



# ЯЗЫК ЗАПРОСОВ SQL

*Мир – это огромная база данных со сложным языком запросов, но большинство пользователей обходится только одним ключевым словом – "Дай".*  
С. Янковский

Технологии баз данных

---

---

---

---

---

---

---

---

## Содержание

2

- Историческая справка
- Основные команды SQL
- Встроенный SQL
- Динамический SQL
- Тренинг написания запросов
- SQL/PSM

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

---

## SQL

3

- SQL (*Structured Query Language, язык структурных запросов*) – стандартный язык реляционных баз данных.
- В базовом варианте SQL является информационно-логическим языком, а не языком программирования.
- Спецификация SQL/PSM (*Persistent Stored Modules, хранящиеся процедуры*) представляет собой процедурное расширение SQL.
- SQL основан на реляционной алгебре.
- Составные части SQL:
  - язык определения данных (*Data Definition Language, DDL*)
  - язык манипулирования данными (*Data Manipulation Language, DML*)
  - язык определения доступа к данным (*Data Control Language, DCL*)
  - язык управления транзакциями (*Transaction Control Language, TCL*)

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

---

## SQL: историческая справка

Год	Название	Отличительные особенности
1970	SEQUEL	Язык управления данными экспериментальной СУБД IBM System R (SEQUEL – Structured English QUEry Language, структурированный английский язык запросов).
1986	SQL-86 или SQL-87	Первый вариант стандарта, принятый институтом ANSI и одобренный ISO в 1987.
1989	SQL-89	Доработанный вариант предыдущего стандарта.
1992	SQL-92	Значительные изменения предыдущего стандарта.
1999	SQL:1999 или SQL-3	Добавлена поддержка регулярных выражений, рекурсивных запросов, поддержка триггеров, базовые процедурные расширения, не скалярные 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.

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

---

## SQL: плюсы и минусы

- **Преимущества**
  - ▣ Независимость от конкретной СУБД
  - ▣ Стандарты
  - ▣ Декларативность
- **Недостатки**
  - ▣ Несоответствие классической РМД
    - Таблица ≠ отношение, строка ≠ кортеж и др.
  - ▣ Сложность
    - Избыточность языковых средств
    - Большой объем стандарта (более 1000 стр.)
  - ▣ Отступления от стандартов
    - "Надмножества подмножества SQL": Oracle PL/SQL, MS TransactSQL, PL/pgSQL и др.

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

---

## Команды SQL

- **DDL**
  - ▣ Создание, удаление, изменение определения доменов.
  - ▣ Создание, удаление, изменение определения таблиц.
- **DML**
  - ▣ Выборка записей из таблиц.
  - ▣ Вставка записей в таблицу.
  - ▣ Обновление записей в таблице.
  - ▣ Удаление записей из таблицы.

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

---

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 'Ж';`  
`constraint chk_gender (check (value in 'М', 'Ж'));`
- Удаление домена
  - `drop domain Color restrict;`
  - `drop domain Gender cascade;`
- Изменение домена
  - `alter domain Gender char`  
`constraint chk_gender (check (value in 'М', 'Ж'));`

Технологии баз данных © М.Л. Цымбалер

---

---

---

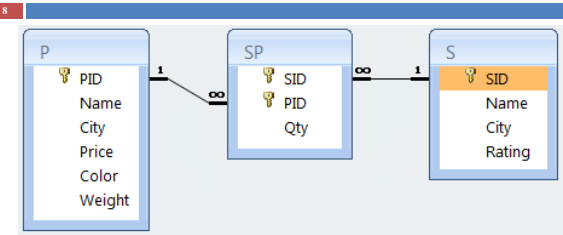
---

---

---

---

Модельная база данных



- S – Поставщики
- P – Детали
- SP – Поставки

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

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**;  
`)`
- `create table P (`  
PID char(4) **primary key**,  
Name char(10) **not null**,  
City City char(10) **not null**,  
Price int **not null**,  
Color char(10) **not null**,  
Weight float **not null**;  
`)`
- `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**,  
**check** (Qty>0);  
`)`

