

# Подзапросы и предикаты

# Проблема

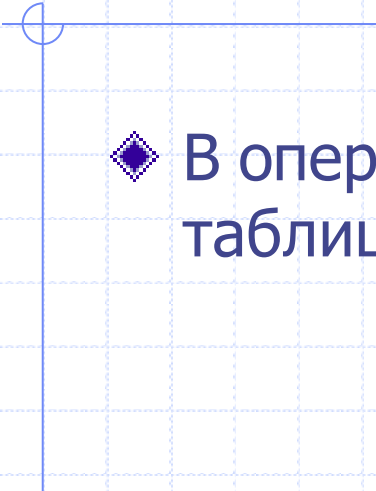
- ◆ Требование реляционной модели – суперпозиция реляционных операций
- ◆ SELECT обеспечивает ее лишь частично

# Решение

- ◆ Раннее решение – использовать представления

```
CREATE [OR ALTER] VIEW <name_view>  
    [ (<nm_col1> [,<nm_col2>, ...])  
AS <select_operator>
```

```
DROP VIEW <name_view>
```

- 
- ◆ В операторе `SELECT` нет различий между таблицами и представлениями

# Пример

```
CREATE VIEW  
EMP_DEPT (EMP_NAME, DEPT_NAME,  
          SALARY, COUNTRY)  
AS  
SELECT FULL_NAME, DEPARTMENT, SALARY, JOB_COUNTRY  
FROM EMPLOYEE A JOIN DEPARTMENT D  
   ON  A.DEPT_NO = D.DEPT_NO
```

---

```
SELECT EMP_NAME, DEPT_NAME, SALARY  
FROM EMP_DEPT  
WHERE COUNTRY = 'USA'
```

# Подзапрос

- ◆ Подзапрос – это оператор `SELECT`, включенный в спецификацию другого оператора `SQL`
- ◆ Если подзапрос ссылается на поля из таблиц в основном запросе, то он называется коррелированным

# Использование подзапросов

- ◆ В операторе INSERT

```
INSERT INTO <таблица> [(<список  
столбцов>)]  
(SELECT ...)
```

# Использование подзапросов

- ◆ для формирования выходного столбца основного оператора SELECT

```
SELECT e.FULL_NAME, e.SALARY,  
(SELECT MAX (m.SALARY)  
     FROM EMPLOYEE m  
     WHERE m.JOB_COUNTRY='USA') AS  
     MAX_SALARY_USA  
FROM EMPLOYEE e  
WHERE e.JOB_COUNTRY='USA'
```



# Использование подзапросов

◆ в вычисляемом столбце

```
SELECT e.FULL_NAME, e.SALARY,  
(SELECT MAX (m.SALARY)  
    FROM EMPLOYEE m  
    WHERE m.JOB_COUNTRY='USA') - e.SALARY  
    AS DIFF_SALARY_USA  
FROM EMPLOYEE e  
WHERE e.JOB_COUNTRY='USA'
```

# Использование подзапросов

```
SELECT FULL_NAME, SALARY,  
(SELECT CURRENCY FROM COUNTRY c  
  WHERE  
c.COUNTRY=e.JOB_COUNTRY) AS CURRENCY  
FROM EMPLOYEE e
```

Такой подзапрос называется коррелированным.  
Это неявная форма операции соединения

# Использование подзапросов

- ◆ в предложении FROM (начиная с версии Firebird 2.0)
- ◆ в этом случае подзапрос выступает в роли «производной таблицы» (*derived table*)

# Derived table

(select-запрос)

[[AS] алиас производной таблицы]

[( <псевдонимы полей производной таблицы> )]

# Derived table

```
SELECT P.PROJ_NAME, T.DEPT_NAME  
FROM  
PROJECT P LEFT JOIN
```

```
(SELECT PD.PROJ_ID, D.DEPARTMENT  
FROM PROJ_DEPT_BUDGET PD  
JOIN DEPARTMENT D  
ON (PD.DEPT_NO=D.DEPT_NO)  
WHERE PD.FISCAL_YEAR=1995)
```

```
AS T ( PROJ_ID, DEPT_NAME)  
ON (P.PROJ_ID=T.PROJ_ID)
```

