

# Язык запросов SQL (продолжение)

Технологии баз данных. Лекция 6

# Структура SQL

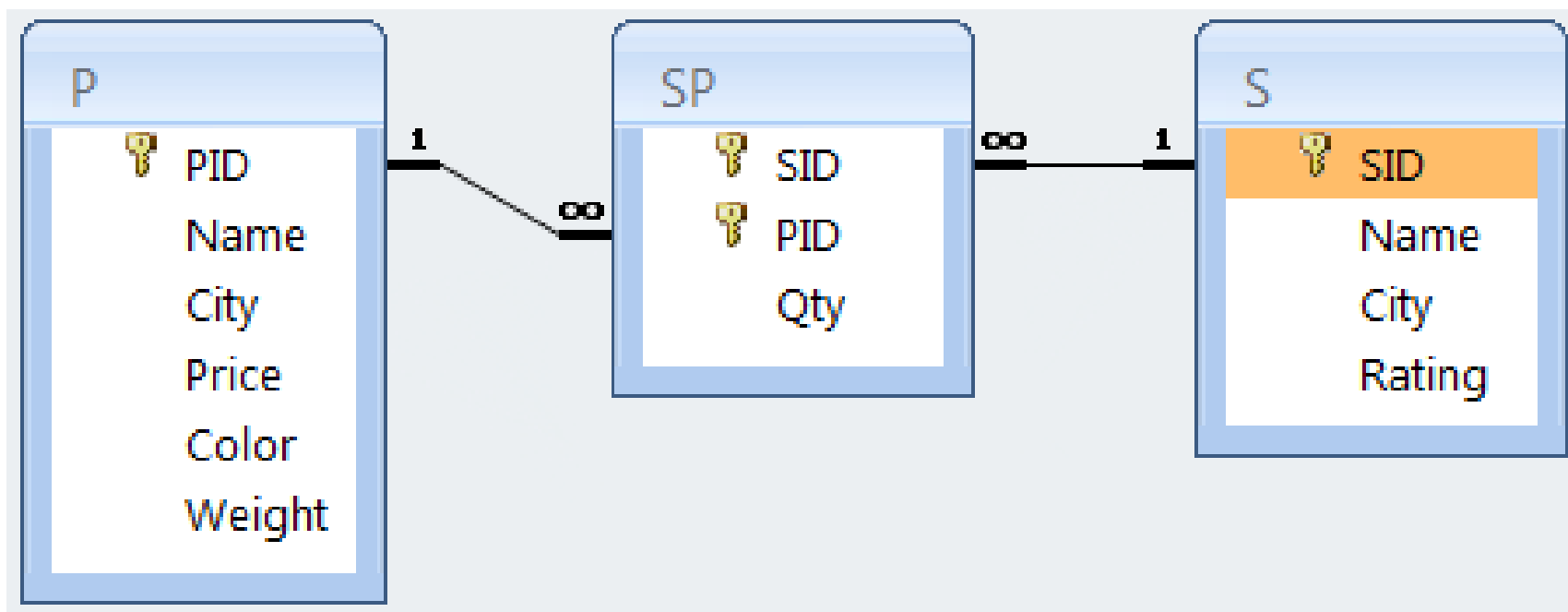
- Язык определения данных (Data Definition Language, DDL)
- **Язык манипулирования данными (Data Manipulation Language, DML)  $\supset$  язык запросов (Data Query Language, DQL)**
- Язык определения доступа к данным (Data Control Language, DCL)
- Язык управления транзакциями (Transaction Control Language,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

# Модельная база данных

Детали

Поставки

Поставщики



# 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');
```

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

## Результат

Name
ИТ-монстр
Разноимпорт

# 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');
```

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

*Результат подзапроса –  
производное отношение:*

SP.SID
S3
S2
S8



```
select distinct S.Name  
from S  
where S.SID in  
({'S3'}, {'S2'}, {'S8'});
```



**Результат**

Name
ИТ-монстр
Разноимпорт

# Выборка с подзапросом

- 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	
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

# Выборка с подзапросом

- 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');
```

*Результат подзапроса вычисляется для каждой строки из S:*

Для 1-й строки:

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

Для 2-й строки:

SP.SID	SP.PID	SP.Qty
S2	P2	30

Для 3-й строки: ...

Для 1-й строки:

```
select distinct S.Name
from S
where exists (
  {}
);
```

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

**Результат**

Name
------

# Выборка с подзапросом

- 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	Саспенс +	S1	P5	10
S2	ИТ-монстр	S3	P2	20
S3	Разноимпорт	S2	P2	30
S5	МакроХард	S7	P1	40
S7	Овощторг	S5	P4	50
S8	Разноимпорт	S8	P2	30

*Результат подзапроса вычисляется для каждой строки из S:*

Для 1-й строки:

SP.SID	SP.PID	SP.Qty

Для 2-й строки:

SP.SID	SP.PID	SP.Qty
S2	P2	30

Для 3-й строки: ...

Для 2-й строки:

```
select distinct S.Name
from S
where exists (
  {'S2', 'P2', '30'}
);
```

**Результат**

Name
ИТ-монстр



# Выборка с подзапросом

- 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	Саспенс +	S1	P5	10
S2	ИТ-монстр	S3	P2	20
S3	Разноимпорт	S2	P2	30
S5	МакроХард	S7	P1	40
S7	Овощторг	S5	P4	50
S8	Разноимпорт	S8	P2	30

*Результат подзапроса вычисляется для каждой строки из S:*

Для 1-й строки:

SP.SID	SP.PID	SP.Qty

Для 2-й строки:

SP.SID	SP.PID	SP.Qty
S2	P2	30

Для 3-й строки: ...

Для 3-й строки:

```
select distinct S.Name
from S
where exists (
  {'S3', 'P2', '20'}
);
```

**Результат**

Name
ИТ-монстр
Разноимпорт

# Выборка с подзапросом

- 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');
```

*Результат подзапроса вычисляется для каждой строки из S:*

Для 1-й строки:

SP.SID	SP.PID	SP.Qty
--------	--------	--------

Для 2-й строки:

SP.SID	SP.PID	SP.Qty
S2	P2	30

Для 3-й строки: ...

Для 4-й и 5-й строк:

```
select distinct S.Name
from S
where exists (
  {}
);
```

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

# Выборка с подзапросом

- 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');
```

*Результат подзапроса вычисляется для каждой строки из S:*

Для 1-й строки:

SP.SID	SP.PID	SP.Qty
--------	--------	--------

Для 2-й строки:

SP.SID	SP.PID	SP.Qty
S2	P2	30

Для 3-й строки: ...

Для 6-й строки:

```
select distinct S.Name
from S
where exists (
  {'S8', 'P2', '30'}
);
```

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

**Результат**

Name
ИТ-монстр
Разноимпорт

# Выборка с подзапросом

- 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	
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

# Выборка с подзапросом

- 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	Саспенс +	S1	P5	10
S2	ИТ-монстр	S3	P2	20
S3	Разноимпорт	S2	P2	30
S5	МакроХард	S7	P1	40
S7	Овощторг	S5	P4	50
S8	Разноимпорт	S8	P2	30

*Результат подзапроса вычисляется для каждой строки из S:*

Для 1-й строки:

SP.PID
P5

Для 2-й строки:

SP.PID
P2

Для 3-й строки: ...

Для 1-й строки:

**select distinct S.Name**  
**from S**  
**where 'P2' = any (**  
**{'P5'}**  
**);**

**Результат**

Name

# Выборка с подзапросом

- 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  | Саспенс +   | S1  | P5  | 10  |
| S2  | ИТ-монстр   | S3  | P2  | 20  |
| S3  | Разноимпорт | S2  | P2  | 30  |
| S5  | МакроХард   | S7  | P1  | 40  |
| S7  | Овощторг    | S5  | P4  | 50  |
| S8  | Разноимпорт | S8  | P2  | 30  |

*Результат подзапроса вычисляется для каждой строки из S:*

Для 1-й строки:

| SP.PID |
|--------|
| P5     |

Для 2-й строки:

| SP.PID |
|--------|
| P2     |

Для 3-й строки: ...

Для 2-й строки:

```
select distinct S.Name
from S
where 'P2' = any (
  {'P2'}
);
```

**Результат**

| Name      |
|-----------|
| ИТ-монстр |

# Выборка с подзапросом

- Получить имена поставщиков, поставляющих деталь P2.

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

| 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 |

# Выборка с подзапросом

- Получить имена поставщиков, поставляющих деталь P2.

