Лекция 2. Средства определения базовых таблиц и ограничений целостности

- Введение
- Средства определения, изменения и ликвидации базовых таблиц
  - определение базовой таблицы
  - определение табличного ограничения
- Средства определения и отмены общих ограничений целостности
  - определение общих ограничений целостности
  - отмена определения общего ограничения целостности
  - немедленная и откладываемая проверка ограничений
- Заключение

Введение (1)

- Предыдущая лекция посвящалась тому, что (т. е. данные каких типов) может храниться в SQL-ориентированной базе данных
- Теперь следует понять, где хранятся эти данные
- Как и в реляционной модели данных, в модели SQL поддерживается единственная родовая структура данных, называемая в данном случае *базовой таблицей*
- В первом из двух основных разделов лекции обсуждаются средства языка SQL, предназначенные для определения, изменения определения и отмены определения базовых таблиц

Введение (2)

- Понятие базовой таблицы родственно понятию отношения: можно считать, что базовая таблица обладает заголовком, в котором содержатся различаемые имена столбцов и их типы данных
  - заголовок базовой таблицы является множеством и представляет собой близкий аналог заголовка отношения,
- и телом, включающим строки, которые соответствуют заголовку таблицы
  - казалось бы, здесь мы имеем аналоги тела отношения и кортежей
- Но коренное отличие базовой таблицы от истинного отношения состоит в том, что тело таблицы не обязательно является множеством
- Среди строк тела таблицы могут встречаться дубликаты, и в общем случае тело базовой таблицы SQL представляет собой мультимножество строк

Средства определения, изменения и ликвидации базовых таблиц (1)

- Базовые (реально хранимые в базе данных) таблицы создаются (определяются) с использованием оператора CREATE TABLE
- Для изменения определения базовой таблицы применяется оператор ALTER TABLE
- Уничтожить хранимую таблицу (отменить ее определение) можно с помощью оператора DROP TABLE

Средства определения, изменения и ликвидации базовых таблиц (2)

- Хотя внешне операторы CREATE TABLE, ALTER TABLE и DROP TABLE похожи на соответствующие операторы определения, изменения определения и отмены определения домена, между ними имеется принципиальное различие
  - Определение домена приводит всего лишь к созданию некоторых новых описателей, входящих в состав метаданных базы данных
  - Создание базовой таблицы, кроме создания соответствующих описателей, порождает новую область внешней памяти, в которой будут храниться данные, поставляемые пользователями
- Тем самым, базовая таблица SQL-ориентированной базы данных является прямым аналогом переменной отношения реляционной модели данных

Средства определения, изменения и ликвидации базовых таблиц (3)

- **Определение базовой таблицы**
- Оператор создания базовой таблицы CREATE TABLE имеет следующий синтаксис:
  base_table_definition ::= CREATE TABLE base_table_name (base_table_element_commalist)
  base_table_element ::= column_definition | base_table_constraint_definition
- Здесь base_table_name задает имя новой (изначально пустой) базовой таблицы
- Каждый элемент определения базовой таблицы является
  - либо определением столбца,
  - либо определением табличного ограничения целостности

Средства определения, изменения и ликвидации базовых таблиц (4)

- **Определение столбца**
- Элемент определения столбца специфицируется на основе следующих синтаксических правил:
  column_definition ::=
  column_name { data_type | domain_name }
  [default_definition]
  [column_constraint_definition_list]
- В элементе определения столбца column_name задает имя определяемого столбца
- Тип столбца специфицируется путем явного указания типа данных (data_type) или путем указания имени ранее определенного домена (domain_name)

Средства определения, изменения и ликвидации базовых таблиц (5)

- **Значения столбца по умолчанию**
- Необязательный раздел определения значения столбца по умолчанию имеет тот же синтаксис, что и раздел определения значения по умолчанию в операторах определения или изменения определения домена: DEFAULT { literal | niladic_function | NULL }
- Действующее значение по умолчанию для данного столбца определяется следующим образом:
  - если в определении столбца явно присутствует раздел DEFAULT, то значением столбца по умолчанию является значение, указанное в этом разделе;
  - иначе, если столбец определяется на домене и в определении этого домена явно присутствует раздел DEFAULT, то значением столбца по умолчанию является значение, указанное в этом разделе;
  - иначе значением по умолчанию столбца является NULL

Средства определения, изменения и ликвидации базовых таблиц (5)

- Если значением по умолчанию неявно объявлено неопределенное значение (NULL), но среди ограничений целостности столбца присутствует ограничение NOT NULL, то считается,
  - что у столбца вообще отсутствует значение по умолчанию
- Это означает, что при любой вставке новой строки в соответствующую базовую таблицу значение данного столбца должно быть задано явно

Средства определения, изменения и ликвидации базовых таблиц (6)

- **Ограничения целостности столбца**
- Элемент необязательного списка ограничений целостности столбца определяется следующими синтаксическими правилами:
  column_constraint_definition ::=
  [CONSTRAINT constraint_name]
  NOT NULL |
  { PRIMARY KEY | UNIQUE } |
  references_definition |
  CHECK (conditional_expression)
- Любое ограничение целостности на уровне столбца может быть эквивалентным образом выражено в виде табличного ограничения целостности
- Единственный резон определения ограничений на уровне столбца состоит в том, что в этом случае в ограничении целостности не требуется явно указывать имя столбца

Средства определения, изменения и ликвидации базовых таблиц (7)

- Ограничение NOT NULL означает, что в определяемом столбце никогда не должны содержаться неопределенные значения
- Если определяемый столбец имеет имя C, то это ограничение эквивалентно следующему табличному ограничению: CHECK (C IS NOT NULL)

Средства определения, изменения и ликвидации базовых таблиц (8)

- В определение столбца может входить одно из ограничений: ограничение первичного ключа (PRIMARY KEY) или ограничение возможного ключа (UNIQUE)
- Включение в определение столбца любого из этих ограничений означает требование уникальности значений определяемого столбца
  - т.е. во все время существования определяемой таблицы во всех ее строках значения данного столбца должны быть различны
- Ограничение PRIMARY KEY, в дополнение к этому, влечет ограничение NOT NULL для определяемого столбца
- Эти ограничения столбца эквивалентны следующим табличным ограничениям: PRIMARY KEY (C) и UNIQUE (C)
- В случае UNIQUE SQL опирается на семантику неопределенных значений, отличную от семантики, используемой в большинстве других случаев
  - здесь считается, что (NULL = NULL) = *true* и что (a = NULL) = (NULL = a) = *false* для любого значения a, отличного от NULL

Средства определения, изменения и ликвидации базовых таблиц (9)

- Ограничение references_definition означает объявление определяемого столбца внешним ключом таблицы и обладает следующим синтаксисом:

  references_definition ::=
  REFERENCES base_table_name [column_commalist)]
  [MATCH { SIMPLE | FULL | PARTIAL }]
  [ON DELETE referential_action]
  [ON UPDATE referential_action]
- Эта синтаксическая конструкция работает и в случае определения внешнего ключа на уровне таблицы
  - в одном из определений табличных ограничений целостности

Средства определения, изменения и ликвидации базовых таблиц (10)

- Отложим обсуждение до рассмотрения этого общего случая
- Заметим только то, что при использовании конструкции на уровне определения столбца column_commalist может содержать имя только одного столбца
  - потому что внешний ключ состоит из одного определяемого столбца
- Ограничение эквивалентно следующему табличному ограничению:
  FOREGN KEY (C) references_definition

Средства определения, изменения и ликвидации базовых таблиц (11)

- Определение проверочного ограничения
  - CHECK (conditional_expression)

  приводит к тому, что в данном столбце могут находиться только те значения, для которых вычисление conditional_expression приводит к результату *true*
- В условном выражении проверочного ограничения столбца разрешается использовать имя только определяемого столбца
- Проверочное ограничение столбца может быть без всяких изменений перенесено на уровень определения табличных ограничений

Средства определения, изменения и ликвидации базовых таблиц (12)

- **Определение табличного ограничения**
- Элемент определения табличного ограничения целостности задается в следующем синтаксисе:
  base_table_constraint_definition ::=
  [CONSTRAINT constraint_name]
  { PRIMARY KEY | UNIQUE } (column_commalist)
  | FOREIGN KEY (column_commalist) references_definition
  | CHECK (conditional_expression)
- Имеются три разновидности табличных ограничений: ограничение первичного или возможного ключа (PRIMARY KEY или UNIQUE), ограничение внешнего ключа (FOREGN KEY) и проверочное ограничение (CHECK)
- Любому ограничению может явным образом назначаться имя, если перед определением ограничения поместить конструкцию CONSTRAINT constraint_name

Средства определения, изменения и ликвидации базовых таблиц (12)

- Наличие табличного ограничения первичного или возможного ключа
  - { PRIMARY_KEY | UNIQUE } (column_commalist)

  означает требование уникальности составных значений указанной группы столбцов
  - т.е. во все время существования определяемой таблицы во всех ее строках составные значения данной группы столбцов должны быть различны
- Ограничение PRIMARY KEY, в дополнение к этому, влечет ограничение NOT NULL для всех столбцов, упоминаемых в определении ограничения

Средства определения, изменения и ликвидации базовых таблиц (13)

- В определении таблицы допускается произвольное число определений возможного ключа
  - для разных комбинаций столбцов,

  но не более одного определения первичного ключа
- В языке SQL действительно допускается определение таблиц, у которых отсутствуют возможные ключи
  - эта особенность языка, среди прочего, очевидным образом противоречит базовым требованиям реляционной модели данных

Средства определения, изменения и ликвидации базовых таблиц (14)

