Лекция 8

Операция Деление

Делимое и делитель

|  |  |  |
| --- | --- | --- |
| A | B | C |
| A | С | D |
| B | С | D |
| A | P | Q |
| b | X | y |

|  |
| --- |
| A |
| A |
| b |

Мы хотим получить отношение, содержащее номера складов и всё, что к ним относится, на которых можно получить интересующие нас запчасти с одного склада полным комплектом.

Частное и остаток

|  |  |
| --- | --- |
| b | c |
| c | d |

|  |  |  |
| --- | --- | --- |
| A | B | C |
| a | p | q |
| b | x | y |

A=b\*c+в

Операция реляционного деления обозначается DIVIDE BY, имеет два операнда:

Бинарный операнд с заголовком A {a1, a2, …, an; b1, b2, …, bm}.

Унарный операнд B {b1, b2, …, bm}.

Составные атрибуты отношения A и отношения B обладают одним и тем же именем и определены на одном и том же домене. При этом составной атрибут B является подмножеством составного атрибута AB

Определение операции деления:

Результатом операции деления является набор кортежей отношения R, определённого на множестве, которые определены на множестве атрибутов C, равном A-B, которые соответствуют комбинации всех кортежей отношения делителей.

C=(A+B) - b

Результирующее отношение состоит из унарных кортежей, включающих значение первого атрибута кортежей первого операнда таких, что множество значений второго атрибута при фиксированном значении первого включает множество значений второго операнда.

Операцию деления можно представить через другие операции

1) Проекция (T1 <- Пс(R)

2) Непонятная хрень, которую Гусарова не может назвать (T2 <- Пс((S\*T1) -R)

3) Разность T3 <- T1-T2

8 операторов Кодда, которые мы рассмотрели, не представляют минимального набора операторов, фактически имеются пять первичных (примитивных) операция. Это выборка, проекция, произведение, объединение и вычитание. Остальные операции из набора Кодда (соединение, пересечение, деление) можно определить через пять примитивных. Например, естественное соединение – это проекция выборки соединения.

Однако в силу популярности последних операций имеет смысл обеспечить их непосредственную поддержку, то есть рассматривать их как самостоятельные операции. Тем не менее постоянно предлагаются новые операция в рамках SQL и постоянно делаются замены.

Вычислительные методы реляционной алгебры.

Введение любой алгебры, в том числе реляционной, предполагает не просто содержательное применение отдельных операций, но и возможность составления и решения соответствующих уравнений.

Иначе говоря, каждую таблицу можем записать в виде функциональной зависимости, а всю базу в виде системы функциональных зависимостей. В большинстве такая система не решается, а используются итерационные методы вычислительной математики.

Кроме непосредственного решения система реляционных уравнений существует путь оптимизации запросов и транзакций, выполняемых в БД.

Существуют стратегии преобразования выражений реляционной алгебры, которые позволяют выполнять запросы оптимально, то есть с минимальными затратами времени и вычислительной мощности. Стратегии являются эвристическими.

Эвристика – это строго недоказуемое, но в большинстве случаев реализуемое правило.

Основное эмпирическое правило – максимально возможное сокращение результирующего набора перед выполнением бинарных операций.

Некоторые эвристики, применяемые для обработки запросов.

1) Выполнение операции выборки на самых ранних этапах обработки

2) Объединение в одну операцию соединения операция декартовое произведение и следующей за ней выборки, предикат который представляет условие соединения.

Операция выборки типа тета-соединение и операция декартовое произведение могут быть представлены в виде тета-соединения.

3) Переупорядочивание бинарных операций таким образом, чтобы операции с наиболее ограничительными выборками выполнялись в первую очередь.

4) Как можно более раннее выполнение операций проекций.

5) Однократное вычисление общих выражений

Если одно и тоже выражение встречается в дереве реляционной алгебре несколько раз и результат его вычисления не слишком велик, то можно сохранить этот результат с целью повторного использования. Однако выигрыш получается только в том случае, если размер результирующей таблицы общего выражения помещается в оперативной памяти, либо стоимость его выборки внешней памяти меньше стоимости его повторного вычисления.

