

# Язык запросов SQL

Технологии баз данных. Лекция 5

# Язык структурных запросов SQL



- SQL (Structured Query Language, язык структурных запросов) – стандартный язык запросов к реляционным базам данных.
- SQL основан на реляционной алгебре.
- В базовом варианте SQL является информационно-логическим языком, а не языком программирования. Спецификация SQL/PSM (Persistent Stored Modules, хранимые процедуры) представляет собой процедурное расширение SQL.
- Прототип языка SQL был разработан в конце 70-х годов в компании IBM Research. Он был реализован в первом прототипе реляционной СУБД фирмы IBM System R. В дальнейшем этот язык применялся во многих коммерческих СУБД и в силу своего широкого распространения постепенно стал стандартом "де-факто" для языков манипулирования данными в реляционных СУБД.

# SQL: историческая справка

Год	Название	Отличительные особенности
1970	SEQUEL	Язык управления данными экспериментальной СУБД IBM System R (SEQUEL – Structured English QUery Language).
1986	SQL-86 или SQL-87	Первый вариант стандарта, принятый институтом ANSI и одобренный ISO в 1987.
1989	SQL-89	Доработанный вариант предыдущего стандарта. Добавлена концепция первичного и внешнего ключей.
1992	SQL-92	Значительные изменения предыдущего стандарта.
1999	SQL:1999 или SQL-3	Добавлена поддержка регулярных выражений, рекурсивных запросов, поддержка триггеров, базовые процедурные расширения, нескаллярные типы данных и некоторые объектно-ориентированные возможности.
2003	SQL:2003	Добавлена поддержка работы с XML-данными, функции для работы с OLAP-базами данных, генераторы последовательностей и основанные на них типы данных.
2006	SQL:2006	Значительное расширение поддержки работы с XML-данными. Возможность совместно использовать в запросах SQL и XQuery.
2008	SQL:2008	Улучшение возможностей OLAP-функций, устранение некоторых неоднозначностей стандарта SQL:2003.
2011	SQL:2011	Поддержка хронологических баз данных.



# SQL: плюсы и минусы

- Преимущества
  - Независимость от конкретной СУБД
  - Стандарты
  - Декларативность
- Недостатки
  - Несоответствие классической РМД
    - Таблица  $\neq$  отношение, строка  $\neq$  кортеж и др.
  - Сложность
    - Избыточность языковых средств
    - Большой объем стандарта
  - Отступления от стандартов
    - "Надмножества подмножества SQL": Oracle PL/SQL, MS TransactSQL, PL/pgSQL и др.

# Структура SQL

- Язык определения данных (Data Definition Language, DDL)
- Язык манипулирования данными (Data Manipulation Language, DML)  $\supset$  язык запросов (Data Query Language, DQL)
- Язык определения доступа к данным (Data Control Language, DCL)
- Язык управления транзакциями (Transaction Control Language, TCL)
- Программный SQL