- Определение табличного ограничения вида CHECK (conditional_expression) приводит к тому, что указанное условное выражение будет вычисляться при каждой попытке обновления соответствующей таблицы
  - вставке новой строки, удалении или модификации существующей строки
- Считается, что попытка обновления таблицы нарушает проверочное ограничение целостности, если после выполнения операции обновления вычисление условного выражения дает результат *false*
- Другими словами, таблица находится в соответствии с данным проверочным табличным ограничением, если для всех строк таблицы результатом вычисления соответствующего условного выражения не является *false*
  - про условные выражение позже

Средства определения, изменения и ликвидации базовых таблиц (15)

- **Табличное ограничение внешнего ключа**
- Синтаксис и семантика определения внешнего ключа в операторе SQL определения базовой таблицы являются довольно запутанными и сложными
- Табличное ограничение
  FOREIGN KEY (column_commalist) references_definition
  означает объявление внешним ключом группы столбцов, имена которых перечислены в списке column_commalist
- Обсудим смысл ограничения внешнего ключа при разных вариантах формирования определения ссылок (references_definition)

Средства определения, изменения и ликвидации базовых таблиц (16)

- **Табличное ограничение внешнего ключа**
- Синтаксис и семантика определения внешнего ключа в операторе SQL определения базовой таблицы являются довольно запутанными и сложными
- Табличное ограничение
  FOREIGN KEY (column_commalist) references_definition
  означает объявление внешним ключом группы столбцов, имена которых перечислены в списке column_commalist
- Обсудим смысл ограничения внешнего ключа при разных вариантах формирования определения ссылок (references_definition)

Средства определения, изменения и ликвидации базовых таблиц (17)

- Повторим синтаксическое правило:
- references_definition ::=
  REFERENCES base_table_name
  [column_commalist)]
  [MATCH { SIMPLE | FULL | PARTIAL }]
  [ON DELETE referential_action]
  [ON UPDATE referential_action]
- В этом определении base_table_name должно представлять собой имя некоторой базовой таблицы
  - пусть, например, эта таблица имеет имя *T*

Средства определения, изменения и ликвидации базовых таблиц (18)

- Если определение ссылок включает список столбцов (column_commalist), то этот список должен совпадать
  - с точностью до порядка следования имен столбцов
- со списком имен столбцов, использованных в некотором определении первичного или возможного ключа (PRIMARY_KEY или UNIQUE) в определении таблицы *T*
- Если в определении ссылок список столбцов явно не задан, то считается, что он совпадает со списком столбцов, использованных в определении первичного ключа (PRIMARY_KEY) таблицы *T*

Средства определения, изменения и ликвидации базовых таблиц (19)

- Пусть определяемая таблица имеет имя *S*
- Обсудим смысл необязательного раздела определения внешнего ключа MATCH { SIMPLE | FULL | PARTIAL }
- Если этот раздел отсутствует или если присутствует и имеет вид MATCH SIMPLE, то ограничение внешнего ключа (ссылочное ограничение) удовлетворяется в том и только в том случае, когда для каждой строки таблицы *S*
  - либо *какой-либо* столбец, входящий в состав внешнего ключа, содержит NULL,
  - либо таблица *T* содержит в *точности одну строку* такую, что значение внешнего ключа в данной строке таблицы *S* совпадает со значением соответствующего возможного ключа в этой строке таблицы *T*

Средства определения, изменения и ликвидации базовых таблиц (20)

- Если раздел MATCH присутствует в определении внешнего ключа и имеет вид MATCH PARTIAL, то ограничение внешнего ключа удовлетворяется в том и только в том случае, когда для каждой строки таблицы S
  - либо *каждый* столбец, входящий в состав внешнего ключа, содержит NULL,
  - либо таблица T содержит, *по крайней мере*, одну такую строку, что для каждого столбца данной строки таблицы S, значение которого отлично от NULL, его значение совпадает со значением соответствующего столбца возможного ключа в этой строке таблицы T

Средства определения, изменения и ликвидации базовых таблиц (21)

- Если раздел MATCH имеет вид MATCH FULL, то ограничение внешнего ключа удовлетворяется в том и только в том случае, когда
  - для каждой строки таблицы *S* либо *каждый* столбец, входящий в состав внешнего ключа, содержит NULL,
  - либо *ни один* столбец, входящий в состав внешнего ключа, не содержит NULL, и таблица *T* содержит ***в точности одну строку*** такую, что значение внешнего ключа в данной строке таблицы *S* совпадает со значением соответствующего возможного ключа в этой строке таблицы *T*
- Легко видеть, что только при наличии спецификации MATCH FULL ссылочное ограничение соответствует требованиям реляционной модели
- Тем не менее, в определении ограничения внешнего ключа базовых таблиц в SQL по умолчанию предполагается наличие спецификации MATCH SIMPLE

Средства определения, изменения и ликвидации базовых таблиц (22)

- В связи с определением ограничения внешнего ключа нам осталось обсудить еще два необязательных раздела
  - ON DELETE referential_action и
  - ON UPDATE referential_action
- Прежде всего, приведем синтаксическое правило:

  referential_action ::=
  { NO ACTION | RESTRICT | CASCADE |
  SET DEFAULT | SET NULL }
- Чтобы объяснить, в каких случаях, и каким образом выполняются эти действия, требуется сначала определить понятие *ссылающейся строки (referencing row)*

Средства определения, изменения и ликвидации базовых таблиц (23)

- Если в определении ограничения внешнего ключа отсутствует раздел MATCH или присутствуют спецификации MATCH SIMPLE или MATCH FULL, то для данной строки *t* таблицы *T* строкой таблицы *S*, ссылающейся на строку *t*, называется
  - каждая строка таблицы *S*, значение внешнего ключа которой совпадает со значением соответствующего возможного ключа строки *t*

Средства определения, изменения и ликвидации базовых таблиц (24)

- Если в определении ограничения внешнего ключа присутствует спецификация MATCH PARTIAL, то для данной строки *t* таблицы *T* строкой таблицы *S*, ссылающейся на строку *t*, называется
  - каждая строка таблицы *S*, отличные от NULL значения столбцов внешнего ключа которой совпадают со значениями соответствующих столбцов соответствующего возможного ключа строки *t*
- В случае MATCH PARTIAL строка таблицы *S* называется *ссылающейся исключительно на строку t* таблицы *T*,
  - если эта строка таблицы S является ссылающейся на строку *t* и не является ссылающейся на какую-либо другую строку таблицы *T*

Средства определения, изменения и ликвидации базовых таблиц (24)

- Если в определении ограничения внешнего ключа присутствует спецификация MATCH PARTIAL, то для данной строки *t* таблицы *T* строкой таблицы *S*, ссылающейся на строку *t*, называется
  - каждая строка таблицы *S*, отличные от NULL значения столбцов внешнего ключа которой совпадают со значениями соответствующих столбцов соответствующего возможного ключа строки *t*
- В случае MATCH PARTIAL строка таблицы *S* называется *ссылающейся исключительно на строку t* таблицы *T*,
  - если эта строка таблицы S является ссылающейся на строку *t* и не является ссылающейся на какую-либо другую строку таблицы *T*

Средства определения, изменения и ликвидации базовых таблиц (25)

- Пусть определение ограничения внешнего ключа содержит раздел ON DELETE referential_action
- Предположим, что предпринимается попытка удалить строку *t* из таблицы *T*
- Тогда, если в качестве требуемого ссылочного действия указано NO ACTION или RESTRICT, то
  - операция удаления отвергается, если ее выполнение вызвало бы нарушение ограничения внешнего ключа
- Если в качестве требуемого ссылочного действия указано CASCADE, то
  - строка *t* удаляется,

  и если в определении ограничения внешнего ключа отсутствует раздел MATCH или присутствуют спецификации MATCH SIMPLE или MATCH FULL, то
  - удаляются все строки, ссылающиеся на *t*
- Если же в определении ограничения внешнего ключа присутствует спецификация MATCH PARTIAL, то
  - удаляются только те строки, которые ссылаются исключительно на строку *t*

Средства определения, изменения и ликвидации базовых таблиц (26)

- Если в качестве требуемого ссылочного действия указано SET DEFAULT, то
  - строка *t* удаляется,

  и во всех столбцах, которые входят в состав внешнего ключа, всех строк, ссылающихся на строку *t*, проставляется заданное при их определении значение по умолчанию
- Если в определении внешнего ключа содержится спецификация MATCH PARTIAL, то подобному воздействию подвергаются только те строки таблицы S, которые ссылаются исключительно на строку *t*
- Если в качестве требуемого ссылочного действия указано SET NULL, то
  - строка *t* удаляется,

  и во всех столбцах, которые входят в состав внешнего ключа, всех строк, ссылающихся на строку

Средства определения, изменения и ликвидации базовых таблиц (27)

- Пусть определение ограничения внешнего ключа содержит раздел ON UPDATE referential_action
- Предположим, что предпринимается попытка обновить столбцы соответствующего возможного ключа в строке *t* из таблицы *T*
- Тогда если в качестве требуемого ссылочного действия указано NO ACTION или RESTRICT, то
  - операция обновления отвергается, если ее выполнение вызвало бы нарушение ограничения внешнего ключа
- Если в качестве требуемого ссылочного действия указано CASCADE, то
  - строка *t* обновляется,
- и если в определении ограничения внешнего ключа отсутствует раздел MATCH или присутствуют спецификации MATCH SIMPLE или MATCH FULL, то
  - соответствующим образом обновляются все строки, ссылающиеся на *t*
    - в них должным образом изменяются значения столбцов, входящих в состав внешнего ключа
- Если же в определении ограничения внешнего ключа присутствует спецификация MATCH PARTIAL, то обновляются только те строки, которые ссылаются исключительно на строку *t*