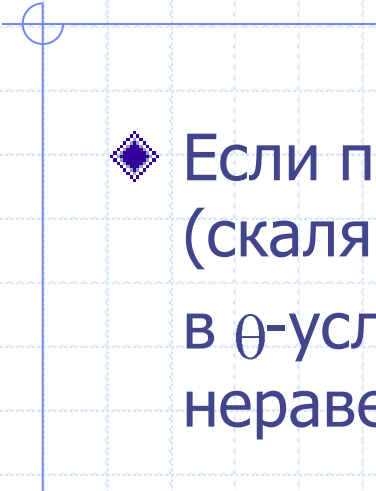
# Альтернатива - представление

```
CREATE VIEW DT ( PROJ_ID, DEPT_NAME)
AS
SELECT PD.PROJ_ID, D.DEPARTMENT
FROM PROJ_DEPT_BUDGET PD
JOIN DEPARTMENT D
ON (PD.DEPT_NO=D.DEPT_NO)
WHERE PD.FISCAL_YEAR=1995;
```

```
SELECT P.PROJ_NAME, T.DEPT_NAME
FROM PROJECT P LEFT JOIN DT ON
(P.PROJ_ID=DT.PROJ_ID)
```

# Использование подзапросов

- ◆ для получения значений или условий, используемых в предикатах поиска предложения WHERE операторов SELECT, UPDATE и DELETE, а также в предложении HAVING для группирующего запроса

- 
- ◆ Если подзапрос возвращает одно значение (скалярный запрос), его можно использовать в  $\theta$ -условии для сравнения на равенство или неравенство



```
SELECT FULL_NAME  
FROM EMPLOYEE a  
WHERE  
a.SALARY >  
(SELECT AVG(b.SALARY) FROM EMPLOYEE b);
```

- ◆ Если подзапрос возвращает таблицу, содержащую один столбец и произвольное количество строк, для использования его в подзапросе используются предикаты
- ◆ В этом случае возвращаемый столбец рассматривается как множество

# Предикат IN

```
SELECT FULL_NAME  
FROM EMPLOYEE  
WHERE DEPT_NO  
      IN  
      (SELECT DEPT_NO  
        FROM DEPARTMENT  
        WHERE MNGR_NO IS NULL)
```

# Предикаты ALL, SOME и ANY

```
SELECT PROJ_ID, DEPT_NO, FISCAL_YEAR
FROM PROJ_DEPT_BUDGET
WHERE
    PROJECTED_BUDGET >
    ALL
    (SELECT BUDGET
     FROM DEPARTMENT)
```

```
SELECT e.EMP_NO, e.FULL_NAME, e.HIRE_DATE  
FROM EMPLOYEE e  
WHERE e.HIRE_DATE+365>  
SOME
```

```
(SELECT sh.CHANGE_DATE  
FROM SALARY_HISTORY sh  
WHERE sh.EMP_NO = e.EMP_NO)
```

-- *ВОЗМОЖНО ИСПОЛЬЗОВАНИЕ ANY*

# Предикат SINGULAR (NOT SINGULAR)

- ◆ проверяет, возвращает ли подзапрос в точности один кортеж

Если возвращается NULL или более одного кортежа, то SINGULAR возвращает ложь

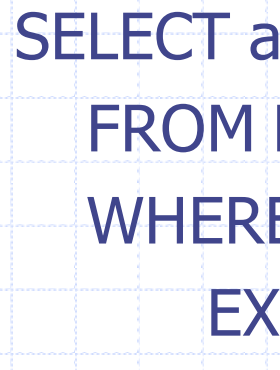
Значение предиката не зависит от количества столбцов в подзапросе и от значений в них

```
SELECT FULL_NAME
FROM EMPLOYEE e
WHERE
SINGULAR (SELECT *
FROM SALES s
WHERE s.SALES_REP=e.EMP_NO);
```

