Лекция 2

ОСНОВЫ РЕЛЯЦИОННОЙ АЛГЕБРЫ

Глоссарий (1)

Реляционная алгебра — замкнутая система операций над отношениями в реляционной модели данных.

Отношение: n-арным отношением (отношением степени n) называют подмножество декартова произведения множеств, не обязательно различных. Исходные множества называют в модели *доменами* (в СУБД – *множество значений, определяемых типом данных*).
Графическая интерпретация отношения — таблица, столбцы (поля, атрибуты) которой соответствуют вхождениям доменов в отношение, а строки (записи) — наборам из значений, взятых из исходных доменов. Число строк (кортежей) называют *кардинальным числом отношения (кардинальностью)*, или *мощностью* отношения.

Свойства отношения:
1. Нет двух одинаковых кортежей.
2. Атрибуты не упорядочены.
3. Значения атрибутов атомарны.
4. Порядок кортежей произвольный.

Глоссарий (2)

Реляционная модель данных (РМД) — логическая модель данных, прикладная теория построения баз данных, которая является приложением к задачам обработки данных таких разделов математики как теории множеств и логика первого порядка. На реляционной модели данных строятся реляционные базы данных.

Логика первого порядка (*исчисление предикатов*) — формальное исчисление, допускающее высказывания относительно переменных, фиксированных функций и предикатов.

Реляционная модель данных включает:

1. Структурный аспект — данные в БД представляют собой набор отношений.
2. Аспект целостности — отношения отвечают определенным условиям целостности.
3. Аспект обработки (манипулирования) — РМД поддерживает операторы манипулирования отношениями (реляционная алгебра, реляционное исчисление)

+

теория нормализации

Основные определения логики первого порядка

Язык логики первого порядка строится на основе сигнатуры, состоящей из множества функциональных символов **F** и множества предикатных символов **P**. С каждым функциональным и предикатным символом связана арность - число возможных аргументов.

Используются следующие дополнительные символы:

- Символы переменных (обычно **x,y,x1,y1** и т. д.),

Пропозициональные связки: $\wedge, \vee, \rightarrow, \neg, \equiv$.

Кванторы: всеобщности **∀** и существования **∃**,

- Служебные символы: скобки и запятая.

Перечисленные символы вместе с символами из и образуют *Алфавит* **логики первого порядка.**

Пример

“Все студенты сдают экзамены”,

“Некоторые студенты сдают экзамены на отлично”.

Введем предикаты:

P – «сдавать экзамены»

Q – «сдавать экзамены на отлично».

Предметная область данных предикатов представляет собой множество студентов.

Тогда исходные выражения примут вид:

$(\forall x)\ P(x)$

$(\exists x)\ Q(x)$

Реляционная алгебра

Отношение характеризуется схемой (заголовком) и набором кортежей (телом или расширением).

Заголовок отношения представляет собой множество пар <имя-атрибута:имя-домена>.

Число атрибутов в отношении называют *степенью* (или -арностью) отношения.

Тело отношения может изменяться: изменяются кортежи, добавляются новые и удаляются существующие → реляционная база данных — это множество изменяющихся во времени отношений.

Число кортежей отношения называют *мощностью* или *кардинальным числом* отношения.

Отношения *совместимы по типу*, если они имеют идентичные заголовки, а именно:

1) Отношения имеют одинаковое множество имен атрибутов, т.е. для любого атрибута в одном отношении найдется атрибут с таким же наименованием в другом отношении.
2) Атрибуты с одинаковыми именами определены на одних и тех же доменах.

Операции реляционной алгебры (1)

Основные восемь операций реляционной алгебры (предложены Э. Коддом):

1. Объединение
2. Пересечение
3. Вычитание
4. Декартово произведение
5. Выборка
6. Проекция
7. Соединение
8. Деление

теоретико-множественные операции

Специальные операции

Результат любой операции алгебры над отношениями – еще одно *отношение*, которое может участвовать в других операциях.