Средства определения, изменения и ликвидации базовых таблиц (28)

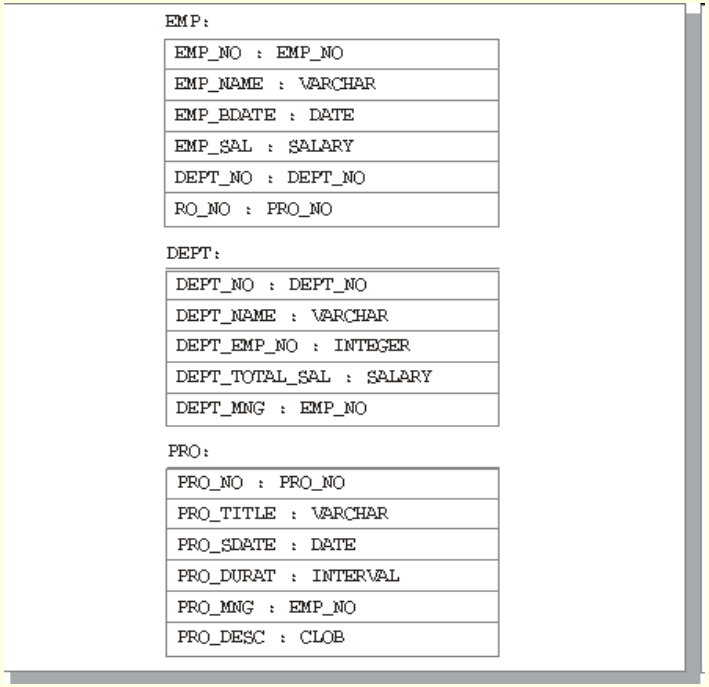
- Если в качестве требуемого ссылочного действия указано SET DEFAULT, то
  - строка *t* обновляется, и
  - во всех столбцах, которые входят в состав внешнего ключа и соответствуют изменяемым столбцам таблицы *T*, всех строк, ссылающихся на строку *t*, проставляется заданное при их определении значение по умолчанию
- Если в определении внешнего ключа содержится спецификация MATCH PARTIAL, то подобному воздействию подвергаются только те строки таблицы *S*, которые
  - ссылаются исключительно на строку *t*, причем в них изменяются значения только тех столбцов, которые не содержали NULL

Средства определения, изменения и ликвидации базовых таблиц (29)

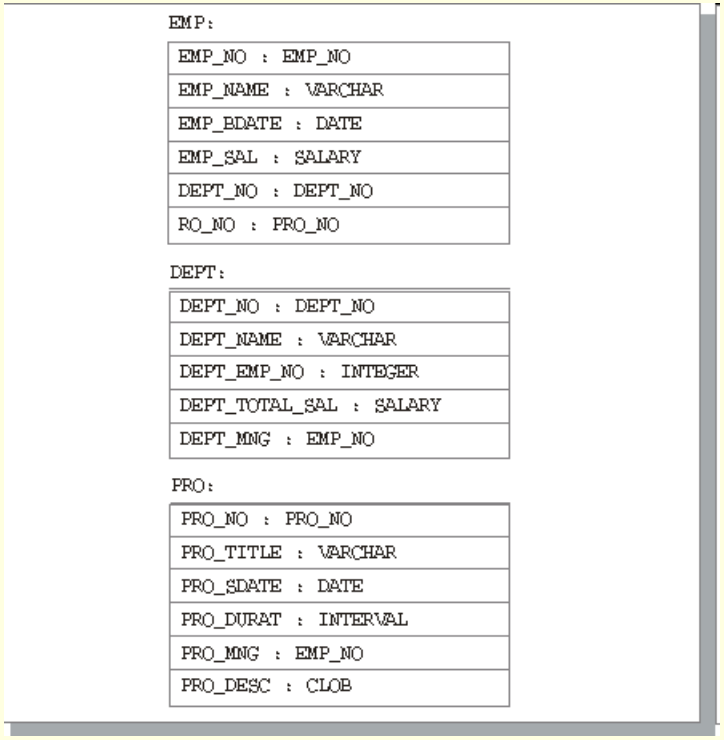
- Если в качестве требуемого ссылочного действия указано SET NULL, то
  - строка *t* обновляется, и
  - во всех столбцах, которые входят в состав внешнего ключа и соответствуют изменяемым столбцам таблицы *T*, всех строк, ссылающихся на строку *t*, проставляется NULL
- Если в определении внешнего ключа содержится спецификация MATCH PARTIAL, то подобному воздействию подвергаются только те строки таблицы S, которые
  - ссылаются исключительно на строку *t*

Средства определения, изменения и ликвидации базовых таблиц (30)

- **Примеры определений базовых таблиц**
- Определим таблицы служащих (EMP), отделов (DEPT) и проектов (PRO)
- Эти таблицы имеют заголовки, показанные на рисунке



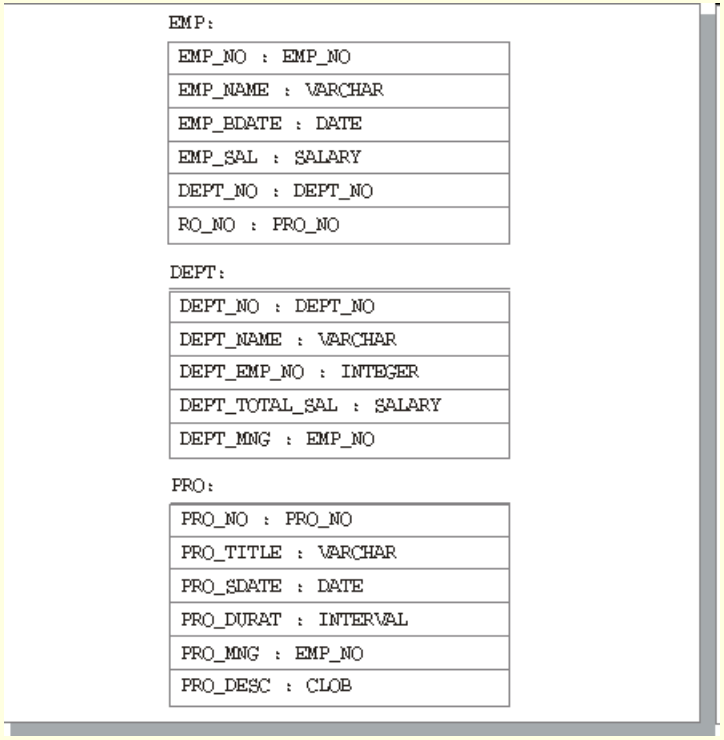
Средства определения, изменения и ликвидации базовых таблиц (31)



- Столбцы EMP_NO, EMP_SAL, DEPT_NO, PRO_NO, DEPT_TOTAL_SAL, DEPT_MNG и PRO_MNG определяются на ранее определенных доменах
  - ➢ определения доменов EMP_NO и SALARY приведены в предыдущей лекции
- Первичными ключами отношений EMP, DEPT и проектов PRO являются столбцы EMP_NO, DEPT_NO и PRO_NO соответственно

- В таблице EMP столбцы DEPT_NO и PRO_NO являются внешними ключами, указывающими на отдел, в котором работает служащий, и на выполняемый им проект соответственно

Средства определения, изменения и ликвидации базовых таблиц (31)



- В таблице DEPT внешним ключом является столбец DEPT_NO, указывающий на служащего, являющегося руководителем соответствующего отдела
- В таблице PRO внешним ключом является столбец PRO_MNG, указывающий на служащего, являющегося менеджером соответствующего проекта

- Другие ограничения целостности мы обсудим позже

Средства определения, изменения и ликвидации базовых таблиц (32)

- Определим таблицу EMP:

```
(1)     CREATE TABLE EMP (
(2)     EMP_NO EMP_NO PRIMARY KEY,
(3)     EMP_NAME VARCHAR(20) DEFAULT 'Incognito' NOT NULL,
(4)     EMP_BDATE DATE DEFAULT NULL CHECK (
        VALUE >= DATE '1917-10-24'),
(5)     EMP_SAL SALARY,
(6)     DEPT_NO DEPT_NO DEFAULT NULL REFERENCES
        DEPT ON DELETE SET NULL,
(7)     PRO_NO PRO_NO DEFAULT NULL,
(8)     FOREIGN KEY PRO_NO REFERENCES PRO (PRO_NO)
        ON DELETE SET NULL,
(9)     CONSTRAINT PRO_EMP_NO CHECK
      ((SELECT COUNT (*) FROM EMP E
        WHERE E.PRO_NO = PRO_NO) <= 50));
```

• В части (1) указывается, что создается таблица с именем EMP
• В части (2) определяется столбец EMP_NO на домене EMP_NO
➢ у этого столбца не определено значение по умолчанию, и он объявлен первичным ключом таблицы
✓ это ограничение целостности добавляется через AND к ограничениям, унаследованным столбцом от определения домена
✓ помимо прочего, это означает неявное указание запрета для данного столбца неопределенных значений