# Язык определения данных DDL

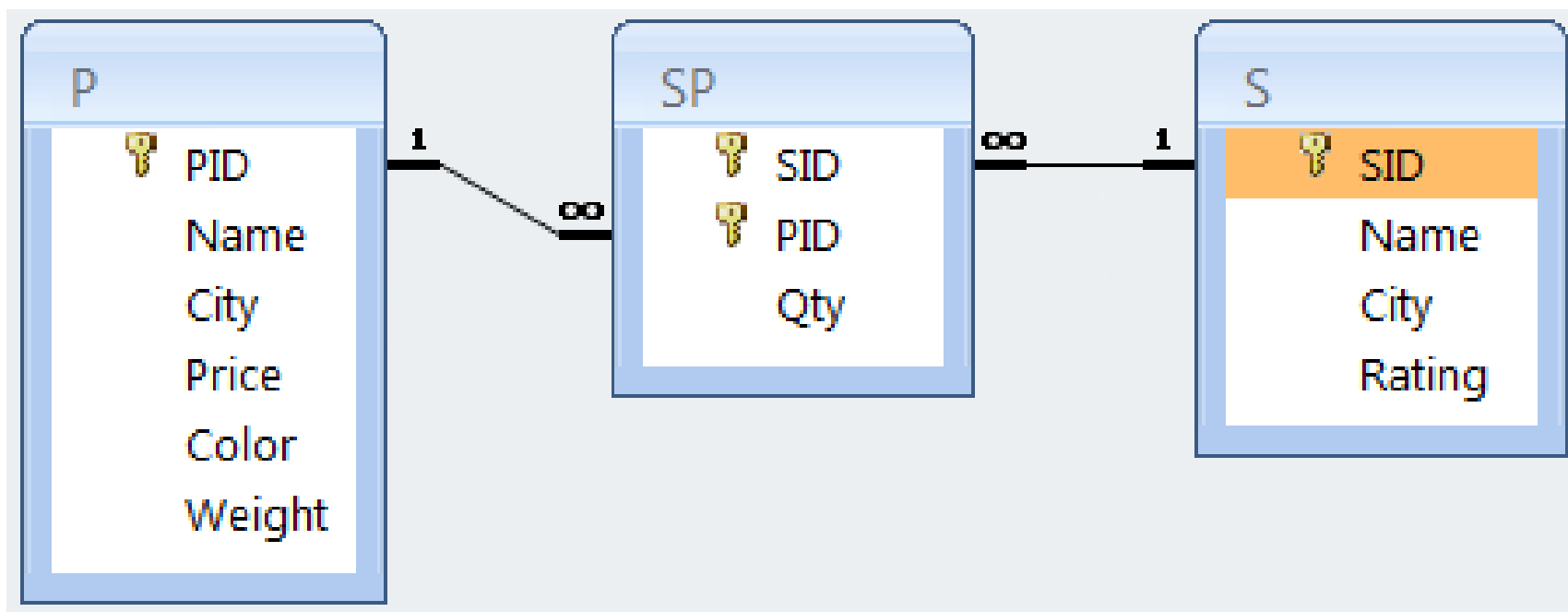
- DDL (Data Definition Language) - Создание, удаление, изменение определения объектов БД
  - CREATE
  - DROP
  - ALTER