\*Допустимы режимы cascade, set null, set default и no action.  
Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

## Операции с таблицами: изменение и удаление

10

- **alter table P**  
**add column** Discount float default 0;
- **alter table P**  
**add constraint** check (Price>0) ,  
**add constraint** check (Weight>0);
- **drop table S** restrict;
- **drop table SP** cascade;

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

---

## Операции со словарем БД

11

- Доступные по чтению таблицы словаря
  - ▣ SCHEMATA
  - ▣ DOMAINS
  - ▣ TABLES
  - ▣ VIEWS
  - ▣ COLUMNS
  - ▣ TABLE\_PRIVILEGES
  - ▣ COLUMN\_PRIVILEGES
  - ▣ TABLE\_CONSTRAINTS
  - ▣ REFERENTIAL\_CONSTRAINTS
  - ▣ и др.

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

---

## Выборка данных из таблиц

12

- **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];*

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

---

Выборка

13

- 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

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

Выборка

14

- 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

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

Выборка

15

- 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

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

Выборка

16

- Выдать название деталей и цену деталей в €.
- select distinct P.Name, P.Цена/36.00 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	2
Шуруп	3
Гайка	0,5

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

Выборка

17

- Выдать список всех пар поставщиков и деталей, размещенных в одном городе.
- select S.SID, S.Name, S.City, P.PID, P.Name from P, S where S.City=P.City;
- select S.SID, S.Name, S.City, P.PID, P.Name from S join P using City;
- select S.SID, S.Name, S.City, P.PID, P.Name from S natural join P;

P

PID	Name	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	Name	City	Rating
S1	Бендер	Черноморск	15
S34	ИТ-монстр	Москва	29
S2	КТБ	Челябинск	20

Результат

PID	S.Name	City	PID	P.Name
S2	КТБ	Челябинск	P34	Шуруп
S2	КТБ	Челябинск	P22	Гайка

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

Выборка

18

- 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

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

Выборка

19

- 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	КТБ	Челябинск	20
S7	Разномиллорт	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

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

Выборка

20

- 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	КТБ	Челябинск	20
S7	Разномиллорт	Москва	31
S5	Саспенс+	Челябинск	2
S4	Horns&Hoofs	Черноморск	15

Результат

Text	CountS
Общее число поставщиков	6

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

Выборка

21

- 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

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

Выборка

23

- Получить код и общее количество поставки каждой детали.
  - `select SP.PID, sum(SP.Qty) as Total from SP group by SP.PID;`
  - `select P.PID, (select sum(SP.Qty) from SP where SP.PID=P.PID) as Total from P;`

SP		
SID	PID	Qty
S1	P5	10
S3	P2	20
S2	P4	30
S7	P1	40
S5	P5	20
S5	P3	50
S8	P2	30
S4	P1	80

Результат	
PID	Total
P5	30
P2	50
P4	30
P1	120
P3	50

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

---

Выборка

23

- Получить коды деталей, поставляемых более чем одним поставщиком.
  - `select SP.PID from SP group by SP.PID having count(SP.SID)>1;`

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

Результат	
PID	
P2	
P4	

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

---

Выборка

24

- Получить названия деталей, поставляемых более чем одним поставщиком.
  - `select distinct P.Name from P where P.PID in (select SP.PID from SP group by SP.PID having count(SP.SID)>1);`

P		SP		
PID	Name	SID	PID	Qty
P1	Болт	S1	P5	10
P2	Гайка	S3	P2	20
P3	Шуруп	S2	P4	30
P4	Гайка	S7	P1	40
P5	Уголок	S5	P4	50
		S8	P2	30

Результат	
Name	
Гайка	

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

---



Выборка

25

□ 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');`

S		SP		
SID	Name	SID	PID	Qty
S1	Системс +	S1	P5	10
S2	ИТ-монстр	S3	P2	20
S3	Разномпорт	S2	P2	30
S5	МакроХард	S7	P1	40
S7	Овоцторг	S5	P4	50
S8	Разномпорт	S8	P2	30

Результат

Name
ИТ-монстр
Разномпорт

---

---

---

---

---

---

---

---

Выборка

26

□ Получить имена поставщиков, поставляющих деталь P2.

