

Базы данных



Дьячкова
Ирина Сергеевна

Кафедра ПМиК:
430а (главный корпус),
406 (новый корпус)

Формы освоения материала

Лекции

Лабораторные работы

Расчетно-графическая работа

Формы контроля знаний

Контроль посещения лекций и практик

Защита лабораторных работ (оценка)

Защита РГР (оценка)

Подсчет баллов

Экзамен (оценка)

Рейтинг (баллы)

Лекции: посещение (+5), ответы (+), пропуск (-5)

Лабораторные работы: оценки 3, 4, 5
пропуск (-5)

Автомат: $\geq 80\%$ от max, все лабораторные ≥ 4 ,
контрольные сроки ≥ 1

Собеседование: от 60% до 80% от max,
все лабораторные ≥ 3 , контрольные сроки ≥ 1

Экзамен по билетам: $< 60\%$ от max с предварительной
отработкой лабораторных работ

База данных (БД) - набор специальным образом организованных данных, хранимых в памяти вычислительной системы и отображающих состояние объектов и их взаимосвязей в рассматриваемой предметной области.

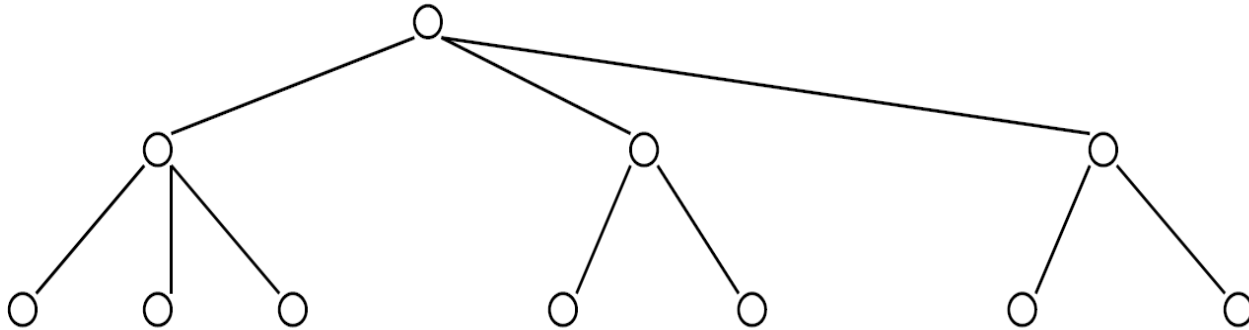
Модель представления данных - логическая структура данных, хранимых в базе.

Основные модели представления данных:

- 1) иерархическая,
- 2) сетевая,
- 3) реляционная,
- 4) постреляционная,
- 5) многомерная,
- 6) объектно-ориентированная.

Иерархическая модель

В иерархической модели связи между данными можно описать с помощью упорядоченного графа (или дерева).



Отдел

Отд_Номер	Отд_Размер	Отд_Зарп.
-----------	------------	-----------

Начальник

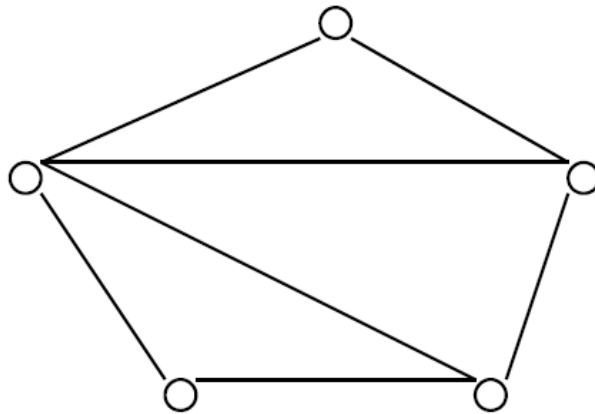
Нач_Номер	Нач_Имя	Нач_Телеф.
-----------	---------	------------

Сотрудники

Сотр_Номер	Сотр_Имя	Сотр_Зарп.
------------	----------	------------

Сетевая модель

Сетевая модель данных позволяет отображать разнообразные взаимосвязи элементов данных в виде произвольного графа, обобщая тем самым иерархическую модель данных.



