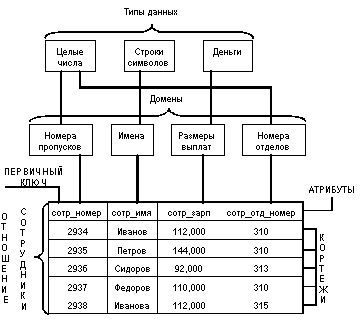
**Тема 2. Реляционная модель данных. Специальные операции реляционной алгебры.**

**Основные понятия реляционной модели**

Основными понятиями реляционных баз данных являются тип данных, домен, атрибут, кортеж, первичный ключ и отношение.

Для начала покажем смысл этих понятий на примере отношения СОТРУДНИКИ, содержащего информацию о сотрудниках некоторой организации:



**Тип данных**

Понятие тип данных в реляционной модели данных полностью адекватно понятию типа данных в языках программирования. Обычно в современных реляционных БД допускается хранение символьных, числовых данных, битовых строк, специализированных числовых данных (таких как "деньги"), а также специальных "темпоральных" данных (дата, время, временной интервал). Достаточно активно развивается подход к расширению возможностей реляционных систем абстрактными типами данных (соответствующими возможностями обладают, например, системы семейства Ingres/Postgres). В нашем примере мы имеем дело с данными трех типов: строки символов, целые числа и "деньги".

**Домен**

Понятие домена более специфично для баз данных, хотя и имеет некоторые аналогии с подтипами в некоторых языках программирования. В самом общем виде домен определяется заданием некоторого базового типа данных, к которому относятся элементы домена, и произвольного логического выражения, применяемого к элементу типа данных. Если вычисление этого логического выражения дает результат "истина", то элемент данных является элементом домена.

Наиболее правильной интуитивной трактовкой понятия домена является понимание домена как допустимого потенциального множества значений данного типа. Например, домен "Имена" в нашем примере определен на базовом типе строк символов, но в число его значений могут входить только те строки, которые могут изображать имя (в частности, такие строки не могут начинаться с мягкого знака).

Следует отметить также семантическую нагрузку понятия домена: данные считаются сравнимыми только в том случае, когда они относятся к одному домену. В нашем примере значения доменов "Номера пропусков" и "Номера групп" относятся к типу целых чисел, но не являются сравнимыми. Заметим, что в большинстве реляционных СУБД понятие домена не используется, хотя в Oracle V.7 оно уже поддерживается.

**Схема отношения, схема базы данных**

Схема отношения - это именованное множество пар {имя атрибута, имя домена (или типа, если понятие домена не поддерживается)}. Степень или "арность" схемы отношения - мощность этого множества. Степень отношения СОТРУДНИКИ равна четырем, то есть оно является 4-арным. Если все атрибуты одного отношения определены на разных доменах, осмысленно использовать для именования атрибутов имена соответствующих доменов (не забывая, конечно, о том, что это является всего лишь удобным способом именования и не устраняет различия между понятиями домена и атрибута).

Схема БД (в структурном смысле) - это набор именованных схем отношений.

**Кортеж, отношение**

Кортеж, соответствующий данной схеме отношения, - это множество пар {имя атрибута, значение}, которое содержит одно вхождение каждого имени атрибута, принадлежащего схеме отношения. "Значение" является допустимым значением домена данного атрибута (или типа данных, если понятие домена не поддерживается). Тем самым, степень или "арность" кортежа, т.е. число элементов в нем, совпадает с "арностью" соответствующей схемы отношения. Попросту говоря, кортеж - это набор именованных значений заданного типа.

Отношение - это множество кортежей, соответствующих одной схеме отношения. Иногда, чтобы не путаться, говорят "отношение-схема" и "отношение-экземпляр", иногда схему отношения называют заголовком отношения, а отношение как набор кортежей - телом отношения. На самом деле, понятие схемы отношения ближе всего к понятию структурного типа данных в языках программирования. Было бы вполне логично разрешать отдельно определять схему отношения, а затем одно или несколько отношений с данной схемой.