Средства определения, изменения и ликвидации базовых таблиц (33)

```
(1)     CREATE TABLE EMP (
(2)     EMP_NO EMP_NO PRIMARY KEY,
(3)     EMP_NAME VARCHAR(20) DEFAULT 'Incognito' NOT NULL,
(4)     EMP_BDATE DATE DEFAULT NULL CHECK (
        VALUE >= DATE '1917-10-24'),
(5)     EMP_SAL SALARY,
(6)     DEPT_NO DEPT_NO DEFAULT NULL REFERENCES
        DEPT ON DELETE SET NULL,
(7)     PRO_NO PRO_NO DEFAULT NULL,
(8)     FOREIGN KEY PRO_NO REFERENCES PRO (PRO_NO)
        ON DELETE SET NULL,
(9)     CONSTRAINT PRO_EMP_NO CHECK
      ((SELECT COUNT (*) FROM EMP E
         WHERE E.PRO_NO = PRO_NO) <= 50));
```

• В части (3) определен столбец EMP_NAME на базовом типе данных символьных строк переменной длины с максимальной длиной 20

➢ для столбца указано значение по умолчанию – строка 'Incognito', и в качестве ограничения целостности запрещены неопределенные значения

• В части (4) определяется столбец EMP_BDATE

➢ он имеет тип данных DATE, значением по умолчанию является NULL

➢ кроме того, ограничение столбца запрещает принимать на работу лиц, о которых известно, что они родились до Октябрьского переворота

Средства определения, изменения и ликвидации базовых таблиц (34)

```
(1)     CREATE TABLE EMP (
(2)     EMP_NO EMP_NO PRIMARY KEY,
(3)     EMP_NAME VARCHAR(20) DEFAULT 'Incognito' NOT NULL,
(4)     EMP_BDATE DATE DEFAULT NULL CHECK (
        VALUE >= DATE '1917-10-24'),
(5)     EMP_SAL SALARY,
(6)     DEPT_NO DEPT_NO DEFAULT NULL REFERENCES
        DEPT ON DELETE SET NULL,
(7)     PRO_NO PRO_NO DEFAULT NULL,
(8)     FOREIGN KEY PRO_NO REFERENCES PRO (PRO_NO)
        ON DELETE SET NULL,
(9)     CONSTRAINT PRO_EMP_NO CHECK
      ((SELECT COUNT (*) FROM EMP E
         WHERE E.PRO_NO = PRO_NO) <= 50));
```

- В части (5) определен столбец EMP_SAL на домене SALARY
- значение по умолчанию и ограничения целостности наследуются из определения домена
- В части (6) столбец DEPT_NO определяется на одноименном домене,
  - но явно объявляется, что значением по умолчанию этого столбца будет NULL
- Кроме того, добавляется ограничение внешнего ключа: столбец DEPT_NO ссылается на первичный ключ таблицы DEPT
  - определено ссылочное действие: при удалении строки из таблицы DEPT во всех строках таблицы EMP, ссылавшихся на эту строку, столбцу DEPT_NO должно быть присвоено неопределенное значение

Средства определения, изменения и ликвидации базовых таблиц (35)

```
(1)     CREATE TABLE EMP (
(2)     EMP_NO EMP_NO PRIMARY KEY,
(3)     EMP_NAME VARCHAR(20) DEFAULT 'Incognito' NOT NULL,
(4)     EMP_BDATE DATE DEFAULT NULL CHECK (
        VALUE >= DATE '1917-10-24'),
(5)     EMP_SAL SALARY,
(6)     DEPT_NO DEPT_NO DEFAULT NULL REFERENCES
        DEPT ON DELETE SET NULL,
(7)     PRO_NO PRO_NO DEFAULT NULL,
(8)     FOREIGN KEY PRO_NO REFERENCES PRO (PRO_NO)
        ON DELETE SET NULL,
(9)     CONSTRAINT PRO_EMP_NO CHECK
      ((SELECT COUNT (*) FROM EMP E
         WHERE E.PRO_NO = PRO_NO) <= 50));
```

- В части (7) определяется столбец PRO_NO
- Его определение аналогично определению столбца DEPT_NO, но ограничение внешнего ключа вынесено в часть (8), где оно определяется в полной форме как табличное ограничение
- Наконец, в части (9) определяется табличное проверочное ограничение с именем PRO_EMP_NO, которое требует, чтобы ни в одном проекте не участвовало больше 50 служащих

Средства определения, изменения и ликвидации базовых таблиц (36)

- Определим таблицу DEPT:

```
(1)      CREATE TABLE DEPT (
(2)      DEPT_NO DEPT_NO PRIMARY KEY,
(3)      DEPT_EMP_NO INTEGER NO NULL CHECK (
         VALUE BETWEEN 1 AND 100),
(4)      DEPT_NAME VARCHAR(200) DEFAULT 'Nameless' NOT NULL
(5)      DEPT_TOTAL_SAL SALARY DEFAULT 1000000.00
   NO NULL CHECK (VALUE > = 100000.00),
(6)      DEPT_MNG EMP_NO DEFAULT NULL
       REFERENCES EMP ON DELETE SET NULL
       CHECK (IF (VALUE IS NOT NULL) THEN
          ((SELECT COUNT(*) FROM DEPT
             WHERE DEPT.DEPT_MNG = VALUE) = 1),
(7)      CHECK (DEPT_EMP_NO =
       (SELECT COUNT(*) FROM EMP
          WHERE DEPT_NO = EMP.DEPT_NO)),
(8)      CHECK (DEPT_TOTAL_SAL >=
       (SELECT SUM(EMP_SAL) FROM EMP
          WHERE DEPT_NO = EMP.DEPT_NO)));
```

• В части (3) столбец DEPT_EMP_NO определен на базовом типе INTEGER без значения по умолчанию, с запретом неопределенного значения и с проверочным ограничением, устанавливающем допустимый диапазон значений числа служащих в отделе

• Еще одно проверочное ограничение этого столбца – (7) – вынесено на уровень определения табличного ограничения

➢ в каждой строке таблицы DEPT значение столбца DEPT_EMP_NO должно равняться общему числу строк таблицы EMP, в которых значение столбца DEPT_NO равно значению одноименного столбца данной строки таблицы DEPT

Средства определения, изменения и ликвидации базовых таблиц (37)

```
(1)      CREATE TABLE DEPT (
(2)      DEPT_NO DEPT_NO PRIMARY KEY,
(3)      DEPT_EMP_NO INTEGER NO NULL CHECK (
         VALUE BETWEEN 1 AND 100),
(4)      DEPT_NAME VARCHAR(200) DEFAULT 'Nameless' NOT NULL
(5)      DEPT_TOTAL_SAL SALARY DEFAULT 1000000.00
   NO NULL CHECK (VALUE > = 100000.00),
(6)      DEPT_MNG EMP_NO DEFAULT NULL
       REFERENCES EMP ON DELETE SET NULL
       CHECK (IF (VALUE IS NOT NULL) THEN
          ((SELECT COUNT(*) FROM DEPT
             WHERE DEPT.DEPT_MNG = VALUE) = 1),
(7)      CHECK (DEPT_EMP_NO =
       (SELECT COUNT(*) FROM EMP
          WHERE DEPT_NO = EMP.DEPT_NO)),
(8)      CHECK (DEPT_TOTAL_SAL >=
       (SELECT SUM(EMP_SAL) FROM EMP
          WHERE DEPT_NO = EMP.DEPT_NO)));
```

• В части (5) для определения столбца DEPT_TOTAL_SAL (объем фонда заработной платы отдела) используется домен SALARY

• но при этом явно установлено значение столбца по умолчанию, запрещено наличие неопределенных значений и введено дополнительное проверочное ограничение, определяющее нижний порог объема фонда заработной платы отдела

• проверочное ограничение (8) вынесено на уровень определения табличного ограничения

• в каждой строке таблицы DEPT значение столбца DEPT_TOTAL_SAL должно быть не меньше суммы значений столбца EMP_SAL во всех строках таблицы EMP, в которых значение столбца DEPT_NO равно значению одноименного столбца данной строки таблицы DEPT

Средства определения, изменения и ликвидации базовых таблиц (38)

```
(1)      CREATE TABLE DEPT (
(2)      DEPT_NO DEPT_NO PRIMARY KEY,
(3)      DEPT_EMP_NO INTEGER NO NULL CHECK (
         VALUE BETWEEN 1 AND 100),
(4)      DEPT_NAME VARCHAR(200) DEFAULT 'Nameless' NOT NULL
(5)      DEPT_TOTAL_SAL SALARY DEFAULT 1000000.00
   NO NULL CHECK (VALUE > = 100000.00),
(6)      DEPT_MNG EMP_NO DEFAULT NULL
       REFERENCES EMP ON DELETE SET NULL
       CHECK (IF (VALUE IS NOT NULL) THEN
          ((SELECT COUNT(*) FROM DEPT
             WHERE DEPT.DEPT_MNG = VALUE) = 1),
(7)      CHECK (DEPT_EMP_NO =
       (SELECT COUNT(*) FROM EMP
          WHERE DEPT_NO = EMP.DEPT_NO)),
(8)      CHECK (DEPT_TOTAL_SAL >=
       (SELECT SUM(EMP_SAL) FROM EMP
          WHERE DEPT_NO = EMP.DEPT_NO)));
```

- Обратите внимание на определение столбца DEPT_MNG – часть (6)
- этот столбец объявляется внешним ключом таблицы DEPT
- у отдела могут временно отсутствовать руководители, поэтому в столбце допускаются неопределенные значения
- но если у отдела имеется руководитель, то он должен являться руководителем только этого отдела
- на первый взгляд можно было бы воспользоваться ограничением столбца UNIQUE
- но такое ограничение допускало бы наличие неопределенного столбца DEPT_MNG только в одной строке таблицы DEPT, а мы хотим допустить отсутствие руководителя у нескольких отделов
- поэтому потребовалось ввести более громоздкое проверочное ограничение столбца