Операции реляционной алгебры (2)

Унарные:

1. Выборка
2. Проекция

Бинарные:

1. Объединение
2. Пересечение
3. Вычитание
4. Декартово произведение
5. Соединение
6. Деление

Замкнутость РА

Реляционная алгебра представляет собой набор таких операций над отношениями, что результат каждой из операций также является отношением.

Операции над одним отношением называются *унарными*, над двумя отношениями — *бинарными*, над тремя — *тернарными*.

N-арную реляционную операцию *f* можно представить функцией, возвращающей отношение и имеющей *n* отношений в качестве аргументов:

$R = f(R_1, R_2, \ldots, R_n)$

Поскольку реляционная алгебра является замкнутой, в качестве операндов в реляционные операции можно подставлять другие выражения реляционной алгебры (подходящие по типу):

$R = f(f_1(R_{11}, R_{12}, \ldots), f_2(R_{21}, R_{22}, \ldots), \ldots)$

Ограничения на операции

Некоторые операции (*объединение*, *пересечение* и *взятие разности)* требуют, чтобы отношения имели совпадающие (одинаковые) заголовки (схемы) → совпадают количество атрибутов, названия атрибутов и тип (домен) одноимённых атрибутов.

Теоретико-множественные операции (1)

Объединением (Union) двух отношений называется отношение, содержащее множество кортежей, принадлежащих либо первому, либо второму отношению.

$$R_1 = \{ r_1 \}, \quad R_2 = \{ r_2 \} \qquad R_1 \cup R_2 = \{ r \mid r \in R_1 \vee r \in R_2 \}$$

Пересечением (Intersect) отношений называется отношение, которое содержит множество кортежей, принадлежащих одновременно и первому и второму отношению.

$$R_1 \cap R_2 = \{ r \mid r \in R_1 \wedge r \in R_2 \}$$

Разностью (Minus) отношений называется отношение, содержащее множество кортежей, принадлежащих R_1 и не принадлежащих R_2 :

$$R_1 \setminus R_2 = \{ r \mid r \in R_1 \wedge r \notin R_2 \}$$

Теоретико-множественные операции (2)

Декартовым произведением (Times) отношения степени n со схемой и отношения степени m со схемой , содержащее кортежи, полученные сцеплением каждого кортежа r отношения с каждым кортежем q отношения

$$R_1 = \{ r \}, \quad R_2 = \{ q \} \qquad R_1 \otimes R_2 = \{(r,q) | r \in R_1 \wedge q \in R_2\}$$

Специальные операции реляционной алгебры (1)

Операция выбора (Select)/Ограничение, заданная на отношении R в виде булевского выражения, определенного на атрибутах отношения R, называется отношение, включающее те кортежи из исходного отношения, для которых истинно условие выбора:

$$R[\alpha(r)] \quad \{r \mid r \in R \ \wedge \alpha(r) = "Истина" \ \}$$

A WHERE c

Операция проектирования (Project)/Проекция или вертикального выбора называется отношение со схемой, соответствующей набору атрибутов B, содержащему кортежи, полученные из кортежей исходного отношения R путем удаления из них значений, не принадлежащих атрибутам из набора B.

A[X, Y, …, Z]

или

PROJECT A {x, y, …, z}

Специальные операции реляционной алгебры (2)

Операция соединения (Join) возвращает отношение, кортежи которого – это сочетание двух кортежей, имеющих общее значение для одного или нескольких общих атрибутов этих двух отношений.

```
(A TIMES B) WHERE c
```

Операция деления (Divide) возвращает отношение, содержащее все значения одного атрибута отношения, которые соответствуют (в другом атрибуте) всем значениям во втором отношении.

```
A DIVIDEBY B
```

Соединение (1)

Типы операции соединения:

- Общая операция соединения,
- θ-соединение (тэта-соединение),
- Эквисоединение.
- Естественное соединение.