Однако в реляционных базах данных это не принято. Имя схемы отношения в таких базах данных всегда совпадает с именем соответствующего отношения-экземпляра. В классических реляционных базах данных после определения схемы базы данных изменяются только отношения-экземпляры. В них могут появляться новые и удаляться или модифицироваться существующие кортежи. Однако во многих реализациях допускается и изменение схемы базы данных: определение новых и изменение существующих схем отношения. Это принято называть эволюцией схемы базы данных.

Обычным житейским представлением отношения является таблица, заголовком которой является схема отношения, а строками - кортежи отношения-экземпляра; в этом случае имена атрибутов именуют столбцы этой таблицы. Поэтому иногда говорят "столбец таблицы", имея в виду "атрибут отношения". Когда мы перейдем к рассмотрению практических вопросов организации реляционных баз данных и средств управления, мы будем использовать эту житейскую терминологию. Этой терминологии придерживаются в большинстве коммерческих реляционных СУБД.

Реляционная база данных - это набор отношений, имена которых совпадают с именами схем отношений в схеме БД.

Как видно, основные структурные понятия реляционной модели данных (если не считать понятия домена) имеют очень простую интуитивную интерпретацию, хотя в теории реляционных БД все они определяются абсолютно формально и точно.

**Основы реляционной алгебры**

Реляционная алгебра базируется на теории множеств и является основой логики работы баз данных. Для начала введем понятие реляционной базы данных, в которой будем выполнять все действия.

Реляционной базой данных называется совокупность отношений, содержащих всю информацию, которая должна хранится в базе. В данном определении нам интересен термин отношение, но пока оставим его без строго определения.

Лучше представим себе таблицу продуктов.

таблица PRODUCTS

|  |  |  |  |
| --- | --- | --- | --- |
| ID | NAME | COMPANY | PRICE |
| 123 | Печеньки | ООО ”Темная сторона” | 190 |
| 156 | Чай | ООО ”Темная сторона” | 60 |
| 235 | Ананасы | ОАО ”Фрукты” | 100 |
| 623 | Томаты | ООО ”Овощи” | 130 |

Таблица состоит из 4-х строк, строка в таблице является кортежем в реляционной теории. Множество упорядоченных кортежей называется отношением.

Перед тем как дать определение отношения, введем еще один термин — домен. Домены применительно к таблице это столбцы.

Для ясности, теперь введем строгое определение отношения.

Пусть даны N множеств D1,D2, …. Dn (домены), отношением R над этими множествами называется множество упорядоченных N-кортежей вида <d1,d1,...dn>, где d1 принадлежит D1 и тд. Множества D1,D2,..Dn называются доменами отношения R.

Каждый элемент кортежа представляет собой значение одного из атрибутов, соответствующего одному из доменов.

**Ключи в отношениях**

В отношении требованием является то, что все кортежи должны различаться. Для однозначной идентификации кортежа существует первичный ключ. Первичный ключ это атрибут или набор из минимального числа атрибутов, который однозначно идентифицирует конкретный кортеж и не содержит дополнительных атрибутов.

Подразумевается, что все атрибуты в первичном ключе должны быть необходимыми и достаточными для идентификации конкретного кортежа, и исключение любого из атрибутов в ключе сделает его недостаточным для идентификации.

Например, в такой таблице ключом будет сочетание атрибутов из первого и второго столбца.

таблица DRIVERS

|  |  |
| --- | --- |
| COMPANY | DRIVER |
| ООО ”Темная сторона” | Владимир |
| ООО ”Темная сторона” | Михаил |
| ОАО ”Фрукты” | Руслан |
| ООО ”Овощи” | Владимир |

Видно, что в организации может быть несколько водителей, и чтобы однозначно идентифицировать водителя необходимо и значение из столбца “Название организации” и из “Имя водителя”. Такой ключ называется составным.