Средства определения, изменения и ликвидации базовых таблиц (39)

```
(1)      CREATE TABLE EMP (
(2)      EMP_NO EMP_NO PRIMARY KEY,
(3)      EMP_NAME VARCHAR(20) DEFAULT 'Incognito' NOT NULL,
(4)      EMP_BDATE DATE DEFAULT NULL CHECK (
         VALUE >= DATE '1917-10-24'),
(5)      EMP_SAL SALARY,
(6)      DEPT_NO DEPT_NO DEFAULT NULL REFERENCES
         DEPT ON DELETE SET NULL,
(7)      PRO_NO PRO_NO DEFAULT NULL,
(8)      FOREIGN KEY PRO_NO REFERENCES PRO (PRO_NO)
         ON DELETE SET NULL,
(9)      CONSTRAINT PRO_EMP_NO CHECK
       ((SELECT COUNT (*) FROM EMP E
          WHERE E.PRO_NO = PRO_NO) <= 50));
```

```
(1)      CREATE TABLE DEPT (
(2)      DEPT_NO DEPT_NO PRIMARY KEY,
(3)      DEPT_EMP_NO INTEGER NO NULL CHECK (
         VALUE BETWEEN 1 AND 100),
(4)      DEPT_NAME VARCHAR(200) DEFAULT 'Nameless' NOT NULL
(5)      DEPT_TOTAL_SAL SALARY DEFAULT 1000000.00
    NO NULL CHECK (VALUE > = 100000.00),
(6)      DEPT_MNG EMP_NO DEFAULT NULL
       REFERENCES EMP ON DELETE SET NULL
       CHECK (IF (VALUE IS NOT NULL) THEN
          ((SELECT COUNT(*) FROM DEPT
             WHERE DEPT.DEPT_MNG = VALUE) = 1),
(7)      CHECK (DEPT_EMP_NO =
       (SELECT COUNT(*) FROM EMP
          WHERE DEPT_NO = EMP.DEPT_NO)),
(8)      CHECK (DEPT_TOTAL_SAL >=
       (SELECT SUM(EMP_SAL) FROM EMP
          WHERE DEPT_NO = EMP.DEPT_NO)));
```

• По поводу двух приведенных определений базовых таблиц у читателей могут возникнуть два вопроса:

(a) почему проверочное ограничение (9) в первом определении и проверочные ограничения (7) и (8) во втором определении вынесены из определений соответствующих столбцов, хотя формально являются именно ограничениями столбцов?

(b) почему ограничению (9) в первом определении присвоено явное имя, а ограничения (7) и (8) во втором определении оставлены безымянными?

Средства определения, изменения и ликвидации базовых таблиц (40)

```
(1)      CREATE TABLE EMP (
(2)      EMP_NO EMP_NO PRIMARY KEY,
(3)      EMP_NAME VARCHAR(20) DEFAULT 'Incognito' NOT NULL,
(4)      EMP_BDATE DATE DEFAULT NULL CHECK (
         VALUE >= DATE '1917-10-24'),
(5)      EMP_SAL SALARY,
(6)      DEPT_NO DEPT_NO DEFAULT NULL REFERENCES
         DEPT ON DELETE SET NULL,
(7)      PRO_NO PRO_NO DEFAULT NULL,
(8)      FOREIGN KEY PRO_NO REFERENCES PRO (PRO_NO)
         ON DELETE SET NULL,
(9)      CONSTRAINT PRO_EMP_NO CHECK
       ((SELECT COUNT (*) FROM EMP E
          WHERE E.PRO_NO = PRO_NO) <= 50));
```

```
(1)      CREATE TABLE DEPT (
(2)      DEPT_NO DEPT_NO PRIMARY KEY,
(3)      DEPT_EMP_NO INTEGER NO NULL CHECK (
         VALUE BETWEEN 1 AND 100),
(4)      DEPT_NAME VARCHAR(200) DEFAULT 'Nameless' NOT NULL
(5)      DEPT_TOTAL_SAL SALARY DEFAULT 1000000.00
   NO NULL CHECK (VALUE > = 100000.00),
(6)      DEPT_MNG EMP_NO DEFAULT NULL
       REFERENCES EMP ON DELETE SET NULL
       CHECK (IF (VALUE IS NOT NULL) THEN
          ((SELECT COUNT(*) FROM DEPT
             WHERE DEPT.DEPT_MNG = VALUE) = 1),
(7)      CHECK (DEPT_EMP_NO =
       (SELECT COUNT(*) FROM EMP
          WHERE DEPT_NO = EMP.DEPT_NO)),
(8)      CHECK (DEPT_TOTAL_SAL >=
       (SELECT SUM(EMP_SAL) FROM EMP
          WHERE DEPT_NO = EMP.DEPT_NO)));
```

- На первый вопрос можно ответить следующим образом
- да, эти ограничения можно было бы включить в определения столбцов
- это дело вкуса
- но все три ограничения являются очень важными с точки зрения организации таблиц в целом
- поэтому лучше показывать их на уровне определения табличных ограничений

Средства определения, изменения и ликвидации базовых таблиц (41)

```
(1)      CREATE TABLE EMP (
(2)      EMP_NO EMP_NO PRIMARY KEY,
(3)      EMP_NAME VARCHAR(20) DEFAULT 'Incognito' NOT NULL,
(4)      EMP_BDATE DATE DEFAULT NULL CHECK (
         VALUE >= DATE '1917-10-24'),
(5)      EMP_SAL SALARY,
(6)      DEPT_NO DEPT_NO DEFAULT NULL REFERENCES
         DEPT ON DELETE SET NULL,
(7)      PRO_NO PRO_NO DEFAULT NULL,
(8)      FOREIGN KEY PRO_NO REFERENCES PRO (PRO_NO)
         ON DELETE SET NULL,
(9)      CONSTRAINT PRO_EMP_NO CHECK
       ((SELECT COUNT (*) FROM EMP E
          WHERE E.PRO_NO = PRO_NO) <= 50));
```

```
(1)      CREATE TABLE DEPT (
(2)      DEPT_NO DEPT_NO PRIMARY KEY,
(3)      DEPT_EMP_NO INTEGER NO NULL CHECK (
         VALUE BETWEEN 1 AND 100),
(4)      DEPT_NAME VARCHAR(200) DEFAULT 'Nameless' NOT NULL
(5)      DEPT_TOTAL_SAL SALARY DEFAULT 1000000.00
   NO NULL CHECK (VALUE > = 100000.00),
(6)      DEPT_MNG EMP_NO DEFAULT NULL
       REFERENCES EMP ON DELETE SET NULL
       CHECK (IF (VALUE IS NOT NULL) THEN
          ((SELECT COUNT(*) FROM DEPT
             WHERE DEPT.DEPT_MNG = VALUE) = 1),
(7)      CHECK (DEPT_EMP_NO =
       (SELECT COUNT(*) FROM EMP
          WHERE DEPT_NO = EMP.DEPT_NO)),
(8)      CHECK (DEPT_TOTAL_SAL >=
       (SELECT SUM(EMP_SAL) FROM EMP
          WHERE DEPT_NO = EMP.DEPT_NO)));
```

- Вот ответ на второй вопрос
- Ограничение (9) в первом определении и ограничения (7) и (8) во втором определении внешне похожи, но сильно отличаются по своей сути
- Ограничения (7) и (8) связаны с агрегатной семантикой столбцов DEPT_EMP_NO и DEPT_TOTAL_SAL таблицы DEPT
- отмена ограничений изменила бы смысл этих столбцов
- Ограничение (9) является текущим административным ограничением
- если руководство примет решение разрешить использовать в одном проекте более 50 служащих, ограничение можно отменить без изменения смысла столбцов таблицы EMP
- имея это в виду, мы ввели явное имя ограничения (9), чтобы при необходимости имелась простая возможность отменить это ограничение с помощью оператора ALTER TABLE

Средства определения, изменения и ликвидации базовых таблиц (42)

- Наконец, определим таблицу PRO:

```
(1)     CREATE TABLE PRO (
(2)     PRO_NO PRO_NO PRIMARY KEY,
(3)     PRO_TITLE VARCHAR(200)DEFAULT 'No title' NOT NULL,
(4)     PRO_SDATE DATE DEFAULT CURRENT_DATE NOT NULL,
(5)     PRO_DURAT INTERVAL YEAR DEFAUL INTERVAL '1'
        YEAR NOT NULL,
(6)     PRO_MNG EMP_NO UNIQUE NOT NULL
        REFERENCES EMP ON DELETE NO ACTION,
(7)     PRO_DESC CLOB(10M));
```

• Столбец PRO_SDATE содержит дату начала проекта, а столбец PRO_DURAT – продолжительность проекта в годах

• В этом определении имеет смысл прокомментировать часть (6)

➢ мы считаем, что если отдел, по крайней мере временно, может существовать без руководителя, то у проекта всегда должен быть менеджер

➢ поэтому определение столбца PRO_MNG является гораздо более строгим, чем определение столбца DEPT_MNG в таблице DEPT

➢ сочетание ограничений UNIQUE и NOT NULL при отсутствии значений по умолчанию приводит к абсолютной уникальности значений столбца PRO_MNG

Средства определения, изменения и ликвидации базовых таблиц (43)