# Модельная база данных

Детали

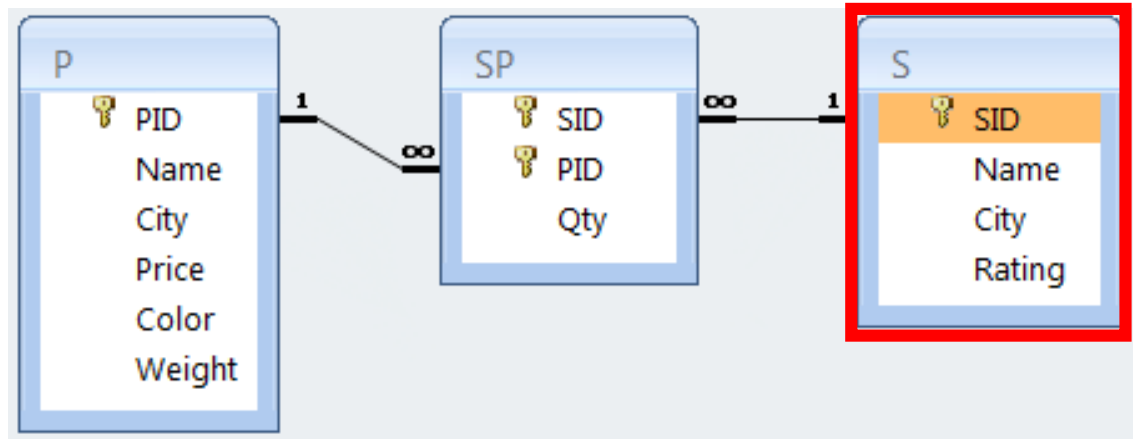
Поставки

Поставщики



# Операции с таблицами: создание

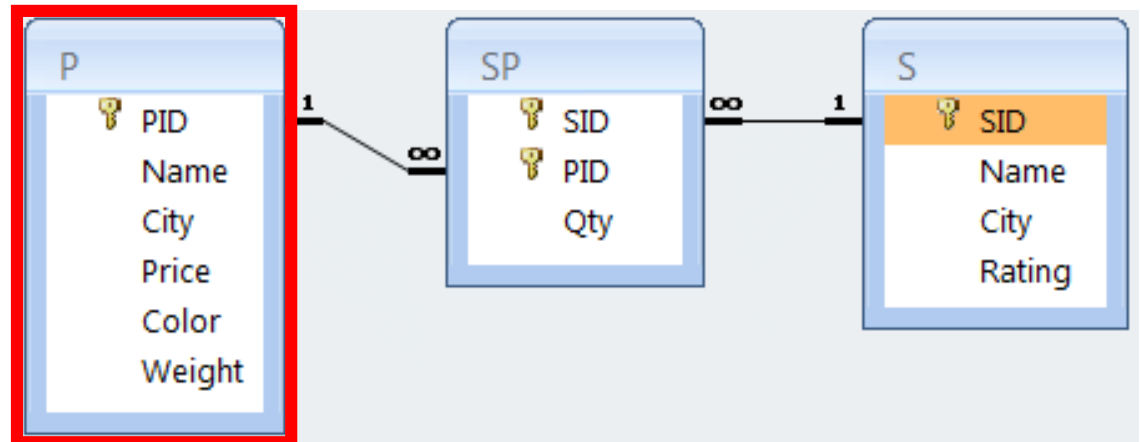
```
create table S (  
  SID char(4) primary key,  
  Name char(10) not null,  
  City char(10) not null,  
  Rating int not null  
);
```





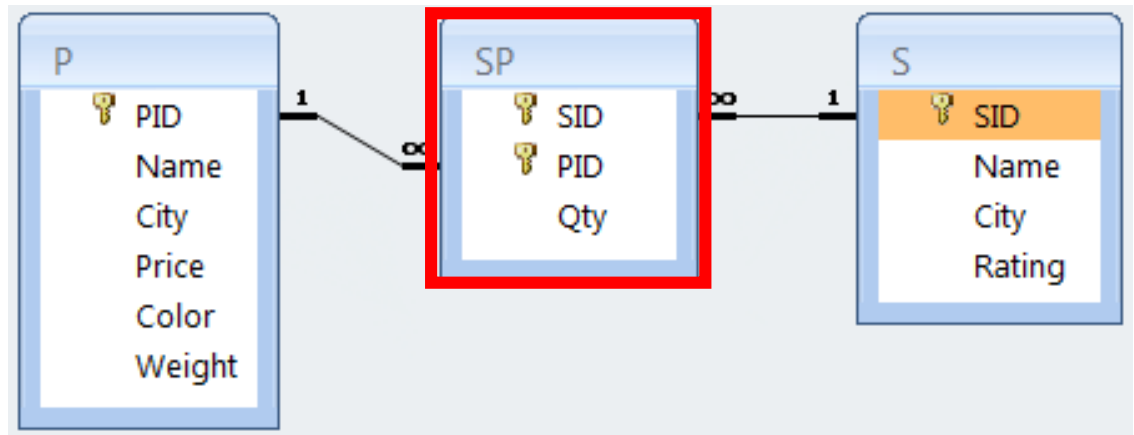
# Операции с таблицами: создание

```
create table P (  
  PID char(4) primary key,  
  Name char(10) not null,  
  City char(10) not null,  
  Price int not null,  
  Color char(10) not null,  
  Weight float not null  
);
```



# Операции с таблицами: создание

```
create table SP (  
  SID char(4),  
  PID char(4),  
  Qty int not null,  
  primary key (SID, PID),  
  foreign key (SID)  
    references S (SID)  
    on delete cascade*  
    on update cascade,  
  foreign key (PID)  
    references P (PID)  
    on delete cascade  
    on update cascade  
);
```