```
select distinct S.Name
from S,
(select S.SID as 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  | Саспенс +   | S1  | P5  | 10  |
| S2  | ИТ-монстр   | S3  | P2  | 20  |
| S3  | Разноимпорт | S2  | P2  | 30  |
| S5  | МакроХард   | S7  | P1  | 40  |
| S7  | Овощторг    | S5  | P4  | 50  |
| S8  | Разноимпорт | S8  | P2  | 30  |

Производное отношение  
SIDs\_P2:

| SIDs_P2 |
|---------|
| SID     |
| S2      |
| S3      |
| S8      |



```
select distinct S.Name
from S, SIDs_P2
where S.SID=SIDs_P2.SID;
```



Результат

| Name        |
|-------------|
| ИТ-монстр   |
| Разноимпорт |



# Выборка с подзапросом

- Получить имена поставщиков, поставляющих деталь P2.

```
select distinct S.Name  
from S  
where 0 <  
(  
select count(*) as cnt  
from 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 |

# Выборка с подзапросом

- Получить имена поставщиков, поставляющих деталь P2.

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

*Результат подзапроса вычисляется для каждой строки из S:*

Для 1-й строки:

| cnt |
|-----|
| 0   |

Для 2-й строки:

| cnt |
|-----|
| 1   |

Для 3-й строки: ...

Для 1-й строки:

```
select distinct S.Name
from S
where 0 < ({0});
```

| 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 |
|------|
|------|

# Выборка с подзапросом

- Получить имена поставщиков, поставляющих деталь P2.

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

*Результат подзапроса вычисляется для каждой строки из S:*

Для 1-й строки:

| cnt |
|-----|
| 0   |

Для 2-й строки:

| cnt |
|-----|
| 1   |

Для 3-й строки: ...

Для 2-й строки:

```
select distinct S.Name
from S
where 0 < ({{1}});
```

**SP**

**S**

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

| 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 |

# Выборка с подзапросом

- 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  | Овощторг    | S1  | P5  | 10  |
| S2  | ИТ-монстр   | S3  | P2  | 20  |
| S3  | Разноимпорт | S2  | P2  | 30  |
| S5  | МакроХард   | S7  | P1  | 40  |
| S7  | Овощторг    | S5  | P4  | 50  |
| S8  | Разноимпорт | S8  | P2  | 30  |

## Результат

| Name      |
|-----------|
| Овощторг  |
| МакроХард |

# Выборка с подзапросом

- 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');
```

**P**

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

# Выборка с подзапросом

- 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    | Шепетовка |
| P3  | Болт   | 50    | Челябинск |
| P4  | Уголок | 40    | Одесса    |
| P5  | Шуруп  | 23    | Челябинск |
| P6  | Гайка  | 77    | Москва    |
| P7  | Болт   | 88    | Копейск   |

Результат подзапроса:

| P.Price |
|---------|
| 30      |
| 50      |
| 23      |



```
select distinct P.Name
from P
where P.Price > all (
    {30}, {50}, {23}
)
```



Результат

| Name  |
|-------|
| Брус  |
| Гайка |
| Болт  |

# Выборка с подзапросом

- 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   |             |
|-----|-------------|
| SID | Name        |
| S1  | Овощторг    |
| 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  |

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

## Результат

| Name      |
|-----------|
| Овощторг  |
| МакроХард |

# Выборка с группировкой

- Получить код и общее количество поставки каждой детали.

```
select SP.PID, sum(SP.Qty) as Total  
from SP  
group by SP.PID;
```

## 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    |



# Выборка с группировкой

- Получить код и общее количество поставки каждой детали.

```
select SP.PID, sum(SP.Qty) as Total  
from SP  
group by SP.PID;
```

Создание групп в таблице SP по полю SP.PID:

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

упорядочивание



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

выделение групп  
по полю SP.PID

## 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    |

# Выборка с группировкой

- Получить код и общее количество поставки каждой детали.

```
select SP.PID, sum(SP.Qty) as Total  
from SP  
group by SP.PID;
```

Вычисление **sum**(SP.Qty) для каждой группы:

| SID | PID | Qty |               |
|-----|-----|-----|---------------|
| S1  | P5  | 10  | 10 + 20 = 30  |
| S5  | P5  | 20  |               |
| S3  | P2  | 20  | 20 + 30 = 50  |
| S8  | P2  | 30  |               |
| S2  | P4  | 30  | 30            |
| S7  | P1  | 40  | 40 + 80 = 120 |
| S4  | P1  | 80  |               |
| S5  | P3  | 50  | 50            |



| SID | PID | Total |
|-----|-----|-------|
| n/a | P5  | 30    |
| n/a | P2  | 50    |
| n/a | P4  | 30    |
| n/a | P1  | 120   |
| n/a | P3  | 50    |

## 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    |

# Выборка с группировкой

- Получить код и общее количество поставки каждой детали.

```
select SP.PID, sum(SP.Qty) as Total  
from SP  
group by SP.PID;
```

Отбрасывание «лишних» столбцов:

| SID | PID | Total |
|-----|-----|-------|
| n/a | P5  | 30    |
| n/a | P2  | 50    |
| n/a | P4  | 30    |
| n/a | P1  | 120   |
| n/a | P3  | 50    |

## 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    |

# Выборка с группировкой

- 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  | P5  | 20  |
| S5  | P3  | 50  |
| S8  | P2  | 30  |
| S4  | P1  | 80  |

## Результат

| PID |
|-----|
| P5  |
| P2  |
| P1  |

# Выборка с группировкой

- Получить коды деталей, поставляемых более чем одним поставщиком.

```
select SP.PID  
from SP  
group by SP.PID  
having count(SP.SID)>1;
```

Создание групп в таблице SP по полю SP.PID  
и вычисление в каждой группе count(SP.SID):

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

P5

P2

P4

P1

P3



| SID | PID | count(SP.SID) |
|-----|-----|---------------|
| n/a | P5  | 2             |
| n/a | P2  | 2             |
| n/a | P4  | 1             |
| n/a | P1  | 2             |
| n/a | P3  | 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 |
|-----|
| P5  |
| P2  |
| P1  |

# Выборка с группировкой

- Получить коды деталей, поставляемых более чем одним поставщиком.

```
select SP.PID
from SP
group by SP.PID
having count(SP.SID)>1;
```

Применение условия having:

| SID | PID | count(SP.SID) |
|-----|-----|---------------|
| n/a | P5  | 2             |
| n/a | P2  | 2             |
| n/a | P4  | 1             |
| n/a | P1  | 2             |
| n/a | P3  | 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 |
|-----|
| P5  |
| P2  |
| P1  |

# Выборка с группировкой

- Получить коды деталей, поставляемых более чем одним поставщиком.

```
select SP.PID
from SP
group by SP.PID
having count(SP.SID)>1;
```

Отбрасывание «лишних» столбцов:

| SID | PID | count(SP.SID) |
|-----|-----|---------------|
| n/a | P5  | 2             |
| n/a | P2  | 2             |
| n/a | P1  | 2             |

## SP

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

## Результат

| PID |
|-----|
| P5  |
| P2  |
| P1  |

# Выборка с объединением

- Получить названия деталей, которые либо имеют цену более 100, либо поставляются поставщиками из Челябинска.
  - **select** P.Name  
**from** P  
**where** 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**

| SID | City      |
|-----|-----------|
| S1  | Москва    |
| S2  | Урюпинск  |
| S3  | Челябинск |
| S5  | Одесса    |
| S7  | Челябинск |
| S8  | Челябинск |

**SP**

| SID | PID | Qty |
|-----|-----|-----|
| S1  | P1  | 10  |
| S2  | P1  | 40  |
| S3  | P5  | 20  |
| S7  | P2  | 100 |
| S8  | P2  | 300 |
| 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 |



# Выборка с объединением

- Получить названия деталей, которые либо имеют цену более 100, либо поставляются поставщиками из Челябинска.