Соединением отношений R_1 и R_2 по условию α называется отношение ($R_1 \otimes R_2$) WHERE α, представляющее собой логическое выражение, в которое могут входить атрибуты отношений R_1 и R_2 и (или) скалярные выражения. Т.е. операция соединения есть результат последовательного применения операций декартового произведения и выборки.

θ-соединением отношения R_1 по атрибуту a с отношением R_2 по атрибуту b называют отношение ($R_1 \otimes R_2$) WHERE a θ b, где θ – один из операторов сравнения (<, =, >, ≠, ≤, ≥). Краткая запись θ-соединения: R_1 [a θ b] R_2.

Соединение (2)

Эквисоединение есть частный случай θ-соединения, когда θ есть равенство: R_1 [a=b] R_2.

Соединение по равенству общих (одного или нескольких) атрибутов (от англ. слова *equal* — равный). Поскольку атрибуты, по которым производилось эквисоединение, включаются в результирующее отношение, потребуется операция переименования для разрешения конфликта имен. Степень производимого отношения будет равна сумме степеней исходных отношений.

Естественным соединением отношений R_1 (a,b) и R_2 (b,c) называется отношение R_1 JOIN R_2 со схемой (a,b,c) и телом, содержащим множество всех кортежей, полученных сцеплением кортежей операндов соединения по общим атрибутам. Т.е. это разновидность эквисоединения, из которого исключены дубликаты атрибутов, по которым оно производилось.

Композиция есть частный случай эквисоединения, из которого исключены атрибуты, по которым оно производилось.

Избыточность РА

Некоторые операторы реляционной алгебры выражаются через другие реляционные операторы.

Любая из операций объединения, пересечения, взятия разности может быть выражена через две других.

Оператор соединения определяется через операторы декартового произведения и выборки. Для оператора естественного соединения добавляется оператор проекции.

Оператор пересечения выражается через вычитание следующим образом:

```
A INTERSECT B = A MINUS(A MINUS B)
```

Оператор деления выражается через операторы вычитания, декартового произведения и проекции:

```
A DIVIDEBY B = A[X] MINUS((A[X] TIMES MINUS A)[X]
```

Общая интерпретация реляционных операций (1)

- Результатом выполнения операции объединения двух отношений является отношение, тело которого включает все кортежи, входящие хотя бы в одно из отношений-операндов.
- Результатом выполнения операции пересечения двух отношений является отношение, включающее все кортежи, входящие в оба отношения-операнда.
- Отношение, являющееся результатом взятия разности двух отношений, включает все кортежи, входящие в отношение-первый операнд, такие, что ни один из них не входит в отношение, являющееся вторым операндом.
- Операции объединения, пересечения и взятия разности возможны на совместимых по типу отношениях.
- Результатом декартова произведения двух отношений является отношение, кортежи которого являются конкатенацией (сцеплением) каждого кортежа первого отношения с каждым кортежем второго отношения. Кардинальное число результирующего отношения есть произведение кардинальных чисел отношений-операндов. Степень – сумма степеней отношений-операндов. Декартово произведение возможно выполнить на любых отношениях. Совместимость по типу не требуется.

Общая интерпретация реляционных операций (2)

- Результатом операции выбора отношения по некоторому условию является отношение, включающее кортежи отношения-операнда, удовлетворяющее этому условию. Таким образом, отношение “уменьшается по вертикали” – исключаются кортежи, не отвечающие заданному критерию.
- При выполнении проекции отношения на заданный набор его атрибутов производится отношение, кортежи которого получаются путем взятия соответствующих значений из кортежей отношения-операнда. В этом случае отношение “уменьшается по горизонтали” – сокращается число атрибутов.
- При соединении двух отношений по некоторому условию образуется результирующее отношение, кортежи которого являются конкатенацией кортежей первого и второго отношений и удовлетворяют этому условию.
- У операции реляционного деления два операнда – бинарное и унарное отношения. Результирующее отношение состоит из одноатрибутных кортежей, включающих значения первого атрибута кортежей первого операнда таких, что множество значений второго атрибута (при фиксированном значении первого атрибута) совпадает со множеством значений второго операнда.