■ `select distinct S.Name  
from S  
where exists (  
select *  
from SP  
where SP.SID=S.SID and  
SP.PID='P2');`

S		SP		
SID	Name	SID	PID	Qty
S1	Системс +	S1	P5	10
S2	ИТ-монстр	S3	P2	20
S3	Разномпорт	S2	P2	30
S5	МакроХард	S7	P1	40
S7	Овоцторг	S5	P4	50
S8	Разномпорт	S8	P2	30

Результат

Name
ИТ-монстр
Разномпорт

---

---

---

---

---

---

---

---

Выборка

27

□ Получить имена поставщиков, поставляющих деталь P2.

■ `select distinct S.Name  
from S,  
(select S.SID  
from S, SP  
where SP.SID=S.SID and  
SP.PID='P2') as SIDs_P2  
where S.SID=SIDs_P2.SID;`

S		SP		
SID	Name	SID	PID	Qty
S1	Системс +	S1	P5	10
S2	ИТ-монстр	S3	P2	20
S3	Разномпорт	S2	P2	30
S5	МакроХард	S7	P1	40
S7	Овоцторг	S5	P4	50
S8	Разномпорт	S8	P2	30

Результат

Name
ИТ-монстр
Разномпорт

---

---

---

---

---

---

---

---

Выборка

28

□ Получить имена поставщиков, поставляющих деталь P2.

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

S		SP		
SID	Name	SID	PID	Qty
S1	Системс +	S1	P5	10
S2	ИТ-монстр	S3	P2	20
S3	Разномпорт	S2	P2	30
S5	МакроХард	S7	P1	40
S7	Овоцторг	S5	P4	50
S8	Разномпорт	S8	P2	30

Результат

Name
ИТ-монстр
Разномпорт

---

---

---

---

---

---

---

---

Выборка

29

□ Получить имена поставщиков, поставляющих деталь P2.

```
select distinct S.Name
from S
where 0 <
(
select count(*)
from SP
where S.SID=SP.SID
and SP.PID='P2'
);
```

S		SP		
SID	Name	SID	PID	Qty
S1	Системс +	S1	P5	10
S2	ИТ-монстр	S3	P2	20
S3	Разномпорт	S2	P2	30
S5	МакроХард	S7	P1	40
S7	Овоцторг	S5	P4	50
S8	Разномпорт	S8	P2	30

Результат

Name
ИТ-монстр
Разномпорт

---

---

---

---

---

---

---

---

Выборка

30

□ Получить имена поставщиков, поставляющих деталь P2.

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

S		SP		
SID	Name	SID	PID	Qty
S1	Системс +	S1	P5	10
S2	ИТ-монстр	S3	P2	20
S3	Разномпорт	S2	P2	30
S5	МакроХард	S7	P1	40
S7	Овоцторг	S5	P4	50
S8	Разномпорт	S8	P2	30

Результат

Name
ИТ-монстр
Разномпорт

---

---

---

---

---

---

---

---

Выборка

31

Получить имена поставщиков, поставляющих деталь P2.

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

S		SP		
SID	Name	SID	PID	Qty
S1	Сасленс +	S1	P5	10
S2	ИТ-монстр	S3	P2	20
S3	Разномпорт	S2	P2	30
S5	МакроХард	S7	P1	40
S7	Овоцторг	S5	P4	50
S8	Разномпорт	S8	P2	30

Результат

Name
ИТ-монстр
Разномпорт

---

---

---

---

---

---

---

---

Выборка

32

Получить имена поставщиков, НЕ поставляющих деталь P2.

```
select distinct S.Name
from S
where not exists (
select *
from SP
where SP.SID=S.SID and P.PID='P2');
select distinct S.Name
from S
where S.SID not in (
select SP.SID
from SP
where SP.PID='P2');
```

S		SP		
SID	Name	SID	PID	Qty
S1	Овоцторг	S1	P5	10
S2	ИТ-монстр	S3	P2	20
S3	Разномпорт	S2	P2	30
S5	МакроХард	S7	P1	40
S7	Овоцторг	S5	P4	50
S8	Разномпорт	S8	P2	30