В реляционной БД таблицы взаимосвязаны и соотносятся друг с другом как главные и подчиненные. Связь главной и подчиненной таблицы осуществляется через первичный ключ (primary key) главной таблицы и внешний ключ ( foreign key ) подчиненной таблицы.

Внешний ключ это атрибут или набор атрибутов, который в главной таблице является первичным ключом.

Этой подготовительной теории будет достаточно для знакомства с основными операциями реляционной алгебры.

**Операции реляционной алгебры**

Основные восемь операций реляционной алгебры были предложены Э.Коддом.

* Объединение
* Пересечение
* Вычитание
* Декартово произведение
* Выборка
* Проекция
* Соединение

Первая половина операций аналогична таким же операциям над множествами. Часть операций можно выразить через другие операции. Рассмотрим большую часть операций с примерами.

Для понимания важно запомнить, что результатом любой операции алгебры над отношениями является еще одно отношение, которое можно потом так же использовать в других операциях.

Создадим еще одну таблицу, которая нам пригодится в примерах.

таблица SELLERS

|  |  |
| --- | --- |
| ID | SELLER |
| 123 | OOO “Дарт” |
| 156 | ОАО ”Ведро” |
| 235 | ЗАО “Овоще База” |
| 623 | ОАО ”Фирма” |

Условимся, что в этой таблице ID это внешний ключ, связанный с первичным ключом таблицы PRODUCTS.

Для начала рассмотрим самую простую операцию — имя отношения. Её результатом будет такое же отношение, то есть выполнив операцию PRODUCTS, мы получим копию отношения PRODUCTS.

**Проекция**

Проекция является операцией, при которой из отношения выделяются атрибуты только из указанных доменов, то есть из таблицы выбираются только нужные столбцы, при этом, если получится несколько одинаковых кортежей, то в результирующем отношении остается только по одному экземпляру подобного кортежа.

Для примера сделаем проекцию на таблице PRODUCTS выбрав из нее ID и PRICE.

Синтаксис операции:

π(ID, PRICE) PRODUCTS

В результате этой операции получим отношение:

|  |  |
| --- | --- |
| ID | PRICE |
| 123 | 190 |
| 156 | 60 |
| 235 | 100 |
| 623 | 130 |

**Выборка**

Выборка — это операция, которая выделяет множество строк в таблице, удовлетворяющих заданным условиям. Условием может быть любое логическое выражение.

Для примера сделаем выборку из таблицы с ценой больше 90.

Синтаксис операции:

σ(PRICE>90) PRODUCTS

|  |  |  |  |
| --- | --- | --- | --- |
| ID | NAME | COMPANY | PRICE |
| 123 | Печеньки | ООО ”Темная сторона” | 190 |
| 235 | Ананасы | ОАО ”Фрукты” | 100 |
| 623 | Томаты | ООО ”Овощи” | 130 |

В условии выборки мы можем использовать любое логическое выражение. Сделаем еще одну выборку с ценой больше 90 и ID товара меньше 300:

σ(PRICE>90 ^ ID<300) PRODUCTS

|  |  |  |  |
| --- | --- | --- | --- |
| ID | NAME | COMPANY | PRICE |
| 123 | Печеньки | ООО ”Темная сторона” | 190 |
| 235 | Ананасы | ОАО ”Фрукты” | 100 |

Совместим операторы проекции и выборки. Мы можем это сделать, потому что любой из операторов в результате возвращает отношение и в качестве аргументов использует также отношение.

Из таблицы с продуктами выберем все компании, продающие продукты дешевле 110.

πCOMPANYσ(PRICE<100 ) PRODUCTS

|  |
| --- |
| COMPANY |
| ООО ”Темная сторона” |
| ОАО ”Фрукты” |

**Умножение**

Умножение или декартово произведение является операцией, производимой над двумя отношениями, в результате которой мы получаем отношение со всеми доменами из двух начальных отношений. Кортежи в этих доменах будут представлять из себя все возможные сочетания кортежей из начальных отношений. На примере будет понятнее.