Общая интерпретация реляционных операций (3)

- Операция переименования производит отношение, тело которого совпадает с телом операнда, но изменены имена атрибутов.
- Операция присваивания позволяет сохранить результат вычисления реляционного выражения в существующем отношении.

Примеры

Отношение *Поставщики*

Номер поставщика	Наименование поставщика
1	Иванов
2	Петров
3	Сидоров

Отношение *Детали*

Номер детали	*Наименование детали*
1	Болт
2	Гайка
3	Винт

Отношение *Поставки*

Номер поставщика	*Номер детали*	Поставляемое количество
1	*1*	100
1	*2*	200
1	*3*	300
2	*1*	150
2	*2*	250
3	*1*	1000

Получить список поставщиков, поставляющих деталь с номером 2

Последовательное выполнение операций РА:

1. Эквисоединение отношений *Поставки* и *Поставщики* по общему атрибуту *Номер поставщика*,
2. Операция выбора по условию *Номер детали = 2*
3. Проекция полученной выборки по атрибуту *Наименование поставщика:*

((DP JOIN P) WHERE DNUM = 2) [PNAME]

Получить список поставщиков, поставляющих по крайней мере одну гайку

“Поставляющих по крайней мере одну гайку”: эквисоединение выборки из отношения *Детали* по фильтру *Наименование детали = Гайка* и отношения *Поставки* по общему атрибуту *Номер детали* даст в результате не пустую выборку, к которой затем необходимо применить эквисоединение с отношением *Поставщики* по общему атрибуту *Номер поставщика* с последующей проекцией по атрибуту *Наименование поставщика:*

1. *((((D WHERE DNAME = Гайка) JOIN DP) JOIN P) [PNAME]*

Эквисоединение отношения *Детали* и отношения *Поставки* по общему атрибуту *Номер детали,* эквисоединение полученного результата с отношением *Поставщики* по общему атрибуту *Номер поставщика,* операция выбора по условию *Наименование детали = Гайка* и проекция по атрибуту *Наименование поставщика:*

2. *((((D JOIN DP) JOIN P) WHERE DNAME = Гайка) [PNAME]*

R_1(ФИО, ДОЛЖНОСТЬ) =

Иванов	инженер
Петров	дворник

R_2(ФИО, ВОЗРАСТ) =

Иванов	27
Петров	20

Запишите схему и расширение отношения S, которое есть результат декартова произведения отношений R_1 и R_2.

Результатом декартова произведения отношений R_1 степени n и R_2 степени m будет отношение степени n+m, содержащее кортежи, полученные сцеплением каждого кортежа отношения R_1 с каждым кортежем отношения R_2.

Кардинальное число (мощность) отношения S равна произведению кардинальных чисел отношений R_1 и R_2: $2 \times 2 = 4$.

ФИО	*ДОЛЖНОСТЬ*	*ФИО_1*	*ВОЗРАСТ*
Иванов	инженер	Иванов	27
Иванов	инженер	Петров	20
Петров	дворник	Иванов	27
Петров	дворник	Петров	20

R_1(ФИО, ДОЛЖНОСТЬ) =

Иванов	инженер
Петров	дворник

R_2(ФИО, ВОЗРАСТ) =

Иванов	27
Петров	20

Запишите схему и расширение отношения S, которое есть результат эквисоединения отношений R_1 и R_2 по первым атрибутам.

Операция соединения есть результат сочетания кортежей, имеющих общее значение для одного или нескольких общих атрибутов отношений.

ФИО	*ДОЛЖНОСТЬ*	*ФИО_1*	*ВОЗРАСТ*
Иванов	инженер	Иванов	27
Петров	дворник	Петров	20

Естественное соединение

Является разновидностью эквисоединения, из которого исключены дубликаты атрибутов, по которым оно проводилось.