Результат

Name
Овоцторг
МакроХард

---

---

---

---

---

---

---

---

Выборка

33

Получить имена поставщиков, поставляющих красные детали.

```
select distinct S.Name
from S
where S.SID in
(select SP.SID
from SP
where SP.PID in
(select P.PID
from P
where P.Color='красный'));
select distinct S.Name
from S, P, SP
where S.SID=SP.SID and
SP.PID=P.PID and P.Color='красный';
```

S		SP		
SID	Name	SID	PID	Qty
S1	Овоцторг	S1	P5	10
S2	ИТ-монстр	S3	P2	20
S3	Разномпорт	S2	P2	30
S5	МакроХард	S7	P1	40
S7	Овоцторг	S5	P4	50
S8	Разномпорт	S8	P2	30

P	
PID	Color
P1	красный
P2	зеленый
P3	синий
P4	красный
P5	красный

Результат

Name
Овоцторг
МакроХард

---

---

---

---

---

---

---

---

Выборка

34

- Получить имена поставщиков, поставляющих ВСЕ детали.
  - ▣ `select distinct S.Name from S where not exists (select * from P where not exists (select * from SP where SP.SID=S.SID and SP.PID=P.PID));`

S		SP		
SID	Name	SID	PID	Qty
S1	ИТ-монстр	S1	P1	10
S2	ИТ-монстр	S1	P2	20
S3	Разноимпорт	S1	P3	30
S5	МакроХард	S2	P1	40
S7	Овоцторг	S2	P2	50
S8	Разноимпорт	S2	P3	30
		S3	P3	20
		S7	P2	100

P		
PID	Name	Price
P1	Болт	200
P2	Гайка	10
P3	Болт	150

Результат	
Name	
ИТ-монстр	

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

Выборка

35

- Получить имена поставщиков, поставляющих ВСЕ детали.
  - ▣ `select distinct S.Name from S where (select count(SP.PID) from SP where SP.SID=S.SID) = (select count(P.PID) from P);`

S		SP		
SID	Name	SID	PID	Qty
S1	ИТ-монстр	S1	P1	10
S2	ИТ-монстр	S1	P2	20
S3	Разноимпорт	S1	P3	30
S5	МакроХард	S2	P1	40
S7	Овоцторг	S2	P2	50
S8	Разноимпорт	S2	P3	30
		S3	P3	20
		S7	P2	100

P		
PID	Name	Price
P1	Болт	200
P2	Гайка	10
P3	Болт	150

Результат	
Name	
ИТ-монстр	

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

Выборка

36

- Получить названия деталей, которые либо имеют цену более 100, либо поставляются поставщиками из Челябинска.
  - ▣ `select P.Name from P, SP where SP.PID=P.PID and P.Price>100 union select P.Name from S, P, SP where SP.PID=P.PID and SP.SID=S.SID and S.City='Челябинск';`

S		SP		
SID	City	SID	PID	Qty
S1	Москва	S1	P1	10
S2	Урюпинск	S2	P1	40
S3	Челябинск	S3	P5	20
S5	Одесса	S7	P2	100
S7	Челябинск	S8	P2	300
S8	Челябинск	S5	P4	500

P		
PID	Name	Price
P1	Болт	200
P2	Гайка	10
P3	Болт	150
P4	Уголок	15
P5	Шуруп	130

Результат	
Name	
Гайка	
Болт	
Шуруп	

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

Выборка

37

- Получить названия деталей, которые имеют цену более 100 и поставляются поставщиками из Челябинска.
  - `select P.Name from P, SP where SP.PID=P.PID and P.Price>100 intersect select P.Name from S, P, SP where SP.PID=P.PID and SP.SID=S.SID and S.City='Челябинск';`

S		SP		
SID	City	SID	PID	Qty
S1	Москва	S1	P1	10
S2	Урюпинск	S2	P1	40
S3	Челябинск	S3	P5	20
S5	Одесса	S7	P2	100
S7	Челябинск	S8	P5	300
S8	Челябинск	S5	P4	500