- `select P.Name  
from P  
where 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';`

запрос 1

запрос 2

Результат запроса 1:

| Name  |
|-------|
| Болт  |
| Болт  |
| Шуруп |

Результат запроса 2:

| Name  |
|-------|
| Шуруп |
| Гайка |
| Гайка |

**union**

=

Результат

| Name  |
|-------|
| Болт  |
| Шуруп |
| Гайка |

| S | SID | City      |
|---|-----|-----------|
|   | S1  | Москва    |
|   | S2  | Урюпинск  |
|   | S3  | Челябинск |
|   | S5  | Одесса    |
|   | S7  | Челябинск |
|   | S8  | Челябинск |

| SP | SID | PID | Qty |
|----|-----|-----|-----|
|    | S1  | P1  | 10  |
|    | S2  | P1  | 40  |
|    | S3  | P5  | 20  |
|    | S7  | P2  | 100 |
|    | S8  | P2  | 300 |
|    | S5  | P4  | 500 |

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

# Операция объединения UNION

Результат запроса 1:

| Name  |
|-------|
| Болт  |
| Болт  |
| Шуруп |

**union**

Результат запроса 2:

| Name  |
|-------|
| Шуруп |
| Гайка |
| Гайка |

=

Результат

| Name  |
|-------|
| Болт  |
| Шуруп |
| Гайка |

- Результатом объединения будет множество, состоящее **из всех строк, входящих в какое-либо одно или в оба первоначальных отношения.**
- Из результата выборки, использующей оператор UNION, всегда **исключаются избыточные дубликаты.**
- Два исходных отношения должны быть совместимыми по объединению, т.е. иметь **одинаковый набор столбцов.**

# Операция объединения UNION ALL

Результат запроса 1:

| Name  |
|-------|
| Болт  |
| Болт  |
| Шуруп |

**union all**

Результат запроса 2:

| Name  |
|-------|
| Шуруп |
| Гайка |
| Гайка |

=

Результат

| Name  |
|-------|
| Болт  |
| Болт  |
| Шуруп |
| Шуруп |
| Гайка |
| Гайка |

- Оператор UNION ALL позволяет, в отличие от UNION, разрешить выборку повторяющихся значений.

# Выборка с пересечением

- Получить названия деталей, которые имеют цену более 100 и поставляются поставщиками из Челябинска.

- `select P.Name  
from P  
where P.Price>100  
intersect`

запрос 1

- `select P.Name  
from S, P, SP  
where SP.PID=P.PID and SP.SID=S.SID  
and S.City='Челябинск';`

запрос 2

**S**

| SID | City      |
|-----|-----------|
| S1  | Москва    |
| S2  | Урюпинск  |
| S3  | Челябинск |
| S5  | Одесса    |
| S7  | Челябинск |
| S8  | Челябинск |

**SP**

| SID | PID | Qty |
|-----|-----|-----|
| S1  | P1  | 10  |
| S2  | P1  | 40  |
| S3  | P5  | 20  |
| S7  | P2  | 100 |
| S8  | P2  | 300 |
| S5  | P4  | 500 |

**P**

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

Результат запроса 1:

| Name  |
|-------|
| Болт  |
| Болт  |
| Шуруп |

Результат запроса 2:

| Name  |
|-------|
| Шуруп |
| Гайка |
| Гайка |

**intersect**

=

Результат

| Name  |
|-------|
| Шуруп |

# Операция пересечения INTERSECT

Результат запроса 1:

| Name  |
|-------|
| Болт  |
| Болт  |
| Шуруп |

**intersect**

Результат запроса 2:

| Name  |
|-------|
| Шуруп |
| Гайка |
| Гайка |

=

Результат

| Name  |
|-------|
| Шуруп |

- В результирующий набор попадают только те **строки, которые присутствуют в обоих запросах.**
- Из результата выборки **исключаются избыточные дубликаты.**
- Два исходных отношения должны быть совместимыми по объединению, т.е. иметь **одинаковый набор столбцов.**

