# Ограничения целостности на уровне отношения

- Некоторые ограничения невозможно преобразовать в выражения, которые будут применимы только к одному столбцу.
- Допустим, требуется, чтобы пользователь оставлял № хотя бы одного из двух телефонов для связи.
- Для MS Access или MS SQL Server соответствующее проверочное выражение будет выглядеть следующим образом: HOME\_PHON IS NOT NULL OR WORK\_PHON IS NOT NULL
- Ограничения целостности на уровне отношения (таблицы) могут задаваться как при создании таблицы, так и позже, посредством оператора ALTER TABLE.

# Ограничения целостности на уровне связи между отношениями

- ▣ Это рассмотренная выше ссылочная целостность.
- ▣ Ссылочные ограничения подразумевают задание обязательности связи, принципов каскадного удаления и каскадного изменения данных, задание поддержки ограничений по мощности связи.
- ▣ Эти виды ограничений могут быть выражены заданием обязательности или необязательности значений внешних ключей во взаимосвязанных отношениях.

# Наложение ограничений целостности

- Ограничения всех видов накладываются владельцем таблицы и влияют на исход последующих операций с данными.
- Перед завершением выполнения SQL-оператора производится проверка имеющихся ограничений. При обнаружении нарушений СУБД сигнализирует о ненормальном завершении и аннулирует внесенные оператором изменения.
- Следует отметить, что для наложения ссылочного ограничения необходимо обладать привилегией REFERENCES по отношению к таблице, на которую делается ссылка.

# Снятие и приостановка ограничений целостности

- Ограничения можно не только накладывать, но и отменять. При этом между ограничениями могут существовать зависимости, и отмена одного из них может потребовать ликвидации других (ссылочных) ограничений, зависящих от первоначального.
- Можно временно приостановить действие ограничений.
- Чтобы «примирить» контроль ограничений целостности и эффективность функционирования АИС при массовом копировании данных контроль ограничений обычно отключается. Это значит, что массовое копирование необходимо дополнять запуском процедуры глобальной проверки целостности.

# Процедурные способы обеспечения целостности

- Не все ограничения целостности можно реализовать декларативно. Соблюдение некоторых из них требует проверки определенных условий Отсюда их название *процедурные ограничения целостности* или *правила*.
- Правила позволяют вызывать выполнение заданных действий при определенных изменениях БД. Правила ассоциируются с таблицами и срабатывают при изменении этих таблиц.
- В отличие от ограничений, которые являются лишь средством контроля относительно простых условий, правила позволяют проверять и поддерживать сколь угодно сложные соотношения между элементами данных в БД.

# Триггеры

- ▣ Процедурные ограничения целостности (правила) реализуются с помощью *триггеров* (англ. trigger) или хранимых процедур.
- ▣ Понятие «триггера» ввела фирма «Say Base» в 1986 году.
- ▣ *Триггер* – это программа особого типа, хранящаяся на сервере БД, которую пользователь не вызывает непосредственно. Ее исполнение обусловлено наступлением определенного события, как правило, это события вставки строки (INSERT), удаление строки (DELETE) или модификация данных в строке (UPDATE) в заданном столбце заданной таблицы реляционной БД.

# Триггеры

- ▣ Триггеры используются для проверки целостности данных, а также для отката транзакций. Они обеспечивают целостность, выполняя комплексную межтабличную проверку допустимости данных вне контекста справочной целостности и ограничений проверки.
- ▣ Наиболее важное назначение триггеров – определение глобального делового правила.
- ▣ Например, триггер может использоваться для обеспечения того, чтобы менеджеру службы продаж посылалось сообщение о недостаточном количестве товара на складе и необходимости пополнения запаса, если общее количество товара на складе стало меньше чем зарезервированное.

# Триггеры

- При добавлении новых строк в одну таблицу в целях обеспечения целостности данных может потребоваться соответствующим образом добавлять или изменять строки других, связанных, таблиц.
- **Пример.** Имеются данные по кафедрам и сотрудникам, работающим на них

Ключ	Название	Количество
1	Кафедра алгебры	3
2	Кафедра программирования	2

Ключ	ФИО	Кафедра
1	Иванов	1
2	Петров	2
3	Сидоров	1
4	Пушников	2
5	Шарипов	1

# Триггеры

- Требуется, чтобы поле «Количество» таблицы КАФЕДРЫ содержало всегда реальное количество сотрудников, числящихся в подразделении.
- Для реализации этого ограничения необходимо создать триггер, запускающийся при вставке, модификации и удалении записей в таблице СОТРУДНИКИ, который будет корректно изменять значение поля «Количество».
- Например, при вставке в таблицу СОТРУДНИКИ новой строки, триггер увеличивает на единицу значение поля Количество, а при удалении строки – уменьшает значение этого поля.