Сетевая БД состоит из набора записей и набора соответствующих связей. На формирование связи особых ограничений не накладывается. Если в иерархических структурах запись-потомок могла иметь только одну запись-предка, то в сетевой модели данных запись-потомок может иметь произвольное число записей-предков (сводных родителей).

Реляционная модель

Реляционная модель основывается на понятии отношение (relation). Отношение представляет собой множество элементов, называемых кортежами. Наглядной формой представления отношения является двумерная таблица. Таблица имеет строки (записи) и столбцы (колонки). Каждая строка таблицы имеет одинаковую структуру и состоит из полей. Строкам таблицы соответствуют кортежи, а столбцам — атрибуты отношения.

ОТДЕЛЫ

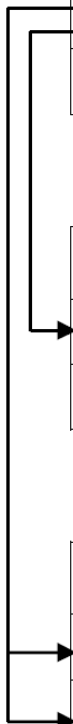
Отд_Номер	Отд_Размер	Отд_Зарп
3	10	50 000
4	15	75 000

НАЧАЛЬНИКИ

Отд_Номер	Нач_Номер	Нач_Имя	Нач_Телеф
3	31	Иванов	0-11
4	41	Васильев	0-12

СОТРУДНИКИ

Отд_Номер	Сотр_Номер	Сотр_Имя	Сотр_Зарп
3	32	Петров	4 000
3	33	Сидоров	4 000



Постреляционная модель

Постреляционная модель данных - расширенная реляционная модель, снимающая ограничение неделимости данных, хранящихся в записях таблиц.

НАКЛАДНЫЕ_ТОВАРЫ

ИНВ_N	НАЗВАНИЕ	КОЛИЧ
73	Ручка	3
73	Линейка	2
74	Тетрадь	1
74	Карандаш	6
74	Блокнот	2
88	Папка	1

а) НАКЛАДНЫЕ

ИНВ_N	ПОКУП_N
73	23
74	45
88	23

б) НАКЛАДНЫЕ

ИНВ_N	ПОКУП_N	НАЗВАНИЕ	КОЛИЧ
73	23	Ручка	3
		Линейка	2
74	45	Тетрадь	1
		Карандаш	6
		Блокнот	2
88	23	Папка	1

Многомерная модель

Многомерные системы управления базами данных (СУБД) являются узкоспециализированными СУБД, предназначенными для интерактивной аналитической обработки информации.

По сравнению с реляционной моделью многомерная организация данных обладает более высокой наглядностью и информативностью.

Реляционная модель:

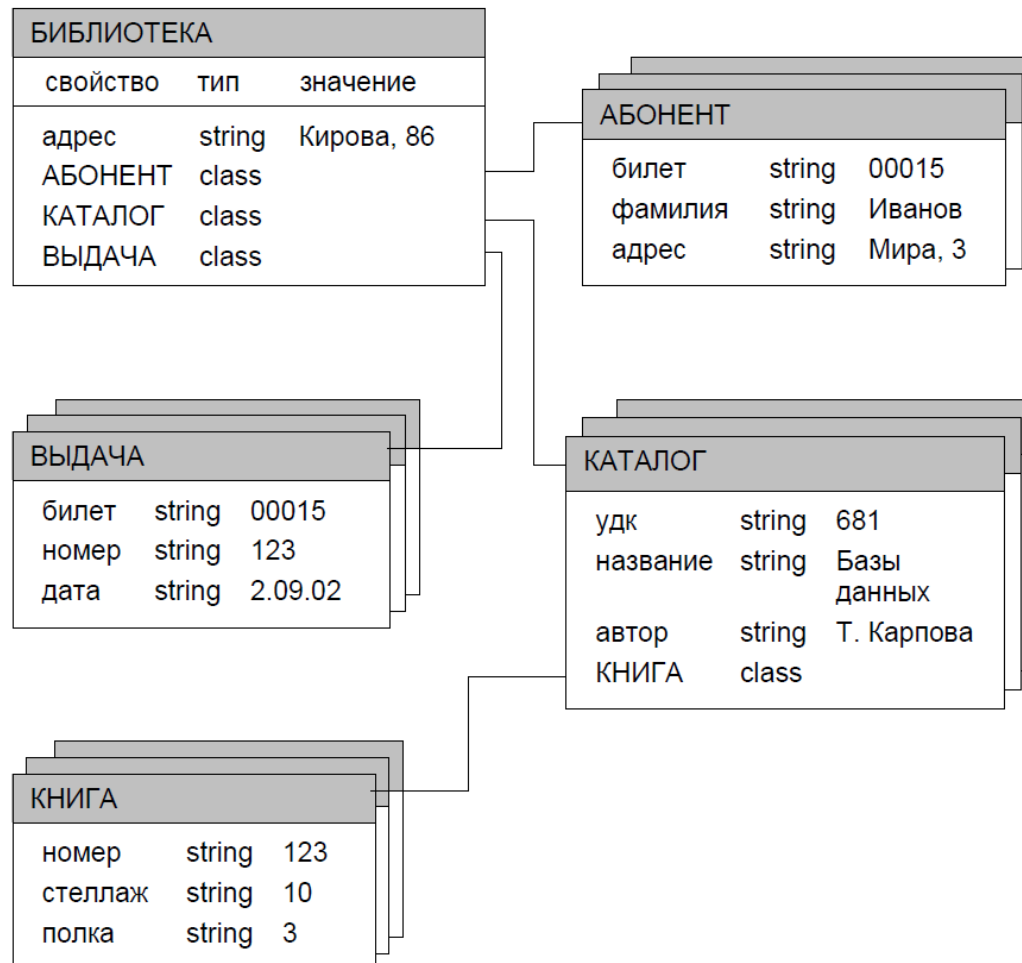
МОДЕЛЬ	МЕСЯЦ	ОБЪЕМ
«Жигули»	июнь	12
«Жигули»	июль	24
«Жигули»	август	5
«Москвич»	июнь	2
«Москвич»	июль	18
«Волга»	июль	19

Многомерная модель:

МОДЕЛЬ	Июнь	Июль	Август
«Жигули»	12	24	5
«Москвич»	2	18	нет
«Волга»	нет	19	нет

Объектно-ориентированная модель

В объектно-ориентированной модели при представлении данных имеется возможность идентифицировать отдельные записи базы. Между записями БД и функциями их обработки устанавливаются взаимосвязи с помощью механизмов, подобных соответствующим средствам в объектно-ориентированных языках программирования.



Система управления базами данных (СУБД) — это комплекс языковых и программных средств, предназначенный для создания, ведения и совместного использования БД многими пользователями.

Банк данных (БнД) в общем случае состоит из следующих компонентов: базы (нескольких баз) данных, системы управления базами данных, словаря данных, администратора, вычислительной системы и обслуживающего персонала. Банк данных является разновидностью ИС, в которой реализованы функции централизованного хранения и накопления обрабатываемой информации, организованной в одну или несколько баз данных.

Приложение — программа или комплекс программ, обеспечивающих автоматизацию обработки информации для прикладной задачи. Нами рассматриваются приложения, использующие БД. Приложения могут создаваться в среде или вне среды СУБД — с помощью системы программирования, использующей средства доступа к БД. Приложения, разработанные в среде СУБД часто называют приложениями СУБД, а приложения, разработанные вне СУБД, — внешними приложениями.

Словарь данных (СД) — подсистема БНД, предназначенная для централизованного хранения информации о структурах данных, взаимосвязях файлов БД друг с другом, типах данных и форматах их представления, принадлежности данных пользователям, кодах защиты и разграничения доступа и т. п.