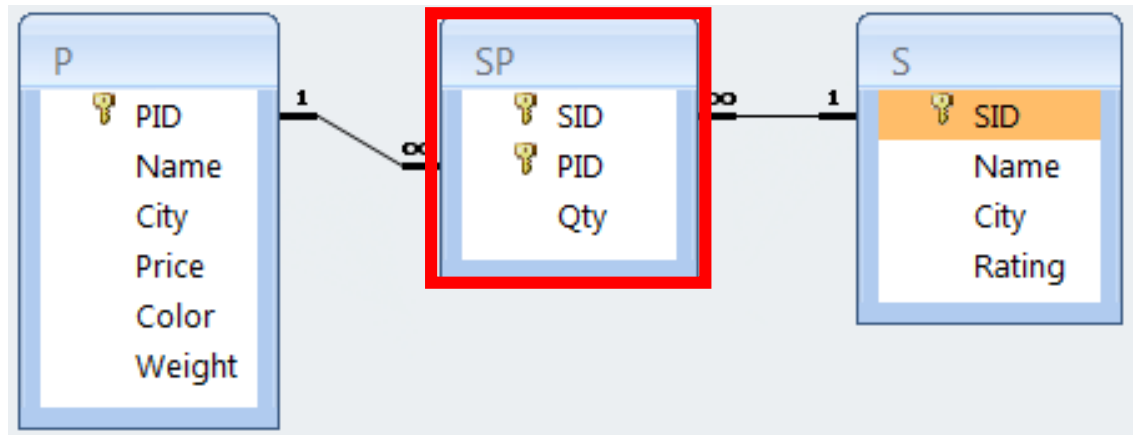


\*Допустимы режимы **cascade**, **set null**, **set default** и **no action**.

# Операции с таблицами: изменение

```
alter table SP  
add constraint qtychk  
check (Qty>0);
```

```
alter table P  
add column Discount  
float default 0;
```



# Операции с таблицами: удаление

- **drop table SP;**
- **drop table P;**
- **drop table S;**

# Операции с доменами

- В SQL домены не ограничивают сравнения, а являются лишь синонимами встроенных типов данных (char, int, date, time, timestamp, bit и др.)
- Создание домена
  - **create domain** Color char(10);
  - **create domain** Gender char  
**default** '?'  
**constraint** chk\_gender (check (value in 'M', 'Ж'));
- Удаление домена
  - **drop domain** Color restrict;
  - **drop domain** Gender cascade;
- Изменение домена
  - **alter domain** Gender char  
**constraint** chk\_gender (check (value in 'M', 'Ж'));

# Язык манипулирования данными DML

- Вставка записей в таблицу: INSERT
- Обновление записей в таблице: UPDATE
- Удаление записей из таблицы: DELETE
- Выборка записей из таблиц: SELECT

# Вставка данных в таблицу

**insert**

**into** *таблица [список полей]*

**values** (*список значений*) / *запрос на выборку;*

- Вставка одной строки

- **insert**

- into** P (PID, Color, Name, Weight, City)

- values** ('P10', 'белый', 'Брус', 3, 'Москва');

- Вставка нескольких строк

- **insert**

- into** AvgRatings (City, Rating)

- select** S.City, avg(S.Rating)

- from** S

- group by** S.City;

# Обновление данных в таблице

**update** *таблица*

**set** *СПИСОК ОБНОВЛЯЕМЫХ ПОЛЕЙ*

**[where** *условие отбора обновляемых записей*];

- Обновление одной строки

- **update** S

- set** S.SID='S007', S.Name='Бонд'
    - where** S.SID='S7';

- Обновление нескольких строк

- **update** S

- set** S.Rating=S.Rating+1
    - where** S.Rating<(select avg(S.Rating) from S);

- **update** P

- set** P.City = (select S.City from S where S.SID='S007')
    - where** P.Color='белый';



# Удаление данных из таблицы

```
delete [ * ]  
from таблица  
[where условие отбора удаляемых записей];
```

- Удаление одной строки
  - **delete**  
**from** S  
**where** S.SID='S007';
- Удаление нескольких строк
  - **delete**  
**from** S  
**where** S.Rating<(select avg(S.Rating) from S);
- Удаление всех строк
  - **delete from** SP;