- Наконец, определим таблицу PRO:

```
(1)     CREATE TABLE PRO (
(2)     PRO_NO PRO_NO PRIMARY KEY,
(3)     PRO_TITLE VARCHAR(200)DEFAULT 'No title' NOT NULL,
(4)     PRO_SDATE DATE DEFAULT CURRENT_DATE NOT NULL,
(5)     PRO_DURAT INTERVAL YEAR DEFAUL INTERVAL '1'
        YEAR NOT NULL,
(6)     PRO_MNG EMP_NO UNIQUE NOT NULL
        REFERENCES EMP ON DELETE NO ACTION,
(7)     PRO_DESC CLOB(10M));
```

➢ другими словами, этот столбец обладает всеми характеристиками первичного ключа, хотя объявлен только как возможный ключ

➢ кроме того, он объявлен как внешний ключ с действием при удалении строки таблицы EMP с соответствующим значением первичного ключа NO ACTION, запрещающим такие удаления

➢ в совокупности это гарантирует, что у любого проекта будет существовать менеджер, являющийся служащим предприятия

• В части (5) столбец PRO_DESC (описание проекта) определен как большой символьный объект с максимальным размером 10 Мбайт

Средства определения, изменения и ликвидации базовых таблиц (44)

- **Изменение определения базовой таблицы**
- Оператор изменения определения базовой таблицы ALTER TABLE имеет следующий синтаксис:

```
base_table_alteration ::= ALTER TABLE base_table_name
    column_alteration_action
    |  base_table_constraint_alternation_action
```

- При выполнении одного оператора ALTER TABLE может быть выполнено
  - либо действие по изменению определения столбца,
  - либо действие по изменению определения табличного ограничения целостности

Средства определения, изменения и ликвидации базовых таблиц (45)

- **Добавление, изменение или удаление определения столбца**
- Действие по изменению определения столбца специфицируется в следующем синтаксисе:

```
column_alteration_action ::=
        ADD [ COLUMN ] column_definition
    | ALTER [ COLUMN ] column_name
        { SET default_definition | DROP DEFAULT }
    | DROP [ COLUMN ] column_name
                { RESTRICT | CASCADE }
```

- С использованием оператора ALTER TABLE можно
  - добавлять к определению таблицы определение нового столбца (действие ADD)
  - и изменять или отменять определение существующего столбца (действия ALTER и DROP соответственно)

Средства определения, изменения и ликвидации базовых таблиц (46)

- Смысл действия ADD COLUMN почти полностью совпадает со смыслом раздела определения столбца в операторе CREATE TABLE
  - указывается имя нового столбца, его тип данных или домен
  - могут определяться значение по умолчанию и ограничения целостности
- Однако имеется одно существенное отличие:
  - столбец, определяемый в действии ADD оператора ALTER TABLE, добавляется к уже существующей таблице, которая, скорее всего, содержит некоторый набор строк

Средства определения, изменения и ликвидации базовых таблиц (47)

- В каждой из существующих строк новый столбец должен содержать некоторое значение, и считается, что
  - сразу после выполнения действия ADD этим значением является значение столбца по умолчанию
- Поэтому столбец, определяемый в действии ADD, обязательно должен иметь значение по умолчанию,
  - т. е. для него недопустима ситуация, когда значением по умолчанию явно или неявно объявлено неопределенное значение (NULL), но среди ограничений целостности столбца присутствует ограничение NOT NULL

Средства определения, изменения и ликвидации базовых таблиц (48)

- В действии ALTER COLUMN можно
  - изменить (SET default_definition) или
  - отменить определение значения по умолчанию для существующего столбца
- Правила определения нового действующего значения столбца по умолчанию совпадают с соответствующими правилами, обсуждавшимися в подразделе определения столбца в операторе CREATE TABLE
- Изменение значения столбца по умолчанию не оказывает влияния на состояние существующих строк таблицы
  - даже если в некоторых из них хранится предыдущее значение столбца по умолчанию
- Если столбец определен на домене, у которого существует значение по умолчанию, то
  - после отмены определения значения столбца по умолчанию для этого столбца начинает действовать значение по умолчанию домена

Средства определения, изменения и ликвидации базовых таблиц (49)

- Действие DROP COLUMN отменяет определение существующего столбца
  - удаляет его из таблицы
- Действие DROP COLUMN отвергается, если:
  - указанный столбец является единственным столбцом таблицы;
  - или в этом действии присутствует спецификация RESTRICT, и данный столбец используется в определении каких-либо представлений или ограничений целостности
- Если в действии присутствует спецификация CASCADE, то его выполнение порождает неявное выполнение оператора DROP для всех представлений и ограничений целостности, в определении которых используется данный столбец

Средства определения, изменения и ликвидации базовых таблиц (50)

- **Примеры изменения определения столбца**
- Пусть на предприятии ввели систему премирования служащих
  - каждый служащий может дополнительно к зарплате получать ежемесячную премию, не превышающую размер его зарплаты
- Тогда разумно добавить к таблице EMP новый столбец EMP_BONUS, используя оператор ALTER TABLE:

```
ALTER TABLE EMP
ADD EMP_BONUS SALARY DEFAULT NULL
   CONSTRAINT BONSAL CHECK (VALUE < EMP_SAL);
```

  - проверочному ограничению столбца присвоили явное имя
  - если ограничения на размер премии изменятся, можно будет легко отменить это ограничение, как табличное

Средства определения, изменения и ликвидации базовых таблиц (51)

- При определении столбца EMP_SAL таблицы EMP для этого столбца явно не определялось значение по умолчанию
  - оно наследовалось из определения домена
- Если в какой-то момент это стало неправильным
  - например, повысился размер минимальной зарплаты,

можно установить новое значение по умолчанию:

```
ALTER TABLE EMP ALTER EMP_SAL SET DEFAULT 15000.00.
```

Средства определения, изменения и ликвидации базовых таблиц (52)

- При определении столбца DEPT_TOTAL_SAL таблицы DEPT для него было установлено значение по умолчанию 1000000
- Такие важные данные нехорошо устанавливать по умолчанию
- Это значение по умолчанию можно отменить:

```
ALTER TABLE DEPT ALTER DEPT_TOTAL_SAL DROP DEFAULT.
```

- После выполнения этого оператора при вставке новой строки в таблицу DEPT всегда потребуется явно указывать значение столбца DEPT_TOTAL_SAL
  - хотя формально у столбца будет существовать значение по умолчанию, наследуемое от домена SALARY (10000.00), оно не может быть занесено в таблицу DEPT, поскольку противоречит ограничению столбца DEPT_TOTAL_SAL CHECK (VALUE >= 100000.00)

Средства определения, изменения и ликвидации базовых таблиц (53)

- Действительно ли требуется поддерживать в таблице DEPT столбец DEPT_EMP_NO?
- Для его поддержки требуется проверять громоздкое ограничение целостности, а число служащих в любом отделе можно получить динамически с помощью простого запроса к таблице EMP
  - этот запрос входит в ограничение целостности
- Может оказаться разумным отменить определение столбца DEPT_EMP_NO, выполнив следующий оператор ALTER TABLE:

```
ALTER TABLE DEPT DROP DEPT_EMP_NO CASCADE.
```

Средства определения, изменения и ликвидации базовых таблиц (54)

- Напомним, что спецификация CASCADE ведет к тому, что при выполнении оператора будет уничтожено не только определение указанного столбца, но и
  - определения всех ограничений целостности и представлений, в которых используется уничтожаемый столбец
- В нашем случае единственное связанное с этим столбцом ограничение целостности, определенное вне определения столбца, было бы отменено, даже если бы в операторе отмены определения столбца DEPT_EMP_NO содержалась спецификация RESTRICT,
  - поскольку это единственное внешнее определение ограничения является ограничением только столбца DEPT_EMP_NO

Средства определения, изменения и ликвидации базовых таблиц (55)

- **Изменение набора табличных ограничений**
- Действие по изменению набора табличных ограничений специфицируется в следующем синтаксисе:

```
base_table_constraint_alternation_action ::=
        ADD [ CONSTRAINT ] base_table_constraint_definitior
    | DROP CONSTRAINT constraint_name
        { RESTRICT | CASCADE }
```

- Действие ADD [CONSTRAINT] позволяет добавить к набору существующих ограничений таблицы новое ограничение целостности
- Можно считать, что новое ограничение добавляется через AND к конъюнкции существующих ограничений, как если бы оно определялось в составе оператора CREATE TABLE

Средства определения, изменения и ликвидации базовых таблиц (55)

- Но имеется одно существенное отличие
- Если внимательно посмотреть на все возможные виды табличных ограничений, можно убедиться, что любое из них удовлетворяется на пустой таблице
  - какой бы набор табличных ограничений ни был определен при создании таблицы, это определение является допустимым и не препятствует выполнению оператора CREATE TABLE
- При добавлении нового табличного ограничения с использованием действия ADD [CONSTRAINT] мы имеем другую ситуацию,
  - поскольку таблица, скорее всего, уже содержит некоторый набор строк, для которого условное выражение нового ограничения может принять значение false
- В этом случае выполнение оператора ALTER TABLE, включающего действие ADD [CONSTRAINT], отвергается

Средства определения, изменения и ликвидации базовых таблиц (56)

- Выполнение действия DROP CONSTRAINT приводит к отмене определения существующего табличного ограничения
  - можно отменить определение только именованных табличных ограничений
- Спецификации RESTRICT и CASCADE осмыслены только в том случае, если отменяемое ограничение является ограничением возможного ключа
  - UNIQUE или PRIMARY KEY
- При указании RESTRICT действие отвергается, если на данный возможный ключ ссылается хотя бы один внешний ключ
- При указании CASCADE действие DROP CONSTRAINT выполняется в любом случае, и все определения таких внешних ключей также отменяются

Средства определения, изменения и ликвидации базовых таблиц (57)