PID	Name	Price
P1	Болт	200
P2	Гайка	10
P3	Болт	150
P4	Уголок	15
P5	Шуруп	130

Результат	
Name	
Болт	
Шуруп	

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

Выборка

38

- Получить названия деталей, которые имеют цену более 100 и поставляются поставщиками НЕ из Челябинска.
  - `select P.Name from P, SP where SP.PID=P.PID and P.Price>100 except select P.Name from S, P, SP where SP.PID=P.PID and SP.SID=S.SID and S.City='Челябинск';`

S		SP		
SID	City	SID	PID	Qty
S1	Москва	S1	P1	10
S2	Урюпинск	S2	P3	40
S3	Челябинск	S3	P5	20
S5	Одесса	S7	P2	100
S7	Челябинск	S8	P5	300
S8	Челябинск	S5	P4	500

PID	Name	Price
P1	Болт	200
P2	Гайка	10
P3	Болт	150
P4	Уголок	15
P5	Шуруп	130

Результат	
Name	
Болт	

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

Выборка

39

- Получить названия деталей, цена которых больше цены любой детали из Челябинска.
  - `select distinct P.Name from P where P.Price > all (select P.Price from P where P.City='Челябинск');`

PID	Name	Price	City
P1	Болт	30	Челябинск
P2	Брус	99	Шенетовка
P3	Болт	50	Челябинск
P4	Уголок	40	Одесса
P5	Шуруп	23	Челябинск
P6	Гайка	77	Москва
P7	Болт	88	Конейск

Результат	
Name	
Брус	
Гайка	
Болт	

Технологии баз данных © М.Л. Цымблер

---

---

---

---

---

---

---

---

## Выборка

□ Получить коды, имена и семантику рейтинга поставщиков.

```
□ select S.SID, S.Name,
  case
    when S.Rating < 10 then 'сомнительный'
    when S.Rating < 15 then 'посредственный'
    when S.Rating < 20 then 'приемлемый'
    when S.Rating < 25 then 'надежный'
  end as Rating
from S;
```

---

---

---

---

---

---

---

## Вставка данных в таблицу

```
□ insert
  into таблица [список полей]
  values (список значений);
□ 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);
  - ❑ **delete from** SP;

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

# Встроенный SQL

- ❑ Стандарт SQL поддерживает возможность *встраивания команд SQL в программу на языке программирования* (Ada, C, Pascal и др.).
- ❑ В команду SQL могут передаваться переменные программы на базовом языке. В команде SQL можно изменять значения этих переменных.
- ❑ В программе на базовом языке можно анализировать результат выполнения команды SQL. В программе на базовом языке можно построчно обрабатывать результат SQL запроса.

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

# Встроенный SQL

```
/* Стоимость поставок заданного поставщика или -1 в случае ошибки*/
int SumSP (char mySID[4])
{
    EXEC SQL begin declare section;
    float sum;
    char SQLSTATE[6];
    EXEC SQL end declare section;

    EXEC SQL
    select sum(SP.Qty*P.Price) into :sum
    from S, P, SP
    where SP.SID = :mySID and SP.PID=P.PID;

    if (strcmp(SQLSTATE,'00000'))
        return sum;
    else
        return -1;
}
```

Технологии баз данных © М.Л. Цымбалер

- Секция объявления общих переменных
- Встраивание SQL кода
- Использование внешней переменной
- Изменение внешней переменной
- Код завершения последней команды SQL

---

---

---

---

---

---

---

## Курсоры

46

- *Курсор* – указатель на область памяти, в которой хранится результат выполнения запроса SQL.
- Курсоры используются для сканирования результирующих отношений в программе на базовом языке программирования.
- Курсор представляет собой объект с методами
  - **open** – открыть
  - **fetch** – передвинуть курсор на следующую строку
  - **close** – закрыть

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

## Курсоры

47

/\* Покортежная обработка результата запроса \*/

void ProcessS (void)