Администратор базы данных (АБД) —лицо или группа лиц, отвечающих за выработку требований к БД, ее проектирование, создание, эффективное использование и сопровождение. В процессе эксплуатации АБД обычно следит за функционированием информационной системы, обеспечивает защиту от несанкционированного доступа, контролирует избыточность, непротиворечивость, сохранность и достоверность хранимой в БД информации.

Для однопользовательских информационных систем функции АБД обычно возлагаются на лиц, непосредственно работающих с приложением БД.

Язык структурных запросов SQL

Большинство современных СУБД построено на реляционной модели данных. Для получения информации из таблиц базы данных в качестве языка манипулирования данными в теоретическом плане используются три абстрактных языка:

- язык реляционной алгебры;
- язык реляционного исчисления на кортежах;
- язык реляционного исчисления на доменах.

В качестве практического языка работы с данными в середине 70-х годов фирмой IBM разработан язык структурных запросов SQL, ставший впоследствии стандартом de-facto при работе с базами данных.

Язык SQL содержит 4 группы операторов:

- Операторы описания данных: CREATE, DROP, ALTER и др.
- Операторы манипуляции данными: INSERT, DELETE, SELECT, UPDATE и др.
- Операторы задания прав доступа в базе данных: GRANT / REVOKE , LOCK / UNLOCK , SET LOCK MODE
- Операторы защиты, восстановления данных и прочие операторы.

Создадим базу данных, таблицы которой описаны следующим образом:

Таблица поставщиков S:

```
Create table S (n_post char(5) not NULL,  
name char(20),  
rating smallint,  
town char(15))
```

Таблица поставок SP:

```
Create table SP (n_post char(5) ,  
n_det char(6),  
date_post date,  
kol smallint)
```

Таблица деталей P:

```
Create table P (n_det char(6) ,  
name char(20),  
cvet char(7),  
ves smallint,  
town char(15))
```



```
INSERT INTO s (n_post, name, rating, town) VALUES ('S2', 'Jones', 10, 'Paris');
INSERT INTO s (n_post, name, rating, town) VALUES ('S3', 'Blake', 30, 'Paris');
INSERT INTO s (n_post, name, rating, town) VALUES ('S4', 'Clark', 20, 'London');
INSERT INTO s (n_post, name, rating, town) VALUES ('S5', 'Adams', 30, 'Athens');
```

```
INSERT INTO sp (n_post, n_det, date_post, kol) VALUES ('S1', 'P1', '2021-02-01', 300);
INSERT INTO sp (n_post, n_det, date_post, kol) VALUES ('S1', 'P2', '2021-04-05', 200);
INSERT INTO sp (n_post, n_det, date_post, kol) VALUES ('S1', 'P3', '2021-05-12', 400);
INSERT INTO sp (n_post, n_det, date_post, kol) VALUES ('S1', 'P4', '2021-06-15', 200);
INSERT INTO sp (n_post, n_det, date_post, kol) VALUES ('S1', 'P5', '2021-07-22', 100);
INSERT INTO sp (n_post, n_det, date_post, kol) VALUES ('S1', 'P6', '2021-08-13', 100);
INSERT INTO sp (n_post, n_det, date_post, kol) VALUES ('S2', 'P1', '2021-03-03', 300);
INSERT INTO sp (n_post, n_det, date_post, kol) VALUES ('S2', 'P2', '2021-06-12', 400);
INSERT INTO sp (n_post, n_det, date_post, kol) VALUES ('S3', 'P2', '2021-04-04', 200);
INSERT INTO sp (n_post, n_det, date_post, kol) VALUES ('S4', 'P2', '2021-03-23', 200);
INSERT INTO sp (n_post, n_det, date_post, kol) VALUES ('S4', 'P4', '2021-06-17', 300);
INSERT INTO sp (n_post, n_det, date_post, kol) VALUES ('S4', 'P5', '2021-08-22', 400);
```