Производится *по всем* одинаковым атрибутам.

В общем случае эквивалентно следующей последовательности реляционных операций:

1. Переименовать одинаковые атрибуты в отношениях
2. Выполнить декартово произведение отношений
3. Выполнить выборку по совпадающим значениям атрибутов, имеющих одинаковые имена
4. Выполнить проекцию, удалив атрибуты-дубликаты
5. Переименовать атрибуты в первоначальные имена

Естественное соединение. Пример

R_1(ФИО, ДОЛЖНОСТЬ) =

Иванов	инженер
Петров	дворник

R_2(ФИО, ВОЗРАСТ) =

Иванов	27
Петров	20

S =

ФИО	*ДОЛЖНОСТЬ*	*ВОЗРАСТ*
Иванов	инженер	27
Петров	дворник	20

Взятие разности

R_1(ФИО, ДОЛЖНОСТЬ) =

Иванов	инженер
Петров	дворник

R_2(ФИО, ВОЗРАСТ) =

Иванов	27
Петров	20

$$S = R_1 \setminus R_2$$

?

Выборка данных

Оператор выбора SELECT

```
SELECT [ ALL ! DISTINCT ] <список полей> ! *
      FROM  <список таблиц>
      [ WHERE        <условие выборки>
      [ GROUP BY  <список полей для группы>
      [ HAVING       <условие выборки для группы>
      [ ORDER BY   < список полей, по которым упорядочить
вывод>
```

Используются:

- ✔ операторы сравнения: =, < , > , <= , >= , <>;
- ✔ булевы операторы: AND, OR, NOT;
- ✔ оператор проверки на вхождение в множество: IN;
- ✔ оператор проверки на вхождение в диапазон: BETWEEN;
- ✔ оператор проверки на существование: EXISTS;
- ✔ оператор проверки удовлетворению шаблону (только для символьных полей) LIKE;
- ✔ операторы сравнения с NULL: IS NULL, IS NOT NULL;
- ✔ встроенные функции;
- ✔ константы и выражения;

Выборка без использования предложения WHERE

Выборка всей информации из таблицы

```
SELECT список_всех_полей_таблицы | * FROM имя_таблицы;
```

-- вывод всей информации из таблицы DEALERS

1. `SELECT D_id, Name, Procent, Comments FROM Dealers;`
2. `SELECT * FROM Dealers;`

Вертикальная фильтрация с указанием порядка вывода атрибутов

```
SELECT поле1[, поле2, …] FROM имя_таблицы;
SELECT Name FROM Dealers;
```

Исключение дубликатов

```
DISTINCT
SELECT DISTINCT Prod_id FROM Outgoing;
```

Выборка вычисляемых значений

```
SELECT Name "Имя", (Procent / 100) "Доля" FROM Dealers;
```

Сортировка результирующего набора данных

```
SELECT * FROM DEALERS ORDER BY Name ASC, Procent DESC;
```

Выборка с использованием предложения WHERE

Использование операторов сравнения

```
SELECT Name FROM Managers WHERE Percent <=50;
```

Использование BETWEEN

```
SELECT Name FROM MANAGERS WHERE Percent BETWEEN 20 AND 40;
SELECT Name FROM MANAGERS WHERE Percent NOT BETWEEN 20 AND 40;
```

Использование IN

```
SELECT Name FROM Managers WHERE Percent IN (5, 10, 15);
```

Использование LIKE

Соответствие текстовому шаблону: *имя_столбца* LIKE *текстовая_константа*
символ _ (подчеркивание) – любой одиночный символ;
символ % (процент) – любая последовательность символов.

```
SELECT * FROM Managers WHERE Name LIKE 'И%ов';
```

Сравнение с неопределенным значением NULL

-- отбор менеджеров, у которых не указан размер комиссионных

```
SELECT Name FROM Managers WHERE Percent IS NULL;
```

Встроенные функции ORACLE SQL (1)

Функция – это оператор ORACLE SQL, который может принимать один или несколько параметров и результат выполнения которого может быть подставлен в выражение. Функции могут быть использованы везде, где используются переменные, столбцы или выражения соответствующего типа.