# Операция пересечения INTERSECT ALL

Результат запроса 1:

| Name  |
|-------|
| Болт  |
| Болт  |
| Шуруп |

**intersect all**

Результат запроса 2:

| Name  |
|-------|
| Шуруп |
| Гайка |
| Гайка |

=

Результат

| Name  |
|-------|
| Шуруп |

- Количество дублированных строк подчиняется следующим правилам ( $n1$  - число дубликатов строк первого запроса,  $n2$  - число дубликатов строк второго запроса):

$$\min(n1, n2)$$

# Выборка с разностью

- Получить названия деталей, которые имеют цену более 100 и поставляются поставщиками НЕ из Челябинска.
  - select** P.Name  
**from** P  
**where** 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**

| SID | City      |
|-----|-----------|
| S1  | Москва    |
| S2  | Урюпинск  |
| S3  | Челябинск |
| S5  | Одесса    |
| S7  | Челябинск |
| S8  | Челябинск |

**SP**

| SID | PID | Qty |
|-----|-----|-----|
| S1  | P1  | 10  |
| S2  | P3  | 40  |
| S3  | P5  | 20  |
| S7  | P2  | 100 |
| S8  | P5  | 300 |
| S5  | P4  | 500 |

**P**

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

**Результат**

| Name |
|------|
| Болт |

# Выборка с разностью

- Получить названия деталей, которые имеют цену более 100 и поставляются поставщиками НЕ из Челябинска.

- `select P.Name  
from P  
where P.Price>100  
except`

запрос 1

- `select P.Name  
from S, P, SP  
where SP.PID=P.PID and SP.SID=S.SID  
and S.City='Челябинск';`

запрос 2

**S**

| SID | City      |
|-----|-----------|
| S1  | Москва    |
| S2  | Урюпинск  |
| S3  | Челябинск |
| S5  | Одесса    |
| S7  | Челябинск |
| S8  | Челябинск |

**SP**

| SID | PID | Qty |
|-----|-----|-----|
| S1  | P1  | 10  |
| S2  | P3  | 40  |
| S3  | P5  | 20  |
| S7  | P2  | 100 |
| S8  | P5  | 300 |
| S5  | P4  | 500 |

**P**

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

Результат запроса 1:

| Name  |
|-------|
| Болт  |
| Болт  |
| Шуруп |

Результат запроса 2:

| Name  |
|-------|
| Шуруп |
| Гайка |
| Гайка |

except

=

Результат

| Name |
|------|
| Болт |



# Операция разности EXCEPT

Результат запроса 1:

| Name  |
|-------|
| Болт  |
| Болт  |
| Шуруп |

**except**

Результат запроса 2:

| Name  |
|-------|
| Шуруп |
| Гайка |
| Гайка |

=

Результат

| Name |
|------|
| Болт |

- В результирующий набор попадают только те **строки первого запроса, которые отсутствуют во втором.**
- Из результата выборки **исключаются избыточные дубликаты.**
- Два исходных отношения должны быть совместимыми по объединению, т.е. иметь **одинаковый набор столбцов.**

# Операция разности EXCEPT ALL

Результат запроса 1:

| Name  |
|-------|
| Болт  |
| Болт  |
| Шуруп |

**except all**

Результат запроса 2:

| Name  |
|-------|
| Шуруп |
| Гайка |
| Гайка |

=

Результат

| Name |
|------|
| Болт |
| Болт |

# Предикат LIKE

expression [ NOT ] LIKE pattern

- Предикат **LIKE** сравнивает строку, указанную в expression, с образцом, который определен в pattern.
- Шаблон может включать в себя следующие спец. символы:

| Символ | Описание                                             |
|--------|------------------------------------------------------|
| %      | Строка любой длины                                   |
| _      | Любой одиночный символ                               |
| []     | Диапазон или последовательность символов             |
| [^]    | Исключающий диапазон или последовательность символов |

# Выборка с предикатом LIKE

- Получить названия деталей, которые поставляются поставщиками из Челябинска.

```
select P.Name  
from S, P, SP  
where SP.PID=P.PID and SP.SID=S.SID  
and S.City LIKE 'Чел%';
```

**S**

| SID | City      |
|-----|-----------|
| S1  | Москва    |
| S2  | Урюпинск  |
| S3  | Челябинск |
| S5  | Одесса    |
| S7  | Челябинск |
| S8  | Челябинск |