Таблица поставщиков (s):

n_post	name	rating	town
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Таблица поставок (sp):

n_post	n_det	date_post	kol
S1	P1	2021-02-01	300
S1	P2	2021-04-05	200
S1	P3	2021-05-12	400
S1	P4	2021-06-15	200
S1	P5	2021-07-22	100
S1	P6	2021-08-13	100
S2	P1	2021-03-03	300
S2	P2	2021-06-12	400
S3	P2	2021-04-04	200
S4	P2	2021-03-23	200
S4	P4	2021-06-17	300
S4	P5	2021-08-22	400

Таблица товаров (p):

n_det	name	cvet	ves	town
P1	Monitor screen	Red	12000	London
P2	Computer mouse	Green	150	Paris
P3	Keyboard	Blue	400	Rome
P4	Keyboard	Red	350	London
P5	Power unit	Blue	1200	Paris
P6	Computer case	Red	7500	London

Простые запросы на языке SQL

Запрос на языке SQL формируется с использованием оператора Select. Оператор Select используется

- для выборки данных из базы данных;
- для получения новых строк в составе оператора Insert;
- для обновления информации в составе оператора Update.

В общем случае оператор Select содержит следующие восемь спецификаторов, расположенных в операторе в следующем порядке:

- спецификатор Select;
- спецификатор From;
- спецификаторы Join;
- спецификатор Where;
- спецификатор Having;
- спецификатор Order by;
- спецификатор Into temp.

1. Выбор всех строк и столбцов таблицы

Пример.

Выдать полную информацию о поставщиках.

Select * from S

Символ * после Select означает, что в результат должны быть включены все столбцы таблицы S.

Результат: таблица S в полном объеме

n_post	name	rating	town
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

2. Изменение порядка следования столбцов

Пример.

Выдать таблицу S в следующем порядке: фамилия, город, рейтинг, номер_поставщика.

Select name, town, rating, n_post from S

Результат: таблица S в требуемом порядке.

name	town	rating	n_post
Smith	London	20	S1
Jones	Paris	10	S2
Blake	Paris	30	S3
Clark	London	20	S4
Adams	Athens	30	S5

3. Выбор заданных столбцов

Пример.

Выдать номера всех поставляемых деталей.

```
Select n_det from SP
```

Результат: столбец n_det таблицы SP

4. Выбор без повторения

Пример.

Выдать номера всех поставляемых деталей,
исключая дублирование.

```
Select distinct n_det from SP
```

n_det
P1
P2
P3
P4
P5
P6

5. Ограничение в выборке

Пример.

Выдать номера всех поставщиков, находящихся в Париже и имеющих рейтинг > 20 .

Select n_post from S where town = 'Paris' and rating > 20

Результат:

n_post
S3

6. Выборка с упорядочиванием

Пример.

Выдать номера поставщиков, находящихся в Париже в порядке убывания рейтинга.

Select n_post, rating from S where town = 'Paris' order by rating desc

Результат:

n_post	rating
S3	30
S2	10

7. Использование в запросах констант и выражений

Пример.

Выдать вес каждой детали в граммах.

```
Select n_det || ' вес в килограммах=' || ves/1000 from P
```

|| - оператор конкатенации строк

Или

```
Select concat(n_det, " вес в килограммах =", ves/1000) as text1 from P
```

text1

P1 вес в килограммах = 12.0000

P2 вес в килограммах = 0.1500

P3 вес в килограммах = 0.4000

P4 вес в килограммах = 0.3500

P5 вес в килограммах = 1.2000

P6 вес в килограммах = 7.5000

8. Упорядочивание по нескольким столбцам

Пример.

Выдать список поставщиков, упорядоченных по городу, в пределах города - по рейтингу.

Select * from S order by 4, 3

n_post	name	rating	town
S5	Adams	30	Athens
S1	Smith	20	London
S4	Clark	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris

9. Фраза between

Пример.

Выдать информацию о деталях,
вес которых лежит в
диапазоне от 150 до 450 г.