```
{  
    EXEC SQL begin declare section;  
    char cSID[4], cName[10], cCity[10];  
    int cRating;  
    char SQLSTATE[6];  
    EXEC SQL end declare section;
```

Секция объявления  
общих переменных

```
    EXEC SQL declare myCursor cursor for  
    select * from S;
```

Объявление курсора

```
    EXEC SQL open myCursor;
```

Открытие курсора

```
    while (1) {
```

```
        EXEC SQL fetch from myCursor into :cSID, :cName, :cCity, :cRating;
```

Извлечение записи

```
        if (strcmp(SQLSTATE, '02000')) break;  
        /*Обработка :cSID, :cName, :cCity, :cRating */
```

```
    }  
    EXEC SQL close myCursor;  
}
```

Закрытие курсора

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

## Курсоры

48

/\* Удаление и обновление записей базовой таблицы посредством курсора \*/

void ProcessS (void)

```
{  
    EXEC SQL begin declare section;  
    char cSID[4], cName[10], cCity[10]; int cRating;  
    char SQLSTATE[6];  
    EXEC SQL end declare section;
```

```
    EXEC SQL declare myCursor cursor for
```

```
    select * from S;
```

```
    EXEC SQL open myCursor;
```

```
    while (1) {
```

```
        EXEC SQL fetch from myCursor into :cSID, :cName, :cCity, :cRating;
```

```
        if (strcmp(SQLSTATE, '02000')) break;
```

```
        EXEC SQL delete from S where current of myCursor;
```

Удаление записи

```
        EXEC SQL update S set ... where current of myCursor;
```

Обновление записи

```
    }  
    EXEC SQL close myCursor;  
}
```

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---



# Динамический SQL

□ Динамический SQL дает возможность выполнить заранее неизвестный запрос, текст которого задан в строковой переменной.