- **Примеры изменения набора табличных ограничений**
- Мы добавили к таблице EMP столбец EMP_BONUS
  - размеры ежемесячных премий служащих
- Предположим, что премии выплачиваются из фонда заработной платы отдела, в котором работает служащий
- Проверочное ограничение столбца DEPT_TOTAL_SAL,
  - объем фонда зарплаты отдела не должен быть меньше суммарной зарплаты служащих этого отдела,

  становится недостаточным,
  - требуется добавить к набору ограничений таблицы DEPT новое ограничение

```
ALTER TABLE DEPT ADD CONSTRAINT TOTAL_INCOME
CHECK (DEPT_TOTAL_SAL >=
    (SELECT SUM(EMP_SAL + COALESCE(EMP_BONUS,0))
   FROM EMP WHERE EMP.DEPT_NO = DEPT_NO)).
```

Средства определения, изменения и ликвидации базовых таблиц (58)

- Хотя это ограничение на вид довольно сложное, смысл его очень прост:
  - суммарный доход служащих отдела не должен превышать объем зарплаты отдела
- В арифметическом выражении под знаком агрегатной операции SUM используется операция COALRSCE
- Эта двуместная операция определяется следующим образом:

```
COALESCE (x, y) IF x IS NOT NULL THEN x ELSE y,
```

- Нам пришлось воспользоваться этой операцией, поскольку в столбце EMP_BONUS допускается наличие неопределенных значений

Средства определения, изменения и ликвидации базовых таблиц (59)

- Понятно, что новое ограничение столбца DEPT_TOTAL_SAL сильнее предыдущего, и это предыдущее ограничение можно было бы отменить
- Конечно, с логической точки зрения наличие обоих ограничений ничему не повредит
  - предыдущее ограничение является логическим следствием нового
- но при использовании не слишком интеллектуальной реализации SQL может привести к замедлению работы системы,
  - поскольку оба ограничения могут проверяться независимо
- К сожалению, при определении таблицы EMP мы не присвоили явное имя проверочному ограничению столбца DEPT_TOTAL_SAL
  - и поэтому не можем немедленно продемонстрировать оператор отмены этого ограничения
- Это не значит, что его нельзя отменить вообще

Средства определения, изменения и ликвидации базовых таблиц (60)

- Новому ограничению мы присвоили явное имя
- К этому привели следующие рассуждения
  - когда создавалась исходная схема базы данных, руководство предприятия ничего не говорило о премиях служащих
  - теперь начальство решило, что премии будут выплачиваться из фонда зарплаты
  - для этого, мы добавили новый столбец и новое ограничение целостности
  - но кто знает, не изменится ли снова решение о премиях?
- Чтобы не добавлять себе работы в будущем, дадим новому ограничению явное имя и не будем отменять предыдущее ограничение

Средства определения, изменения и ликвидации базовых таблиц (61)

- При определении таблицы EMP было специфицировано проверочное табличное ограничение PRO_EMP_NO,
  - над одним проектом не должно работать более 50 служащих
- Это ограничение носит чисто административный характер и может быть отменено без нарушения логики базы данных
- Для отмены ограничения нужно выполнить следующий оператор:

```
ALTER TABLE EMP DROP CONSTRAINT PRO_EMP_NO;
```

Средства определения, изменения и ликвидации базовых таблиц (62)

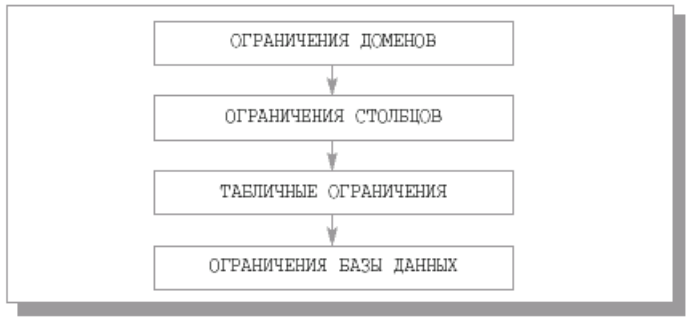
- **Отмена определения (уничтожение) базовой таблицы**
- Для отмены определения (уничтожения) базовой таблицы служит оператор DROP TABLE, задаваемый в следующем синтаксисе:

```
DROP TABLE base_table_name { RESTRICT | CADCADE }
```

- Успешное выполнение оператора приводит к тому, что указанная базовая таблица перестает существовать
- Уничтожаются все ее строки, определения столбцов и табличные определения целостности
- При наличии спецификации RESTRICT выполнение оператора DROP TABLE отвергается, если имя таблицы используется в каком-либо определении представления или ограничения целостности
- При наличии спецификации CASCADE оператор выполняется в любом случае, и все определения представлений и ограничений целостности, содержащие ссылки на данную таблицу, также отменяются

Средства определения и отмены общих ограничений целостности (1)

- Виды ограничений целостности, с которыми мы имели дело в предыдущих разделах этой лекции, образуют иерархию

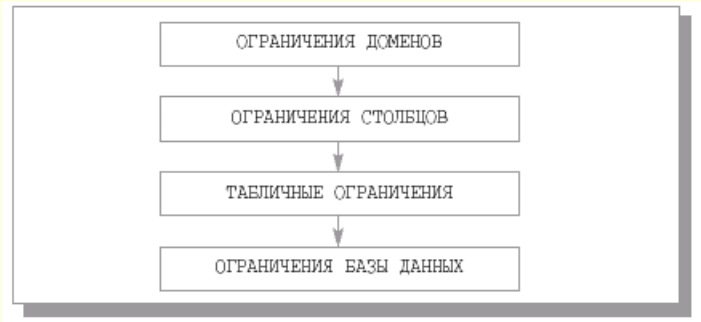


- Ограничения целостности, входящие в определение домена, наследуются всеми столбцами, определенными на этих доменах, и являются ограничениями этих столбцов
  - кроме того, в определение столбца могут входить определения дополнительных ограничений
- Ограничения целостности, входящие в определение столбца
  - включая ограничения, унаследованные из определения домена,

  являются ограничениями таблицы, в состав определения которой входит определение данного столбца
  - кроме того, в определение таблицы могут входить определения дополнительных ограничения

Средства определения и отмены общих ограничений целостности (2)

- Но иерархия видов ограничений целостности этим не исчерпывается



- Ограничения целостности, входящие в определение таблицы
  - включая явные и унаследованные от определения доменов ограничения столбцов,

  представляют собой ограничения базы данных, частью которой является данная таблица
  - кроме того, могут определяться дополнительные ограничения базы данных
- В стандарте SQL такие дополнительные ограничения базы данных называются ASSERTION, а мы их будем называть *общими ограничениями целостности*

Средства определения и отмены общих ограничений целостности (3)

- **Определение общих ограничений целостности**
- Для определения общего ограничения целостности служит оператор CREATE ASSERTION, задаваемый в следующем синтаксисе:

```
CREATE ASSERTION constraint_name
    CHECK (conditional_expression)
```

- Заметим, что при создании общего ограничения целостности его имя всегда должно указываться явно
- Хотя синтаксис определения общего ограничения совпадает с синтаксисом определений ограничений столбца и таблицы, в данном случае допускаются только
  - специальные виды условных выражений

Средства определения и отмены общих ограничений целостности (4)

- Мы не можем сейчас точно сформулировать свойства этих видов условий,
  - поскольку отложили подробное рассмотрение разновидностей условных выражений до следующих лекций
- Если говорить неформально, то особые свойства условий связаны с тем, что
  - при определении общих ограничений целостности контекстом, в котором вычисляется условное выражение, является
    - весь набор таблиц базы данных,
    - а не набор строк таблицы, как это было при определении табличных ограничений.

Средства определения и отмены общих ограничений целостности (5)

- Продемонстрируем и прокомментируем несколько примеров определений общих ограничений целостности
- В определении таблицы EMP содержалось ограничение столбца EMP_BDATE:

```
CHECK (EMP_BDATE >= '1917-10-24')
```

- Вот каким образом можно определить такое же ограничение на уровне общих ограничений целостности:

```
CONSTRAINT PRO_EMP_NO CHECK
    ((SELECT COUNT (*) FROM EMP E
        WHERE E.PRO_NO = PRO_NO) <= 50)
```

- В логическом условии этого общего ограничения выбирается минимальное значение столбца EMP_BDATE
  - дата рождения самого старого служащего
- Значением условного выражения будет false в том и только в том случае, если среди служащих имеется хотя бы один, родившийся до указанной даты

Средства определения и отмены общих ограничений целостности (6)

- Покажем, как можно сформулировать в виде общего ограничения целостности ограничение внешнего ключа
- Например, приведем эквивалентную формулировку для определения внешнего ключа PRO_NO, входящего в состав определения таблицы EMP:

```
FOREIGN KEY PRO_NO REFERENCES PRO (PRO_NO)
```

- В виде общего ограничения целостности это может выглядеть следующим образом:

```
(1) CREATE ASSERTION FK_PRO_NO CHECK
(2) ( NOT EXISTS (SELECT * FROM EMP
        WHERE PRO_NO IS NOT NULL AND
(3)             NOT EXISTS (SELECT * FROM PRO
(4)                 WHERE PRO.PRO_NO = EMP.PRO_NO))).
```

- Логическое выражение этого ограничения выглядит достаточно сложным и нуждается в пояснении

Средства определения и отмены общих ограничений целостности (7)

```
(1) CREATE ASSERTION FK_PRO_NO CHECK
(2) ( NOT EXISTS (SELECT * FROM EMP
        WHERE PRO_NO IS NOT NULL AND
(3)             NOT EXISTS (SELECT * FROM PRO
(4)                 WHERE PRO.PRO_NO = EMP.PRO_NO))).
```