Select n_det, name, ves from P
where ves between 150 and 450

n_det	name	ves
P2	Computer mouse	150
P3	Keyboard	400
P4	Keyboard	350

10. Фраза in (not in)

Пример.

Выдать детали, вес которых
равен 150, 400 или 1200 г.

Select n_det, name, ves from P
where ves in (150, 400, 1200)

n_det	name	ves
P2	Computer mouse	150
P3	Keyboard	400
P5	Power unit	1200

11. Выбор по шаблону

Для запросов с поиском по шаблону, основанных на поиске подстрок в полях типа CHARACTER, по стандарту ANSI используется ключевое слово LIKE.

Включение в выражение ключевого слова NOT порождает условие с обратным смыслом.

СИМВОЛ	ЗНАЧЕНИЕ
LIKE	
%	заменяет последовательность символов
_ (подчерк)	заменяет любой одиночный символ
\	отменяет специальное назначение следующего за ним символа

Примеры.

а) Выбрать список деталей, начинающихся с буквы “С”

Select n_det, name, ves from P where name like 'C%'

Результат:

n_det	name	ves
P2	Computer mouse	150
P6	Computer case	7500

б) Выдать список фамилий поставщиков, третья буква имени которых "a".

Select name from S where name like '__a%'

Результат:

name
Blake
Clark
Adams

Числовой тип данных

Синтаксис типа данных	Описание
INT	Целое число нормального размера, которое может быть подписано или без знака. Если подписанный, допустимый диапазон от -2147483648 до 2147483647. Если без знака допустимый диапазон составляет от 0 до 4294967295. Вы можете указать ширину до 11 цифр.
TINYINT	Очень маленькое целое число, которое может быть подписано или без знака. Если подписанный, допустимый диапазон от -128 до 127. Если unsigned, допустимый диапазон от 0 до 255. Вы можете указать ширину до 4 цифр.
SMALLINT	Маленькое целое число, которое может быть подписано или без знака. Если подписанный, допустимый диапазон от -32768 до 32767. Если unsigned, допустимый диапазон от 0 до 65535. Вы можете указать ширину до 5 цифр.
MEDIUMINT	Среднее целое число, которое может быть подписано или без знака. Если подписанный, допустимый диапазон от -8388608 до 8388607. Если без знака, допустимый диапазон от 0 до 16777215. Вы можете указать ширину до 9 цифр.
BIGINT	Большое целое число, которое может быть подписано или без знака. Если подписанный, допустимый диапазон от -9223372036854775808 до 9223372036854775807. Если без знака допустимый диапазон составляет от 0 до 18446744073709551615. Вы можете указать ширину до 20 цифр.
FLOAT(m,d)	Число с плавающей запятой, которое не может быть беззнаковым. Вы можете определить длину отображения (m) и количество десятичных знаков (d). Это не требуется и по умолчанию будет 10,2, где 2 - число десятичных знаков, а 10 - общее количество цифр (включая десятичные числа). Десятичная точность может достигать 24 мест для поплавка.
DOUBLE(m,d)	Число с плавающей запятой с двойной точностью, которое не может быть беззнаковым. Вы можете определить длину отображения (m) и количество десятичных знаков (d). Это не требуется, и по умолчанию будет 16,4, где 4 - число десятичных знаков. Десятичная точность может достигать 53 мест для двойного. Реальный - синоним двойного.
DECIMAL(m,d)	Число распакованных чисел с плавающей запятой, которое не может быть беззнаковым. В распакованных десятичных знаках каждое десятичное число соответствует одному байту. Определение длины дисплея (m) и количества десятичных знаков (d) требуется. Числовой является синонимом десятичного числа.