# Выборка данных из таблиц

**select** [**distinct**] список выбираемых полей  
**from** список таблиц выбора  
**[where** условие отбора записей]  
**[order by** список упорядочиваемых полей]  
**[group by** список группируемых полей]  
**[having** условие отбора групп];

# Выборка всех строк

- Выдать полную информацию о деталях.

```
select P.*  
from P;
```

```
select *  
from P;
```

**P**

PID	Name	City	Weight	Price
P1	Болт	Париж	15	72
P34	Шуруп	Москва	29	108
P22	Гайка	Челябинск	20	18
P45	Шуруп	Одесса	14	108

**Результат**

PID	Name	City	Weight	Price
P1	Болт	Париж	15	72
P34	Шуруп	Москва	29	108
P22	Гайка	Челябинск	20	18
P45	Шуруп	Одесса	14	108

# Выборка по условию

- Выдать список названий и цен деталей не из Парижа с весом более 10.

```
select P.Name, P.Price  
from P  
where P.City<>'Париж' and  
P.Weight>10;
```

**P**

PID	Name	City	Weight	Price
P1	Болт	Париж	15	40
P34	Шуруп	Москва	29	33
P22	Гайка	Челябинск	20	24
P45	Шуруп	Одесса	14	33

**Результат**

Name	Price
Шуруп	33
Гайка	24
Шуруп	33

# Выборка без повторяющихся значений

- Выдать список названий и цен деталей не из Парижа с весом более 10.

```
select distinct P.Name  
from P  
where P.City<>'Париж' and  
P.Weight>10;
```

**P**

PID	Name	City	Weight	Price
P1	Болт	Париж	15	40
P34	Шуруп	Москва	29	33
P22	Гайка	Челябинск	20	24
P45	Шуруп	Одесса	14	33

**Результат**

Name
Шуруп
Гайка

# Выборка с упорядочиванием

- Выдать список названий и цен деталей (без повторений) не из Парижа с весом более 10, упорядоченный по названию детали.

```
select distinct P.Name, P.Price  
from P  
where P.City<>'Париж' and  
P.Weight>10  
order by P.Name;
```

**P**

PID	Name	City	Weight	Price
P1	Болт	Париж	15	40
P34	Шуруп	Москва	29	33
P22	Гайка	Челябинск	20	24
P45	Шуруп	Одесса	14	33

**Результат**

Name	Price
Гайка	24
Шуруп	33

# Выборка с упорядочиванием

- Выдать список названий и цен деталей (без повторений) не из Парижа с весом более 10, упорядоченный **по убыванию** цены.

```
select distinct P.Name, P.Price  
from P  
where P.City<>'Париж' and  
P.Weight>10  
order by P.Price desc;
```

**P**

PID	Name	City	Weight	Price
P1	Болт	Париж	15	40
P34	Шуруп	Москва	29	33
P22	Гайка	Челябинск	20	24
P45	Шуруп	Одесса	14	33

**Результат**

Name	Price
Шуруп	33
Гайка	24

# Вычисления в выборке

- Выдать название деталей и цену деталей в €.

```
select distinct P.Name,  
P.Price/70.00 as Price_EUR  
from P;
```

**P**

PID	Name	City	Weight	Price
P1	Болт	Париж	15	72
P34	Шуруп	Москва	29	108
P22	Гайка	Челябинск	20	18
P45	Шуруп	Одесса	14	108

**Результат**

Name	Price_EUR
Болт	1.02
Шуруп	1.54
Гайка	0,26



# Выборка из нескольких таблиц

- Выдать список всех пар поставщиков и деталей, размещенных в одном городе.

```
select S.SID, S.SName, S.City, P.PID,  
P.PName  
from P, S  
where S.City=P.City;
```

Соединение таблиц P и S:

P.PID	P.PName	P.City	P.Weight	P.Price	S.SID	S.SName	S.City	S.Rating
P1	Болт	Париж	15	72	S1	Бендер	Черноморск	15
P1	Болт	Париж	15	72	S2	КГБ	Челябинск	20
P34	Шуруп	Челябинск	29	108	S1	Бендер	Черноморск	15
P34	Шуруп	Челябинск	29	108	S2	КГБ	Челябинск	20
P22	Гайка	Челябинск	20	18	S1	Бендер	Черноморск	15
P22	Гайка	Челябинск	20	18	S2	КГБ	Челябинск	20

**P**

PID	PName	City	Weight	Price
P1	Болт	Париж	15	72
P34	Шуруп	Челябинск	29	108
P22	Гайка	Челябинск	20	18

**S**

SID	SName	City	Rating
S1	Бендер	Черноморск	15
S2	КГБ	Челябинск	20

# Выборка из нескольких таблиц

- Выдать список всех пар поставщиков и деталей, размещенных в одном городе.

```
select S.SID, S.SName, S.City, P.PID,  
P.PName  
from P, S  
where S.City=P.City;
```

Выборка строк по условию S.City=P.City:

P.PID	P.PName	P.City	P.Weight	P.Price	S.SID	S.SName	S.City	S.Rating
P1	Болт	Париж	15	72	S1	Бендер	Черноморск	15
P1	Болт	Париж	15	72	S2	КГБ	Челябинск	20
P34	Шуруп	Челябинск	29	108	S1	Бендер	Черноморск	15
P34	Шуруп	Челябинск	29	108	S2	КГБ	Челябинск	20
P22	Гайка	Челябинск	20	18	S1	Бендер	Черноморск	15
P22	Гайка	Челябинск	20	18	S2	КГБ	Челябинск	20

**P**

PID	PName	City	Weight	Price
P1	Болт	Париж	15	72
P34	Шуруп	Челябинск	29	108
P22	Гайка	Челябинск	20	18

**S**

SID	SName	City	Rating
S1	Бендер	Черноморск	15
S2	КГБ	Челябинск	20

# Выборка из нескольких таблиц

- Выдать список всех пар поставщиков и деталей, размещенных в одном городе.

```
select S.SID, S.SName, S.City, P.PID,  
P.PName  
from P, S  
where S.City=P.City;
```

«Отбрасывание» лишних столбцов:

P.PID	P.PName	P.City	P.Weight	P.Price	S.SID	S.SName	S.City	S.Rating
P34	Шуруп	Челябинск	29	108	S2	КГБ	Челябинск	20
P22	Гайка	Челябинск	20	18	S2	КГБ	Челябинск	20

**P**

PID	PName	City	Weight	Price
P1	Болт	Париж	15	72
P34	Шуруп	Челябинск	29	108
P22	Гайка	Челябинск	20	18

**S**

SID	SName	City	Rating
S1	Бендер	Черноморск	15
S2	КГБ	Челябинск	20

**Результат**

S.SID	S.SName	S.City	P.PID	P.PName
S2	КГБ	Челябинск	P34	Шуруп
S2	КГБ	Челябинск	P22	Гайка

# Выборка из нескольких таблиц

- Выдать список всех пар поставщиков и деталей, размещенных в одном городе.

```
select S.SID, S.SName, S.City, P.PID,  
P.PName  
from P, S  
where S.City=P.City;
```

```
select S.SID, S.SName, S.City, P.PID,  
P.PName  
from S join P using City;
```

```
select S.SID, S.SName, S.City, P.PID,  
P.PName  
from S natural join P;
```

**P**

PID	PName	City	Weight	Price
P1	Болт	Париж	15	72
P34	Шуруп	Челябинск	29	108
P22	Гайка	Челябинск	20	18
P45	Шуруп	Одесса	14	108

**S**

SID	SName	City	Rating
S1	Бендер	Черноморск	15
S34	ИТ-монстр	Москва	29
S2	КГБ	Челябинск	20