```
void ExecUserQuery(char * query)
```

```
{ EXEC SQL prepare queryPlan from :query
```

```
EXEC SQL execute queryPlan;
```

Переменная SQL

Общая переменная

Подготовка выполнения запроса

Выполнение запроса

---

---

---

---

---

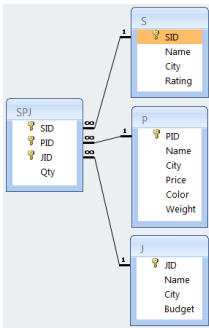
---

---

---

# Модельная база данных

- S – Поставщики
- P – Детали
- J – Проекты
- SPJ – Поставки



---

---

---

---

---

---

---

---

# Запрос

- Получить имена поставщиков проекта J1.
- select distinct S.Name  
from S, SPJ  
where SPJ.JID='J1' and S.SID=SPJ.SID

---

---

---

---

---

---

---

---

Запрос

53

- Получить имя поставщика, имя детали, имя проекта и количество деталей для поставок деталей из Челябинска.
- `select distinct S.Name, P.Name, J.Name, SPJ.Qty  
from S, P, SPJ  
where S.SID=SPJ.SID and P.PID=SPJ.PID and  
P.City='Челябинск'`

---

---

---

---

---

---

---

Запрос

53

- Получить все тройки "код поставщика – код детали – код проекта", где поставщик, деталь и проект размещены в одном городе.
- `select S.SID, P.PID, J.JID  
from S, P, J  
where S.City=P.City and P.City=J.City`

---

---

---

---

---

---

---

Запрос

54

- Получить все тройки "код поставщика – код детали – код проекта", где поставщик, деталь и проект НЕ размещены в одном городе.
- `select S.SID, P.PID, J.JID  
from S, P, J  
where not (S.City=P.City and P.City=J.City)`

---

---

---

---

---

---

---

Запрос

55

- Получить все тройки "код поставщика – код детали – код проекта", где никакие из двух выводимых поставщиков, деталей и проектов НЕ размещены в одном городе.
- ```
select S.SID, P.PID, J.JID
from S, P, J
where S.City<>P.City and P.City<>J.City and
J.City<>P.City
```

---

---

---

---

---

---

---

Запрос

56

- Получить коды деталей, поставляемых поставщиками из Челябинска.
- ```
select distinct SPJ.PID
from SPJ
where 'Челябинск' = (
    select S.City
    from S
    where S.SID=SPJ.SID)
```

---

---

---

---

---

---

---

Запрос

57

- Получить названия деталей, поставляемых поставщиками из Челябинска для проектов в Челябинске.
- ```
select distinct P.Name
from SPJ, P
where 'Челябинск' = (
    select S.City
    from S
    where S.SID=SPJ.SID)
and 'Челябинск' = (
    select J.City
    from J
    where J.JID=SPJ.JID)
and SPJ.PID=P.PID
```

---

---

---

---

---

---

---

Запрос

58

- Получить все пары названий городов, где поставщик из первого города обеспечивает проект из второго города.
- ```
select distinct S.City as SCity, J.City as JCity
from S, J
where exists (
  select *
  from SPJ
  where S.SID=SPJ.SID and J.JID=SPJ.JID)
```

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

Запрос

59

- Получить номера деталей, поставляемых для всех проектов, обеспечиваемых поставщиком из того же города, где размещен проект.
- ```
select distinct SPJ.PID
from SPJ
where (select S.City
  from S
  where S.SID=SPJ.SID) =
  (select J.City
  from J
  where J.JID=SPJ.JID)
```

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

Запрос

60

- Получить номера проектов, обеспечиваемых по крайней мере одним поставщиком из другого города.
- ```
select distinct SPJ.JID
from SPJ
where (select S.City
  from S
  where S.SID=SPJ.SID) <>
  (select J.City
  from J
  where J.JID=SPJ.JID)
```

Технологии баз данных © М.Л. Цымбалер

---

---

---

---

---

---

---

Запрос

61

- Получить общее количество проектов, обеспечиваемых поставщиком S1.
- `select count (distinct SPJ.JID) as N  
from SPJ  
where SPJ.SID='S1'`

---

---

---

---

---

---

---

Запрос

62

- Для каждой детали, поставляемой для проекта, получить номер детали, номер проекта и общее количество.
- `select SPJ.PID, SPJ.JID, sum(SPJ.Qty) as N  
from SPJ  
group by SPJ.PID, SPJ.JID`

---

---

---

---

---

---

---

Запрос

63

- Получить номера поставщиков таких деталей, которые поставляются поставщиком, поставляющим красные детали.
- `select SPJa.SID  
from SPJ as SPJa, SPJ as SPJb, SPJ as SPJc  
where SPJa.PID=SPJb.PID and SPJb.SID=SPJc.SID  
and 'красный'=(  
select P.Color  
from P  
where P.PID=SPJc.PID)`

---

---

---

---

---

---

---

Запрос

64

- Получить номера проектов из города, стоящего последним по алфавиту.
- ```
select J.JID
from J
where J.City=(select max(J.City) from J)
```

---

---

---

---

---

---

---

Запрос

65

- Получить номера поставщиков, поставляющих одну и ту же деталь для всех проектов.
- ```
select S.SID
from S
where exists (
  select * from P
  where not exists (
    select * from J
    where not exists (
      select * from SPJ
      where SPJ.SID=S.SID and
            SPJ.PID=P.PID and SPJ.JID=J.JID)))
```

---

---

---

---

---

---

---

SQL/PSM

66

```
create procedure Statistics(
  aliveS out number, deadS out number,
  aliveP out number, deadP out number) as
declare
  t number;
begin
  select count(*) into t from S;
  select count(*) into aliveS from SP
  where SP.SID=S.SID;
  deadS:=t-aliveS;
  select count(*) into t from P;
  select count(*) into aliveP from SP
  where SP.PID=P.PID;
  deadP:=t-aliveP;
end;
```

---

---

---

---

---

---

---

# Заключение

67

- ❑ SQL – стандартный язык реляционных баз данных.
- ❑ Составные части SQL:
  - ❑ язык определения данных (DDL)
  - ❑ язык манипулирования данными (DML)
  - ❑ язык определения доступа к данным (DCL)
  - ❑ язык управления транзакциями (TCL)
- ❑ SQL является мощным и избыточным языком запросов, а не языком программирования.

---

---

---

---

---

---

---