Все функции делятся на две большие группы: однострочные и групповые. Однострочные функции выполняют операции, которые могут повлиять на каждую строку таблицы в отдельности.

Групповые функции предназначены для получения агрегированной информации о некоторых подмножествах данных.

Системные переменные

SYSDATE возвращает текущие дату и время сервера ORACLE.

USER возвращает идентификатор пользователя ORACLE
USERENV возвращает множество разных сведений о вычислительной среде

Встроенные функции ORACLE SQL (2)

Числовые функции

ROUND округляет числа с любой заданной точностью.
TRUNC усекает число, понижая его точность.

```
SELECT ROUND(1234.5678, 3) FROM DUAL;
SELECT TRUNC(1234.5678, 3) FROM DUAL;
```

Текстовые функции

UPPER, **LOWER и INITCAP**

Меняют регистр переданного им текста.

```
SELECT Name FROM Products WHERE LOWER(Name)='hewlett packard';
SELECT Name FROM Products WHERE UPPER(Name)='HEWLETT PACKARD';
```

LENGTH возвращает длину символьного поля.

SUBSTR возвращает подстроку

```
SUBSTR(исходный_текст, начальная_позиция, количество_символов)
```

INSTR поиск подстроки в строке, определение номера символа в исходной строке

```
INSTR(исходный_текст, подстрока, позиция_начального_символа)
```

Встроенные функции ORACLE SQL (3)

Конкатенация строк ||

-- соединение в один столбец содержимого столбцов разных типов

```
SELECT Name || ',' || Percent || '%' "Комиссионные" FROM
Managers;
```

LTRIM и RTRIM

Удаление избыточных пробелов в начале или конце текстовой строки.

Функции работы с датами

ADD_MONTHS возвращает дату с тем же днем месяца, что и в исходной дате, но отнесенную на заданное количество месяцев в будущее или прошлое.

-- определение наименований товаров, срок годности которых истечет менее, чем через два месяца

```
SELECT Name FROM Products WHERE Expire_Time <
ADD_MONTHS(TRUNC(SYSDATE), 2);
```

LAST_DAY возвращает последний день любого месяца

-- первый день месяца, следующего за месяцем приема на работу

```
SELECT LAST_DAY(Hire_Date)+1 FROM Managers;
```

Встроенные функции ORACLE SQL (4)

MONTHS_BETWEEN возвращает количество месяцев, разделяющих две даты.

-- определение количества месяцев, оставшихся до истечения срока годности товаров

```
SELECT Name, TRUNC(MONTHS_BETWEEN(Expire_Time, SYSDATE), 0)
FROM Products;
```

EXTRACT извлекает значение из даты или значения интервала.

```
EXTRACT (
{ YEAR | MONTH | DAY | HOUR | MINUTE | SECOND }
| { TIMEZONE_HOUR | TIMEZONE_MINUTE }
| { TIMEZONE_REGION | TIMEZONE_ABBR }
FROM { date_value | interval_value } )
```

```
SELECT EXTRACT (YEAR FROM DATE '2020-08-22') FROM DUAL;
```

Встроенные функции ORACLE SQL (5)

Функции преобразования данных

TO_CHAR преобразует дату, время или число в текст.

```
TO_CHAR(входное_значение, маска_форматирования)
```

-- отображение даты и времени приема на работу в формате «дд.мм.гггг чч:мм:сс»

```
SELECT Name, TO_CHAR(Hire_Date, 'dd.mm.yyyy hh24:mi:ss'
FROM Managers;
```