**Результат**

SID	SName	City	PID	PName
S2	КГБ	Челябинск	P34	Шуруп
S2	КГБ	Челябинск	P22	Гайка

# Соединение таблицы с собой

- Получить все пары кодов поставщиков, таких, что оба поставщика в каждой паре из одного и того же города.

```
select First.SID as SA, Second.SID  
as SB
```

```
from S as First, S as Second
```

```
where First.City=Second.City and  
SA<>SB;
```

**S**

SID	Name	City	Rating
S1	Бендер	Черноморск	15
S3	ИТ-монстр	Москва	29
S2	КГБ	Челябинск	20
S7	Разноимпорт	Москва	31
S5	Саспенс+	Челябинск	2
S4	Horns&Hoofs	Черноморск	15

**Результат**

SA	SB
S1	S4
S4	S1
S3	S7
S7	S3
S2	S5
S5	S2

# Выборка с соединением по ключу

- Получить имена поставщиков, поставляющих деталь P2.
  - **select distinct S.Name**  
**from S, SP**  
**where S.SID=SP.SID and SP.PID='P2';**

S	
SID	Name
S1	Саспенс +
S2	ИТ-монстр
S3	Разноимпорт
S5	МакроХард
S7	Овощторг
S8	Разноимпорт

SP		
SID	PID	Qty
S1	P5	10
S3	P2	20
S2	P2	30
S7	P1	40
S5	P4	50
S8	P2	30

## Результат

Name
ИТ-монстр
Разноимпорт

# Агрегирующие функции языка SQL

- **count** определяет количество записей данного поля в группе строк.
- **sum** вычисляет арифметическую сумму всех выбранных значений данного поля.
- **avg** рассчитывает арифметическое среднее (усреднение) всех выбранных значений данного поля.
- **max** находит наибольшее из всех выбранных значений данного поля.
- **min** находит наименьшее из всех выбранных значений данного поля.

# Выборка с агрегацией данных

- Получить общее число поставщиков.

```
select 'Общее число поставщиков' as Text,  
count(*) as CountS  
from S;
```

**S**

SID	Name	City	Rating
S1	Бендер	Черноморск	15
S3	ИТ-монстр	Москва	29
S2	КГБ	Челябинск	20
S7	Разноимпорт	Москва	31
S5	Саспенс+	Челябинск	2
S4	Horns&Hoofs	Черноморск	15

**Результат**

Text	CountS
Общее число поставщиков	6



# Выборка с агрегацией данных

- Получить сумму рейтингов поставщиков из города Черноморск.
  - **select** **sum**(S.Rating) **as** RatingChe  
**from** S  
**where** S.City = 'Черноморск';

**S**

SID	Name	City	Rating
S1	Бендер	Черноморск	15
S3	ИТ-монстр	Москва	29
S2	КГБ	Челябинск	20
S7	Разноимпорт	Москва	31
S5	Саспенс+	Челябинск	2
S4	Horns&Hoofs	Черноморск	15

## Результат

RatingChe
17

# Выборка с агрегацией данных

- Получить минимальное, максимальное и среднее количество поставок для детали P1

```
select max(SP.Qty) as MaxQ,  
min(SP.Qty) as MinQ, avg(SP.Qty) as  
AvgQ  
from SP  
where SP.PID='P1';
```

## SP

SID	PID	Qty
S1	P1	10
S3	P2	20
S2	P1	30
S7	P1	40
S5	P3	50
S4	P1	80

## Результат

MaxQ	MinQ	AvgQ
80	10	40

# Выборка с подзапросом

- Получить имена поставщиков, поставляющих деталь P2.

```
select distinct S.Name  
from S  
where S.SID in  
  (select SP.SID  
   from SP  
   where SP.PID='P2');
```

```
select distinct S.Name  
from S, SP  
where S.SID = SP.SID and SP.PID='P2';
```

**S**

SID	Name
S1	Саспенс +
S2	ИТ-монстр
S3	Разноимпорт
S5	МакроХард
S7	Овощторг
S8	Разноимпорт

**SP**

SID	PID	Qty
S1	P5	10
S3	P2	20
S2	P2	30
S7	P1	40
S5	P4	50
S8	P2	30

**Результат**

Name
ИТ-монстр
Разноимпорт