**SP**

| SID | PID | Qty |
|-----|-----|-----|
| S1  | P1  | 10  |
| S2  | P3  | 40  |
| S3  | P5  | 20  |
| S7  | P2  | 100 |
| S8  | P5  | 300 |
| S5  | P4  | 500 |

**Результат**

| Name  |
|-------|
| Шуруп |
| Гайка |
| Шуруп |

**P**

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

# 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;
```

# 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.

# 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;
```

Секция объявления  
общих переменных

```
EXEC SQL
```

```
select sum(SP.Qty*P.Price) into :sum
```

```
from P, SP
```

```
where SP.SID = :mySID and SP.PID=P.PID;
```

Встраивание SQL кода

Использование внешней  
переменной

Изменение внешней  
переменной

```
if (strcmp(SQLSTATE, '00000'))
```

```
return sum;
```

```
else
```

```
return -1;
```

Код завершения  
последней команды SQL

```
}
```

# Курсоры

- **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



# Курсоры

```
/* 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;
```

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

```
    /*Обработка :cSID, :cName, :cCity, :cRating */
```

Извлечение записи

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

Заккрытие курсора

# Курсоры

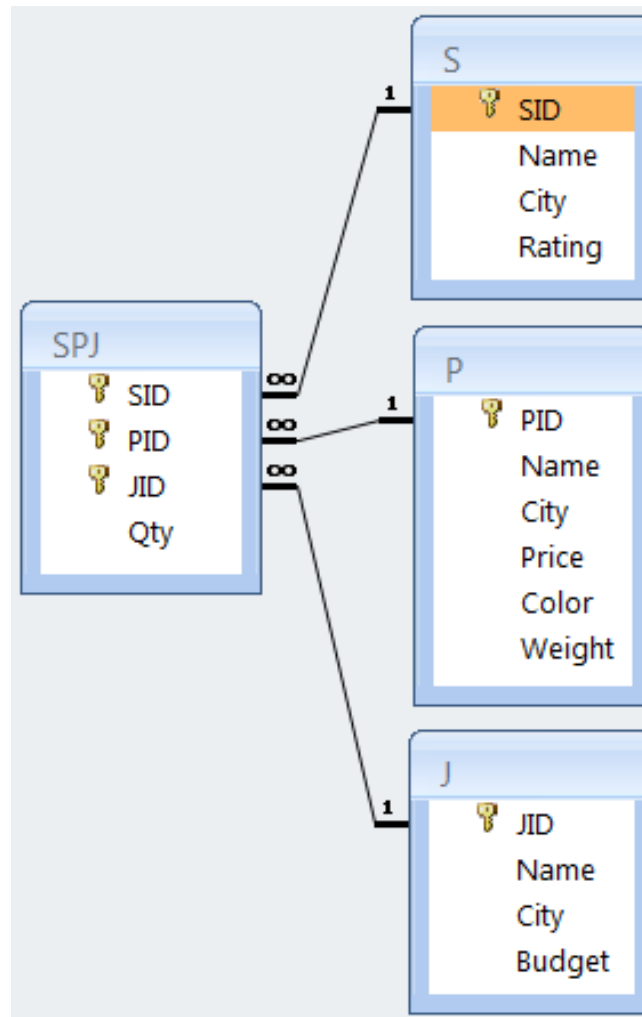
```
/* Удаление и обновление записей базовой таблицы посредством курсора */  
void ProcessS (void)  
{  
EXEC SQL begin declare section;  
    char cSID[4], cName[10], cCity[10]; int cRating;  
    char SQLSTATE[6];  
EXEC SQL end declare section;  
  
EXEC SQL declare myCursor cursor for  
    select * 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;  
        ...  
        EXEC SQL update S set ... where current of myCursor;  
        ...  
    }  
EXEC SQL close myCursor;  
}
```

Удаление записи

Обновление записи

# Модельная база данных

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



# Запрос

- Получить имена поставщиков проекта J1.
- Получить имя поставщика, имя детали, имя проекта и количество деталей для поставок деталей из Челябинска.
- Получить все тройки "код поставщика – код детали – код проекта", где поставщик, деталь и проект размещены в одном городе.