-- отображение цены товаров в денежном эквиваленте

```
SELECT TO_CHAR(Value, '$99,999.00') "Цена", DayFrom,
DayTo FROM Prices;
```

TO_DATE преобразует текстовое представление даты (и/или времени) в действительные значения даты/времени.

```
INSERT INTO Managers(Man_id, Name, Hire_Date)
VALUES(1, 'Петров П.М.',
TO_DATE('12.03.2005','dd.mm.yyyy'));
```

Встроенные функции ORACLE SQL (6)

Функции преобразования данных

TO_NUMBER преобразует строку в число

```
TO_NUMBER( string1, [ format_mask ], [ nls_language ] )
SELECT TO_NUMBER('1242.45', '9999.99')  FROM DUAL;
```

Прочие функции

NVL возвращает указанное значение вместо NULL.

```
NVL(входное_значение, результат_если_NULL)
```

-- для товаров, для которых не указано описание, выводить «нет описания»

```
SELECT Name, NVL(Description, 'нет описания') "
Описание" FROM Products;
```

Встроенные функции ORACLE SQL (7)

Групповые функции

COUNT возвращает количество записей в группе.

1. **COUNT(*)** – подсчет количества записей в группе;
2. **COUNT(поле)** – подсчет количества отличных от NULL значений в указанном поле записей группы;
3. **COUNT(DISTINCT поле)** – подсчет количества уникальных отличных от NULL значений в указанном поле записей группы.

-- подсчет количества строк в таблице MANAGERS

```
SELECT COUNT(*) FROM Managers;
```

-- подсчет количества менеджеров, у которых не указан размер комиссионных

```
SELECT COUNT(*)-COUNT(Percent) FROM Managers;
```

-- подсчет количества всех дат приема на работу без повторений

```
SELECT COUNT(DISTINCT TRUNC(Hire_Date)) FROM Managers;
```

SUM возвращает суммарное значение для группы.

-- подсчет количества товара, проданного менеджером с номером 1

```
SELECT SUM(Quantity) FROM Outgoing WHERE Man_id=1;
```

Встроенные функции ORACLE SQL (8)

Групповые функции

MAX возвращает максимальное значение для группы.

-- подсчет максимального размера комиссионных среди всех менеджеров

```
SELECT MAX(Procent) FROM Managers;
```

MIN возвращает минимальное значение для группы.

-- вычисление даты первой продажи товара менеджера с номером 1

```
SELECT MIN(Out_Date) FROM Outgoing WHERE Man_id=1;
```

AVG возвращает среднее значение для группы.

-- подсчет средней цены товара с номером 1

```
SELECT AVG(Value) FROM Prices WHERE Prod_id=1;
```

Функции AVG и SUM применимы только к числовым полям.

Запросы с использованием соединений (1)

Декартово произведение таблиц

Соединения – это подмножества декартова произведения. Декартово произведение N таблиц – это таблица, содержащая все возможные строки R, такие, что R является сцеплением какой-либо строки из первой таблицы, строки из второй таблицы, ... и строки из N-й таблицы.

Для получения декартова произведения нескольких таблиц с помощью SELECT надо указать в параметре FROM перечень перемножаемых таблиц, а во фразе SELECT – все их столбцы.

```
SELECT Managers.*, Contracts.* FROM Managers, Contracts;
```

Запросы с использованием соединений (2)

Эквисоединение таблиц

Актуальные строки можно отобрать из декартово произведения путем ввода в запрос параметра WHERE, в котором устанавливается соответствие между полями, посредством которых каждая пара таблиц связана между собой.

```
SELECT Managers.*, Contracts.* FROM Managers,
Contracts
WHERE Managers.Man_id = Contracts.Man_id;
```

Запросы с использованием соединений (3)

Эквисоединение таблиц

Актуальные строки можно отобрать из декартово произведения путем ввода в запрос параметра WHERE, в котором устанавливается соответствие между полями, посредством которых каждая пара таблиц связана между собой.

```
SELECT Managers.*, Contracts.* FROM Managers,
Contracts WHERE Managers.Man_id = Contracts.Man_id;
```