Получим декартово произведения таблиц PRODUCTS и SELLERS.

Синтаксис операции:

PRODUCTS × SELLERS

Можно заметить, что у двух этих таблиц есть одинаковый домен ID. В подобной ситуации домены с одинаковыми названиями получают префикс в виде названия соответствующего отношения, как показано ниже.

Для краткости перемножим не полные отношения, а выборки с условием ID<235

(цветом выделены одни и те же кортежи)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PRODUCTS.ID | NAME | COMPANY | PRICE | SELLERS.ID | SELLER |
| 123 | Печеньки | ООО ”Темная сторона” | 190 | 123 | OOO “Дарт” |
| 156 | Чай | ООО ”Темная сторона” | 60 | 156 | ОАО ”Ведро” |
| 123 | Печеньки | ООО ”Темная сторона” | 190 | 156 | ОАО ”Ведро” |
| 156 | Чай | ООО ”Темная сторона” | 60 | 123 | OOO “Дарт” |

Для примера использования этой операции представим себе необходимость выбирать продавцов с ценами меньше 90. Без произведения необходимо было бы сначала получить ID продуктов из первой таблицы, потом по этим ID из второй таблицы получить нужные имена SELLER, а с использованием произведения будет такой запрос:

π(SELLER) σ(RODUCTS.ID=SELLERS.ID ^ PRICE<90) PRODUCTS × SELLERS

В результате этой операции получим отношение:

|  |
| --- |
| SELLER |
| ОАО ”Ведро” |

**Соединение и естественное соединение**

Операция соединения обратна операции проекции и создает новое отношение из двух уже существующих. Новое отношение получается конкатенацией кортежей первого и второго отношений, при этом конкатенации подвергаются отношения, в которых совпадают значения заданных атрибутов. В частности, если соединить отношения PRODUCTS и SELLERS, этими атрибутами будут атрибуты доменов ID.

Также для понятности можно представить соединение как результат двух операций. Сначала берется произведение исходных таблиц, а потом из полученного отношения мы делаем выборку с условием равенства атрибутов из одинаковых доменов. В данном случае условием явлется равенство PRODUCTS.ID и SELLERS.ID.

Попробуем соединить отношения PRODUCTS и SELLERS и получим отношение.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PRODUCTS.ID | NAME | COMPANY | PRICE | SELLERS.ID | SELLER |
| 123 | Печеньки | ООО ”Темная сторона” | 190 | 123 | OOO “Дарт” |
| 156 | Чай | ООО ”Темная сторона” | 60 | 156 | ОАО ”Ведро” |
| 235 | Ананасы | ОАО ”Фрукты” | 100 | 235 | ЗАО “Овоще База” |
| 623 | Томаты | ООО ”Овощи” | 130 | 623 | ОАО ”Фирма” |

Натуральное соединение получает схожее отношение, но в случае, если у нас корректно настроена схема в базе ( в данном случае первичный ключ таблицы PRODUCTS ID связан с внешним ключем таблицы SELLERS ID), то в результирующем отношении остается один домен ID.

Синтаксис операции:

PRODUCTS ⋈ SELLERS;

Получится такое отношение:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PRODUCTS.ID | NAME | COMPANY | PRICE | SELLER |
| 123 | Печеньки | ООО ”Темная сторона” | 190 | OOO “Дарт” |
| 156 | Чай | ООО ”Темная сторона” | 60 | ОАО ”Ведро” |
| 235 | Ананасы | ОАО ”Фрукты” | 100 | ЗАО “Овоще База” |
| 623 | Томаты | ООО ”Овощи” | 130 | ОАО ”Фирма” |

**Пересечение и вычитание.**

Результатом операции пересечения будет отношение, состоящее из кортежей, полностью входящих в состав обоих отношений.

Результатом вычитания будет отношение, состоящее из кортежей, которые являются кортежами первого отношения и не являются кортежами второго отношения.

Данные операции аналогичны таким же операциям над множествам, так что, я думаю, нет необходимости подробно их расписывать.