Строковые типы данных

Синтаксис типа данных	Максимальный размер	объяснение
CHAR(size)	Максимальный размер 255 символов.	Где размер - количество сохраняемых символов. Строки с фиксированной длиной. Пространство дополняется справа на символы одинакового размера.
VARCHAR (размер)	Максимальный размер 255 символов.	Где размер - количество сохраняемых символов. Строка переменной длины.
TINYTEXT (размер)	Максимальный размер 255 символов.	Где размер - количество сохраняемых символов.
TEXT (размер)	Максимальный размер 65 535 символов.	Где размер - количество сохраняемых символов.
MEDIUMTEXT (размер)	Максимальный размер 16 777 215 символов.	Где размер - количество сохраняемых символов.
LONGTEXT (размер)	Максимальный размер 4 ГБ или 4 294 967 295 символов.	Где размер - количество сохраняемых символов.
BINARY (размер)	Максимальный размер 255 символов.	Где размер - количество двоичных символов для хранения. Строки с фиксированной длиной. Пространство дополняется справа на символы одинакового размера. (введено в MySQL 4.1.2)
VARBINARY (размер)	Максимальный размер 255 символов.	Где размер - количество сохраняемых символов. Строка переменной длины. (введено в MySQL 4.1.2)

Дата и время

Синтаксис типа данных	Максимальный размер	объяснение
DATE	Значения варьируются от «1000-01-01» до «9999-12-31».	Отображается как «yyyy-mm-dd».
DATETIME	Значения варьируются от «1000-01-01 00:00:00» до «9999-12-31 23:59:59».	Отображается как «yyyy-mm-dd hh: mm: ss».
TIMESTAMP(m)	Значения варьируются от '1970-01-01 00:00:01' UTC до '2038-01-19 03:14:07' TC.	Отображается как «ГГГГ-ММ-ДД ЧЧ: ММ: СС».
TIME	Значения варьируются от '-838: 59: 59' до '838: 59: 59'.	Отображается как «ЧЧ: ММ: СС».
YEAR[(2 4)]	Значение года как 2 цифры или 4 цифры.	Значение по умолчанию - 4 цифры.

Типы данных больших объектов

Синтаксис типа данных	Максимальный размер
TINYBLOB	Максимальный размер 255 байт.
BLOB	Максимальный размер 65 535 байт.
MEDIUMBLOB	Максимальный размер 16 777 215 байт.
LONGTEXT	Максимальный размер 4gb или 4 294 967 295 символов.

Использование функций. Агрегатные функции

Среди наиболее часто используемых функций отметим:

Sum - сумма значений по столбцу;

Avg - среднее значение в столбце;

Max - максимальное значение в столбце;

Min - минимальное значение в столбце.

Примеры.

а) Выдать общее количество поставщиков.

```
Select count (*) from S
```

Результат: 5

б) Выдать общее количество поставщиков, поставляющих в настоящее время детали.

```
Select count (distinct номер_поставщика ) from SP
```

Результат: 4

в) Выдать количество поставок для детали 'P2'.

Select count (*) from SP where номер_детали='P2'

Результат: 4

г) Выдать общее количество поставляемых деталей 'P2'.

Select sum (количество) from SP where номер_детали='P2'

Результат: 1000

д) Выдать средний, минимальный и максимальный объем поставок для поставщика S1 с соответствующим заголовком.

Select avg(количество) as average, min(количество) as minimum, max(количество) as maximum from SP where номер_поставщика='S1'

Результат:

average	minimum	maximum
216.6667	100	400

Строковые функции

Ниже перечислено несколько функций, относящихся к указанной группе. Общий их перечень достаточно широк.

`Substr(s,n,[l])` - функция возвращает подстроку `s`, начинающуюся с `n` длиной `l`;

`Lower(s)` - функция возвращает строку `s`, преобразованную к нижнему регистру;

`Length(s)` - функция возвращает длину строки `s`.

Пример.

Выдать два первых символа имен поставщиков, преобразованных к нижнему регистру.

`Select Substr(lower(name), 1, 2) from S`

Результат: Первые две буквы фамилии

`Substr(lower(name), 1, 2)`

sm

jo

bl

cl

ad