- Условие выборки оператора SELECT на строке (2) состоит из двух частей, связанных через AND
- Первая часть отфильтровывает те строки таблицы EMP, у которых в столбце PRO_NO содержится NULL
- Если этот столбец содержит NULL во всех строках таблицы, то результирующая таблица оператора выборки на строке (2) будет пустой, и значением предиката NOT EXISTS будет true,
  - т. е. ограничение удовлетворяется

Средства определения и отмены общих ограничений целостности (8)

```
(1) CREATE ASSERTION FK_PRO_NO CHECK
(2) ( NOT EXISTS (SELECT * FROM EMP
        WHERE PRO_NO IS NOT NULL AND
(3)             NOT EXISTS (SELECT * FROM PRO
(4)                 WHERE PRO.PRO_NO = EMP.PRO_NO))).
```

- Теперь предположим, что в таблице EMP нашлась строка emp, в столбце PRO_NO которой содержится значение, отличное от NULL
  - назовем это значение cand_pro_no
- Для него вычисляется вторая часть условия выборки оператора SELECT на строке (2)
- Оператор выборки на строке (3) выбирает все строки таблицы PRO, значение столбца PRO_NO которых равняется cand_pro_no
- Если для данного значения cand_pro_no нашлась хотя бы одна такая строка, то результирующая таблица оператора выборки на строке (3) будет непустой, и значением предиката NOT EXISTS на строке (3) будет false
- Соответственно, все условие выборки первого оператора SELECT примет значение false, и строка со значением cand_pro_no в столбце PRO_NO будет отфильтрована

Средства определения и отмены общих ограничений целостности (9)

```
(1) CREATE ASSERTION FK_PRO_NO CHECK
(2) ( NOT EXISTS (SELECT * FROM EMP
        WHERE PRO_NO IS NOT NULL AND
(3)             NOT EXISTS (SELECT * FROM PRO
(4)                 WHERE PRO.PRO_NO = EMP.PRO_NO))).
```

- Если же найдется хотя бы одна строка таблицы EMP с таким значением cand_pro_no столбца PRO_NO, что
  - в таблице PRO не найдется ни одной строки, значение столбца PRO_NO которой равнялось бы этому cand_pro_no,

  то результирующая таблица оператора выборки на строке (3) будет пустой,
  - и значением предиката NOT EXISTS на строке (3) будет true
- Тогда все условие выборки первого оператора SELECT примет значение true, и эта строка таблицы EMP будет пропущена в результирующую таблицу
- Значением предиката NOT EXISTS будет false, т. е. ограничение не удовлетворяется
- Подробное пояснение, чтобы дать понять, во что реально вырождается простая синтаксическая конструкция определения внешнего ключа

Средства определения и отмены общих ограничений целостности (10)

- Если же найдется хотя бы одна строка таблицы EMP с таким значением cand_pro_no столбца PRO_NO, что
  - в таблице PRO не найдется ни одной строки, значение столбца PRO_NO которой равнялось бы этому cand_pro_no,

  то результирующая таблица оператора выборки на строке (3) будет пустой,
  - и значением предиката NOT EXISTS на строке (3) будет true
- Тогда все условие выборки первого оператора SELECT примет значение true, и эта строка таблицы EMP будет пропущена в результирующую таблицу
- Значением предиката NOT EXISTS будет false, т. е. ограничение не удовлетворяется
- Подробное пояснение, чтобы дать понять, во что реально вырождается простая синтаксическая конструкция определения внешнего ключа

Средства определения и отмены общих ограничений целостности (11)

- **Отмена определения общего ограничения целостности**
- Для того чтобы отменить ранее определенное общее ограничение целостности, нужно воспользоваться оператором DROP ASSERTION, задаваемым в следующем синтаксисе:

```
DROP ASSERTION constraint_name
```

- Вот пример оператора, отменяющего определение дискриминационного общего ограничения целостности PRO_MNG_CONSTR:

```
DROP ASSERTION PRO_MNG_CONSTR;
```

Средства определения и отмены общих ограничений целостности (12)

- **Немедленная и откладываемая проверка ограничений**
- На первый взгляд кажется, что ограничения целостности (всех видов) должны немедленно проверяться в случае выполнения любого действия, изменяющего содержимое базы данных
  - вставка в любую таблицу новой строки, изменение или удаление существующих строк
- Однако можно определить такие ограничения целостности, логическое выражение которых будет принимать значение false при любой немедленной проверке
- Одним из примеров такого ограничения является ограничение таблицы DEPT

```
CHECK (DEPT_EMP_NO =
    (SELECT COUNT(*) FROM EMP
        WHERE DEPT_NO = EMP.DEPT_NO))
```

Средства определения и отмены общих ограничений целостности (13)

```
CHECK (DEPT_EMP_NO =
    (SELECT COUNT(*) FROM EMP
        WHERE DEPT_NO = EMP.DEPT_NO))
```

- Предположим, например, что в отдел зачисляется новый служащий
- Тогда нужно выполнить две операции:
  - вставить новую строку в таблицу EMP
  - и изменить соответствующую строку таблицы DEPT
    - прибавить единицу к значению столбца DEPT_EMP_NO
- Очевидно, что в каком бы порядке ни выполнялись эти операции,
  - сразу после выполнения первой из них ограничение целостности будет нарушено,
  - соответствующее действие будет отвергнуто,
  - и мы никогда не сможем принять на работу нового служащего

Средства определения и отмены общих ограничений целостности (14)

- Поскольку ограничения целостности, немедленная проверка которых бессмысленна, являются нужными и полезными, в язык SQL включены средства, позволяющие регулировать время проверки ограничений
- В контексте каждой выполняемой транзакции каждое ограничение целостности должно находиться в одном из двух режимов:
  - режиме немедленной проверки (immediate)
  - или режиме отложенной проверки (deferred)
- Все ограничения целостности, находящиеся в режиме немедленной проверки, проверяются
  - при выполнении в транзакции любой операции, изменяющей состояние базы данных
- Если действие операции нарушает какое-либо немедленно проверяемое ограничение целостности, то это действие отвергается

Средства определения и отмены общих ограничений целостности (15)

- Ограничения целостности, находящиеся в режиме отложенной проверки, проверяются при завершении транзакции
  - выполнении операции COMMIT
- Если действия этой транзакции нарушают какое-либо отложенно проверяемое ограничение целостности, то транзакция откатывается
  - операция COMMIT трактуется как операция ROLLBACK

Средства определения и отмены общих ограничений целостности (16)

- Для этого в качестве заключительной синтаксической конструкции к любому определению ограничения целостности (любого вида) может быть добавлена спецификация INITIALLY в следующей синтаксической форме:

```
INITIALLY { DEFERRED | IMMEDIATE }
    [ [ NOT ] DEFERRABLE ]
```

- Эта спецификация указывает, в каком режиме должно находиться данное ограничение целостности в начале выполнения любой транзакции
  - INITIALLY IMMEDIATE означает, что в начале выполнения транзакции данное ограничение будет находиться в режиме немедленной проверки,
  - а INITIALLY DEFERRED – что в начале любой транзакции ограничение будет находиться в режиме отложенной проверки,
- а также возможности смены режима этого ограничения при выполнении транзакции
  - DEFERRABLE означает, что для данного ограничения может быть установлен режим отложенной проверки,
  - а NOT DEFERRABLE – что не может

Средства определения и отмены общих ограничений целостности (17)

- Комбинация INITIALLY DEFERRED NOT DEFERRABLE является недопустимой
- Если в определении ограничения спецификация начального режима проверки отсутствует, то подразумевается наличие спецификации INITIALLY IMMEDIATE
- При наличии явной или неявной спецификации INITIALLY IMMEDIATE и отсутствии явного указания возможности смены режима подразумевается наличие спецификации NOT DEFERRABLE
- При наличии спецификации INITIALLY DEFERRED и отсутствии явного указания возможности смены режима подразумевается наличие спецификации DEFERRABLE

Средства определения и отмены общих ограничений целостности (18)

- При выполнении транзакции можно изменить режим проверки некоторых или всех ограничений целостности для данной транзакции
- Для этого используется оператор SET CONSTRAINTS, задаваемый в следующем синтаксисе:

```
SET CONSTRAINTS { constraint_name_commalist | ALL }
        { DEFERRED | IMMEDIATE }
```

- Если в операторе указывается список имен ограничений целостности, то все они должны быть DEFERRABLE
  - если хотя бы для одного ограничения из списка это требование не выполняется, то операция SET CONSTRAINTS отвергается
- При указании ключевого слова ALL режим устанавливается для всех ограничений, в определении которых явно или неявно было указано DEFERRABLE

Средства определения и отмены общих ограничений целостности (19)

```
SET CONSTRAINTS { constraint_name_commalist | ALL }
        { DEFERRED | IMMEDIATE }
```

- Если в качестве желаемого режима проверки ограничений задано DEFERRED,
  - то все указанные ограничения переводятся в режим отложенной проверки
- Если в качестве желаемого режима проверки ограничений задано IMMEDIATE,
  - то все указанные ограничения переводятся в режим немедленной проверки
- Если хотя бы одно из этих ограничений не удовлетворяется, то
  - операция SET CONSTRAINTS отвергается,
  - и все указанные ограничения остаются в предыдущем режиме.
- При выполнении операции COMMIT неявно выполняется операция SET CONSTRAINTS ALL IMMEDIATE
- Если эта операция отвергается, то COMMIT срабатывает как ROLLBACK