**Лекция 5:**

1. Декартово соединение таблиц. INNER JOIN
2. Задача на рейтинг
3. Построение гистограммы

**Задание 1:**

В приложении к архиву есть краткая инструкция, как создавать схему данных в Access или MS SQL. Если это Вам интересно (включая поведение таблиц в случае создания разных видов ссылочной целостности).

## Типы соединения таблиц:

**Операции реляционной алгебры:**

1. Объединение, пересечение, разность и декартово произведение
2. Выборка, проекция, соединение, деление

**Выборка:** возвращает отношение, содержащее кортежи из заданного отношения, причем данные удовлетворяют указанным условиям. (SELECT \* WHERE)

**Проекция:** Возвращает отношение, содержащее все кортежи (подкортежи) заданного отношения, которые остались в этом отношении после исключения из него некоторых атрибутов. (SELECT Поле1)

**Произведение:** Возвращает отношение, содержащее все возможные кортежи, которые являются сочетанием двух кортежей, принадлежащих соответственно двум заданным отношениям. (SELECT \* FROM A,B – декартово произведение)

Объединение: **Возвращает отношение, содержащее все кортежи, которые принадлежат либо одному из двух заданных отношений, либо им обоим. (UNION)**

**Пересечение:** Возвращает отношение, содержащее все кортежи, которые принадлежат одновременно двум заданным отношениям.

**Разность:** Возвращает отношение, содержащее все кортежи, которые принадлежат первому из двух заданных отношений и не принадлежат второму.

**Соединение:** Возвращает отношение, содержащее все возможные кортежи, которые представляют собой комбинацию атрибутов двух кортежей, принадлежащих двум заданным отношениям, при условии, что в этих двух комбинируемых кортежах присутствуют одинаковые значения в одном или нескольких общих для исходных отношений атрибутах (причем эти общие значения в результирующем кортеже появляются один раз, а не дважды). (SELECT FROM INNER JOIN)

**Деление:** Для заданных двух унарных отношений и одного бинарного возвращает отношение, содержащее все кортежи из первого унарного отношения, которые содержатся также в бинарном отношении и соответствуют всем кортежам во втором унарном отношении.

**Реализация в SQL:**

***Замечание:***

В запросах обращаться можно не только к таблицам, но и другим запросам на выборку данных. Обращение происходит точно также.

Что делать, если нужно использовать данные из нескольких таблиц? Таблицы нужно соединять, объединять и т.д.

**1. Операции соединения. Типы соединений:**

Соединение таблиц в MS Access производится с помощью операций: JOIN (декартово произведение, CROSS JOIN, «,»), INNER JOIN, LEFT JOIN, RIGHT JOIN (а также FULL JOIN в MS SQL Server).

**Декартово произведение:**

Если Вы указываете таблицы через запятую, значит, что каждая строка одной таблицы соединиться с каждой строкой второй таблицы – декартово произведение строк из одной таблицы на строки в другой таблицы. Если Вы на самом деле хотели соединить иначе, то нужно явно прописать по какому условию Вы соединяете (то есть по какому условию из декартового произведения нужно оставить (отфильтровать) только нужные Вам строки):

DECLARE @T TABLE (id int primary key, val int)

INSERT INTO @T

VALUES (1, 2), (2, 4), (3, 6)

SELECT \*

FROM @T T1, @T T2

SELECT \*

FROM @T T1, @T T2

WHERE T1.id = T2.id

Обратите внимание, что здесь мы соединили таблицу саму с собой. Но можно делать декартово произведение между собой и разных таблиц. Самое главное – не забывать о том, что такое декартово произведение.

**Задача 2:**

Написать запрос, который выводит 3 числа, которые в сумме дают заданное.

DECLARE @param int

SET @param = 100

SELECT T1.val, T2.val, T3.val

FROM T as T1, T as T2, T as T3

WHERE T1.val != T2.val AND T1.val != T3.val AND T2.val != T3.val AND T1.val + T2.val + T3.val = @param

Как изменить запрос, чтобы тройки чисел были уникальными (чтобы не встречались тройки вида {97, 1, 2}, {97, 2, 1}, {1, 2, 97} и т.д.)?

**Задача на рейтинг:**

У Вас есть таблица: Обороты: **Покупатель**, Сумма. Нужно для каждого покупателя вывести его рейтинг. Покупатель с максимальной суммой имеет минимальный рейтинг и т.д. Как написать такой запрос?

|  |  |
| --- | --- |
| **Покупатель\_ID** | **Сумма** |
| 1 | 100 |
| 2 | 50 |
| 3 | 40 |
| 4 | 30 |

SELECT T1.Покупатель\_ID, T1.Сумма, COUNT(T2.Покупатель\_ID) Рейтинг

FROM Обороты T1 INNER JOIN Обороты T2 ON

T1.Сумма <= T2.Сумма

GROUP BY T1.Покупатель\_ID, T1.Сумма

Подумайте, что выведет запрос, если покупатель 3 тоже закупился на 50 у.е.?

Чуть позже разберем специальные функции, которые позволяют нумеровать строки. Пока наша цель - научиться решать такие задачи простейшими доступными средствами.

**Задача: Построение гистограммы:**

У Вас есть таблица: для каждой частицы скорость, с которой она пролетела мимо датчика. Необходимо при помощи SQL получить данные для построения гистограммы.

|  |  |
| --- | --- |
| **N** | **V** |
| 1 | 10 |
| 2 | 12 |
| 3 | 13 |
| 4 | 12 |
| 5 | 11 |

Что такое гистограмма распределения скоростей: для каждой скорости мы должны вычислить, сколько у нас частиц, которые пролетели с такой скоростью.

В чем минусы следующего запроса?

DECLARE @Data TABLE (N int, V float)

SELECT V, COUNT(\*)

FROM @Data

GROUP BY V

V – float, а значит различных значений скоростей может быть достаточно много. И такая гистограмма будет непоказательна. Нужно разбить скорости по корзинам (обычно одинаковой ширины) и уже для корзин посчитать, сколько частиц попало в каждую корзину.

Например, вот так:

DECLARE @h float

SET @h = 2

SELECT FLOOR(V/@h)\*@h B, FLOOR(V/@h + 1)\*@h E, COUNT(\*)

FROM @Data

GROUP BY FLOOR(V/@h)\*@h, FLOOR(V/@h + 1)\*@h

Здесь регулируется ширина корзины. Подумайте, а как изменить этот запрос, чтобы можно было задавать не шаг для гистограммы, а количество «корзин» для статистики?