Естественное соединение таблиц

Эквисоединение с исключенными дубликатами столбцов, по которым оно проводилось.

```
SELECT Managers.Man_id, D_id, Name, Hire_Date, Percent,
Comments, Parent_id, DayFrom, DayTo
FROM Managers, Contracts
WHERE Managers.Man_id = Contracts.Man_id;
```

Запросы с использованием соединений (4)

Композиция таблиц

Эквисоединение, из которого полностью исключены столбцы, по которым оно производилось.

```
SELECT D_id, Name, Hire_Date, Percent, Comments,
Parent_id, DayFrom, DayTo FROM Managers, Contracts
WHERE Managers.Man_id = Contracts.Man_id;
```

Соединение таблиц с дополнительным условием

-- получение информации о менеджерах и заключенных ими контрактах за последнюю неделю

```
SELECT Name, DayFrom, DayTo
FROM Managers, Contracts
WHERE Managers.Man_id = Contracts.Man_id
AND DayFrom BETWEEN TRUNC(SYSDATE)-7 AND SYSDATE;
```

Запросы с использованием соединений (5)

Соединение таблицы со своей копией

В ряде приложений возникает необходимость одновременной обработки данных нескольких копий таблицы, создаваемых на время выполнения запроса.

Временную копию таблицы можно сформировать, указав имя псевдонима за именем таблицы во фразе FROM. Примеры соединения таблиц со своей копией:

-- получение имен менеджеров и имен их руководителей

```
SELECT M1.Name, M2.Name FROM Managers M1, Managers M2
WHERE M1.Parent_id=M2.Man_id;
```

-- получение списка однофамильцев

```
SELECT M1.* FROM Managers M1, Managers M2
WHERE M1.Name=M2.Name AND M1.Man_id<>M2.Man_id;
```

Запросы с использованием соединений (6)

Внутреннее и внешнее соединение таблиц

Во многих СУБД существуют реализации операции внутреннего и внешнего условных соединений таблиц внутри одного запроса – INNER JOIN (внутреннее соединение), LEFT JOIN (полное левое соединение) и RIGHT JOIN (полное правое соединение).

```
SELECT список_полей
FROM таблица1 ( INNER | LEFT | RIGHT ) JOIN таблица2
ON таблица1.связующее_поле = таблица2.связующее_поле;
```

В результате выполнения внутреннего соъединения из кортежей двух объединяемых таблиц остаются только те, для которых выполняется указанное условие.

```
-- получение списка имен менеджеров и связанных с ними дилеров
-- (менеджеры, не связанные с дилерами и дилеры, не связанные
-- с менеджерами исключатся из результата)
SELECT Managers.Name, Dealers.Name FROM Managers
INNER JOIN Dealers ON Managers.D_id=Dealers.D_id;
```

Запросы с использованием соединений (7)

При полном (внешнем) левом соединении из кортежей двух объединяемых таблиц остаются все кортежи таблицы, указанной слева от условного выражения, и кортежи правой таблицы, для которых выполняется указанное условие.

-- получение всех менеджеров и связанных с ними дилеров, включая менеджеров, не связанных с дилерами

```
SELECT Managers.Name, Dealers.Name FROM Managers
LEFT JOIN Dealers ON Managers.D_id=Dealers.D_id;
```

При полном (внешнем) правом соединении из кортежей двух объединяемых таблиц остаются все кортежи таблицы, указанной справа от условного выражения, и кортежи левой таблицы, для которых выполняется указанное условие.

-- получение всех дилеров и связанных с ними менеджеров, включая дилеров, не связанных с менеджерами

```
SELECT Managers.Name, Dealers.Name FROM Managers
RIGHT JOIN Dealers ON Managers.D_id=Dealers.D_id;
```