# Триггеры

- Триггер – это хранимая процедура, которая автоматически вызывается при выполнении того или иного действия с таблицей, направленного на изменение данных.
- Явно триггер вызываться не может. При создании триггера указывается, к какой таблице и к какому действию (UPDATE, INSERT или DELETE) он привязывается.
- Триггер может вызываться один раз при выполнении всего запроса, либо каждый раз, когда необходимо обновить, удалить или вставить очередную строку в таблице (триггер FOR EACH ROW).
- Причем можно указать, когда должен вызываться триггер: до операции (триггер BEFORE) или после (триггер AFTER).

# Триггеры

- Как и в случае ограничений, проверка правил отключается при массовых операциях копирования.
- Админ БД может также явным образом отменить проверку правил, воспользовавшись оператором `SET NORULES`. Оператор `SET RULES` позволяет затем восстановить работу механизма правил. По умолчанию этот механизм включен.
- Для удаления правил служит оператор `DROP RULE` правило.
- СУБД обеспечивает автоматическое удаление правил в тех случаях, когда удаляется соответствующая таблица.

# Триггеры

- Использование триггеров имеет и свои **недостатки**:
  - сложность БД, когда деловые правила становятся частью БД;
  - скрытость правил – логика срабатывания триггеров бывает не всегда понятной, особенно для программистов, которые приходят на стадии модернизации программы и им приходится разбираться в чужой программе и логике.

# Отладка триггеров

- В СУБД ЛИНТЕР для упрощения процесса разработки и отладки хранимых процедур и триггеров ядро СУБД содержит специальную утилиту-отладчик.
- Это позволяет отлаживать процедуры в реальных рабочих условиях (на сервере), в среде запросов и действий конкретного реального приложения.
- Для того чтобы начать отладку, пользователь открывает так называемую «отладочную сессию» для процедуры, после чего ядро работает с данной процедурой в режиме отладки, реагируя на команды отладчика и передавая ему всю необходимую информацию.

# Отладка триггеров

- Можно отлаживать лишь те процедуры, которые оттранслированы вместе с отладочной информацией.
- На необходимость включения отладочной информации указывает ключевая фраза `FOR DEBUG` в заголовке процедуры или триггера.
- Для начала отладки конкретной исполняемой процедуры необходимо либо запустить процедуру из-под отладчика, либо ждать, пока какое-нибудь приложение не вызовет эту процедуру или какой-нибудь запрос не запустит триггер.
- Утилита предоставляет мощную возможность по отладке процедур и триггеров в рамках их «родной» среды, когда они запущены в штатном порядке.

# Применение ограничений и правил

- Что целесообразнее применять – ограничения или правила – триггеры?
- Если ограничение в состоянии решить стоящую проблему, то следует использовать именно его. Если для проверки целостности требуется более сложная логика, то нужен триггер.
- Базовые различия между ограничениями и триггерами:
- Ограничения относятся к профилактическим мерам, то есть, они предотвращают наступление нежелательных событий. Триггеры же являются реакциями на наступление таких событий и устраняют нанесенный урон.

# Применение ограничений и правил

- Ограничения относятся к текущей таблице, а триггеры могут обращаться и к другим таблицам, даже находящимся в других БД или на других серверах.
- Выполняемое триггером действие нужно запрограммировать. Действия ограничений определяются ядром СУБД.
- При нарушении ограничения система выводит только зашифрованное сообщение об ошибке.
- Триггеры позволяют формировать в сообщении дополнительные комментарии об ошибках помимо предусмотренных в КС.

# Применение ограничений и правил

- ▣ В контексте информационной безопасности важно отметить, что создать правило, ассоциируемое с таблицей, может владелец этой таблицы, имеющий право на выполнение соответствующей процедуры.
- ▣ Пользователь, действия которого вызывают срабатывание правила, должен обладать лишь необходимыми правами доступа к таблице.
- ▣ Тем самым **правила неявно расширяют привилегии пользователей**. Подобные расширения нуждаются в строгом административном контроле, поскольку могут серьезно повлиять на защищенность данных.

# Применение ограничений и правил

- Таким образом, обеспечить целостность можно разными методами. В конкретных ситуациях при выборе предпочтительного метода следует руководствоваться правилом: если ограничение в состоянии обеспечить необходимый уровень целостности, то следует использовать именно его
- Ограничения соответствуют стандарту ANSI SQL, поэтому корпорация Microsoft рекомендует по возможности использовать именно ограничения.