Подход эффективен при работе с view (представление).

Обзор основных конструкций языка SQL

SQL – непроцедурный язык, построенный на использовании английского языка.

Оператор SELECT, основной оператор, объединяет три оператора, а именно выборку, проекцию и соединение. При выполнении оператора SELECT создаётся результирующая таблица, содержащая 1 или больше столбцов, 0 и больше строк.

В списке выборки SELECT указываются столбцы или вычисляемые поля, которые дальше присутствуют в результирующей таблице.

В конструкции FROM перечисляются все таблицы и view, доступ к которым необходим для извлечения данных из столбцов, заявленных в списке выборки.

Конструкция WHERE используется для отбора строк данных, которые помещаются в результирующую таблицу. Отбор осуществляется посредствам проверки заданных условий поиска для каждой из строк указанной таблицы.

Конструкция ORDER BY упорядочивает строки результирующей таблицы по значению одного или нескольких столбцов. Место конструкции в конце.

Агрегирующие функции:

Count, sum, avg, min, max

Каждая из которых использует как параметр значение всех элементов указанного столбца и возвращает одно.

В одной конструкции SELECT не допускается смешивать и агрегирующие функции, и имена столбцов за исключением конструкции GROUP BY.

В конструкции GROUP BY строки, которые имеют одно и то же значение в одном или нескольких столбцах объединяются и рассматриваются как исходная информация для агрегирующих функций. В этом случае агрегирующие функции вычисляют единственные значения для каждой группы.

Конструкция HAVING применительна к группам, выполняет те же функции, что и конструкции WHERE в строках, однако в отличие от WHERE в конструкциях HAVING могут использоваться агрегирующие функции

Подзапросы – завершённый оператор SELECT, встроенный в тело другого запроса. Вложенный запрос может помещаться в конструкции WHERE или HAVING внешнего SELECT. Конструкция является рекурсивной. Концептуально в результате выполнения подзапроса создаётся временная таблица, содержимое которой доступно внешнему SELECT.

Имеются три типа подзапросов:

• Скалярные - возвращает таблицу в виде одной строки и одного столбца.

• Строковые – несколько столбцов и одну строку \*чаще всего в предикатах\*

• Табличные – возвращает несколько столбцов и несколько строк. \*Можно использовать в качестве операнда для предиката in\*

Если столбцы результирующей таблицы выбираются из нескольких таблиц, то для последней нужно использовать операция соединения. Имена таблиц через FROM, столбцы в конструкции WHERE.

Помимо языка SQL, существуют операторы языка DATA DEFINITION LANGUAGE, которые позволяются создавать, удалять, модифицировать таблицы.

Обзор процедурных расширений языка SQL

Уточним терминологию:

Внутренний язык СУБД для работы с данными состоит из двух частей:

1) Язык определения данных DDL для определения схемы базы данных.

2) Язык манипулирования данных DML для управления данным, находящимися в базе. В том числе вставки в базу, модификация хранимых сведений, извлечение и удаление сведений. Языки DML имеют разные базовые конструкции.

Процедурные и не процедурные.

Процедурные позволяются указать, как именно получить результат оператора языка. Обычно операторы DML встраиваются в программу на языке высокого уровня, которая содержит конструкции для обеспечения циклической обработки и перехода к другим участкам кода. Такие языки употребляются в сетевых и иерархических СУБД.

Непроцедурные описывают только то, какой именно результат будет получен. Такой подход освобождает пользователя от необходимости знать подробности внутренней реализации структуры данных и особенности конкретных алгоритмов извлечения и преобразования данных.

Часть непроцедурного языка DML, которая отвечает за извлечение данных, называется языком запросов. То есть термин запрос обозначает оператор извлечения данных, выраженный с помощью языка запроса.

Чистый SQL является непроцедурным языком, но часто требуется добавить к нему некоторые возможности процедурных расширений.