**6. Реляционная модель данных. Взаимосвязь отношений. Ключи. Правило ссылочной**

Целостности

Согласно Дейту, реляционная модель состоит из трех частей:

-Структурной части.

-Целостной части.

-Манипуляционной части.

***Структурная часть*** описывает, какие объекты рассматриваются реляционной моделью. Постулируется, что единственной структурой данных, используемой в реляционной модели, являются нормализованные n-арные отношения.

***Целостная часть*** описывает ограничения специального вида, которые должны выполняться для любых отношений в любых реляционных базах данных. Это ***целостность сущностей*** и ***целостность внешних ключей***.

***Манипуляционная часть*** описывает два эквивалентных способа манипулирования реляционными данными - ***реляционную алгебру*** и ***реляционное исчисление***.

**Взаимосвязь отношений** является важнейшим элементом реляционной модели данных. Она поддерживается ***внешними ключами* (*foreign key*)**.

Когда поле одной таблицы является внешним ключом другой таблицы, то такие таблицы называются **связанными**. А процедуру установления внешних ключей между таблицами называют **связывание таблиц**.

Когда пользователю необходимо найти информацию о всех сотрудников, принадлежащих определенной кафедре, он сначала находит кафедру (по значению поля *ИД\_Каф*) в таблице **Кафедры**, затем в таблице **Универистет** находятся все строки, у которых значение в столбце *ИД\_Каф* равно найденному идентификационному номеру кафедры. Будет получено несколько записей, в которых значение *ИД\_Каф* будет то, которое ищем. Последним этапом будет получение информации о сотрудниках, полученных для нужной кафедры в таблице **Сотрудники**.

В нашем примере в качестве первичного ключа везде использовались специализированные поля (*ИД\_???*), которые добавляются в качестве первичного (по умолчанию) всеми СУБД. Однако это можно изменить, и использовать, скажем, название кафедры, или Фамилию сотрудника, но в этом случае возникли бы проблемы с изменением информации. Кроме того, размер первичного ключа, который необходимо хранить как в головной, так и в зависимой таблице, при этом получается довольно большим.

Таким образом, в теории реляционных БД **для внешнего ключа характерны следующие свойства**:

1. Каждое значение атрибута внешнего ключа должно являться значением соответствующего потенциального ключа. Причем обратная ситуация необязательно.

2. Количество атрибутов внешнего ключа должно в точности соответствовать количеству атрибутов первичного ключа. Следствием является то, что внешний ключ будет составным (т.е. состоять более чем из одного атрибута) только тогда, когда соответствующий первичный ключ так же является составным. Соответственно, внешний ключ будет простым только в том случае, если соответствующий первичный ключ является простым.

3. Для внешнего ключа не требуется, чтобы он был компонентом первичного ключа в отношении, к которому он принадлежит. Т.о. значение во внешнем ключе могут повторяться, что не возможно для первичного ключа. Например, в одной и той же …

4. Каждый атрибут, входящий в данный внешний ключ, должен быть определен на том же домене, что и соответствующий атрибут соответствующего первичного ключа.

5. Для атрибутов внешнего ключа разрешается иметь значения NULL

**Правило ссылочной целостности**

БД не должна содержать значений внешних ключей, для которых не существует соответствующих значений первичных ключей.

**Любое состояние БД, *не удовлетворяющее правилу ссылочной целостности*, считается *некорректным***.

1. Ограничить (RESTRICT) операции, приводящие к нарушению этого правила;

2. Допустить выполнение таких операций, но при этом выполнять дополнительные действия, исправляющие эту ситуацию (каскадирование CASCADE);

3. Давать пользователю выбирать либо 1, либо 2.

***Дополнительными стратегиями*** поддержания ссылочной целостности являются:

4. Установить в состояние не определено(SET NULL);

5. Установить в состояние по умолчанию (SET DEFAULT).

В реальных СУБД можно также отказаться от использования какой-либо стратегии поддержания ссылочной целостности:

6. Игнорировать (IGNORE) - выполнять операции, не обращая внимания на нарушения ссылочной целостности.

7. Пользователь может разработать свою уникальную стратегию поддержания ссылочной целостности.

Для примера рассмотрим случай, когда пользователь пытается удалить строки таблицы **Кафедры** (рис 9) со значением К3125 первичного ключа. На это значение ссылается строка таблицы **Университеты**.

Т.о. во-первых, можно вообще запретить удаление этой строки, до тех пор, пока на нее есть ссылки из других таблиц. Удалив все строки со ссылками из таблицы **Университеты**, можно после этого удалять соответствующую строку из таблицы **Кафедры**.

Во-вторых, можно разрешить системе при удалении значения из таблицы **Кафедры** строки со значением К3125 автоматически удалять из таблицы **Университеты** все строки, ссылающиеся на это значение строки. Такая операция называется операцией **каскадного удаления**.

К аналогичным рассуждениям можно придти, если рассматривать операции изменения первичных ключей в таблице, на которые есть ссылки внешних ключей. В этом случае

1) запрещать редактировать строки с первичным ключом, пока есть внешние ссылки;

2) разрешать изменять, но при этом выполнять операции **каскадного изменения**;

3) соответствующие значения внешних ключей принимают значения NULL.

**Первичные ключи**

Любое отношение должно иметь один или несколько атрибутов, значения которых однозначно идентифицируют его кортеж. Такой атрибут (или комбинация атрибутов) называется **первичным ключом**. Для нашего примера это атрибут «пор\_номер». При этом остальные атрибуты будут считаться **альтернативными ключами**.

Первичные ключи должны удовлетворять требованиям:

а) уникальность, т.е. в каждый момент времени никакие два кортежа отношения не должны иметь одинаковых комбинаций значений, входящих в ключ атрибутов.

б) неизбыточность т.е. ни один из входящих в ключ атрибутов не м.б. исключен из ключа без нарушения правила уникальности.

Замечание (!) На практике выполнение первого требования обязательно. Для второго допустимо нарушение.

Правило ***целостности объектов*** (следует из требования а) и из соображений реального мира. Нельзя идентифицировать объект по значению NULL )

**В первичном ключе не м.б. отношений со значением NULL!!!**

В реляционной модели первичные ключи выполняют функцию уникальной идентификации (т.е. служат для предоставления необходимых признаков по идентификации объектов). И если каким-либо значением первичного ключа окажется значение NULL, то можно будет сказать, что существует реальный объект, который не имеет опознавательных признаков, что в действительности бессмысленно.