# Предикат EXISTS (NOT EXISTS)

- ◆ определяет, существует ли (или нет), по крайней мере, один кортеж в выходном результате подзапроса
- ◆ Значение предиката не зависит от количества столбцов в подзапросе





```
SELECT a.FULL_NAME
FROM EMPLOYEE a
WHERE
    EXISTS (SELECT 1
            FROM PROJECT p
            WHERE
                p.TEAM_LEADER = a.EMP_NO)
```

## Подзапрос в условии на группу

```
select d.location  
from department d  
group by location  
having count(*) >=
```

```
All (select count(*)  
from department  
group by location )
```

# Общие табличные выражения (CTE)

◆ Более мощный вариант производных таблиц  
WITH **EMP\_DEPT** (EMP\_NAME, DEPT\_NAME, SALARY,  
COUNTRY)

AS

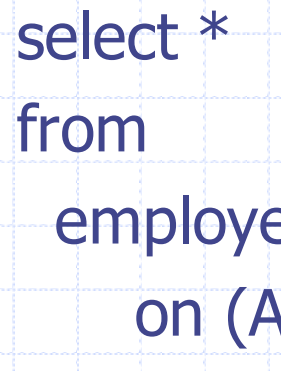
```
(SELECT FULL_NAME, DEPARTMENT, SALARY, JOB_COUNTRY  
FROM EMPLOYEE A JOIN DEPARTMENT D  
ON A.DEPT_NO = D.DEPT_NO)
```

```
SELECT EMP_NAME, DEPT_NAME, SALARY  
FROM EMP_DEPT  
WHERE COUNTRY = 'USA'
```

# План выполнения запроса

- ◆ **План выполнения запроса** — последовательность операций, необходимых для получения результата SQL-запроса в реляционной СУБД

- ◆ План в целом разделяется на две стадии:
  - Выборка результатов
  - Сортировка и группировка, выполнение агрегаций
  
- ◆ *Сортировка и группировка* — это опциональная стадия, которая выполняется, если не найдено путей доступа для получения результата в запрошенном порядке.



```
select *  
from  
    employee A join department B  
        on (A.dept_no =B.dept_no)  
where job_country ='USA'
```



PLAN JOIN (A NATURAL, B INDEX (RDB\$PRIMARY5))

SQL Editor : 3 : class.mmcs.sfedu.ru:/fbdata/employee.fdb (SQL Dialect 3)

SQL Editor | class.mmcs.sfedu.ru:/fbdata/employee.fdb | [Navigation icons]

Edit: 3 | Results | History | Plan Analyzer | Performance Analysis | Logs | Find query

1. Graphical summary | 2. Additional

Legend: [Red] Non-Indexed Reads | [Blue] Indexed Reads | [Green] Updates | [Dark Red] Deletes | [Teal] Inserts | [Yellow] Total Records

Table	Non-Indexed Reads	Indexed Reads
EMPLOYEE	42	0
DEPARTMENT	0	33

Messages | Query columns

Plan  
PLAN JOIN (A NATURAL, B INDEX (RDB\$PRIMARY5))

Adapted Plan  
PLAN JOIN (A NATURAL, B INDEX (INTEG\_16))

----- Performance info -----  
Prepare time = 687ms  
Execute time = 62ms  
Avg fetch time = 3,88 ms  
Current memory = 9 682 752  
Max memory = 9 962 176  
Memory buffers = 2 048  
Reads from disk to cache = 2  
Writes from cache to disk = 8  
Fetches from cache = 259



# Включение плана в запрос

```
SELECT [DISTINCT | ALL]
[FIRST <record_number> }|SKIP <record_number> ]
<select_list>
FROM <reference_expression_list>
[ WHERE <search condition> ]
[ GROUP BY <group_value_list>
[ HAVING <group_condition> ] ]
[UNION <select operator>]
[ PLAN <plan_item_list> ]
```

PLAN <выражение>

<выражение> ::= [JOIN | [SORT] [MERGE]]  
(<элемент> | <выражение>  
[, <элемент> | <выражение> ...])

<элемент> ::= {таблица | псевдоним}  
{NATURAL  
| INDEX (индекс [, индекс ...])  
| ORDER индекс [INDEX (индекс [, индекс...])]}

# Построитель запросов

The screenshot displays the 'SQL builder' application window. The title bar shows the application name and standard window controls. The address bar indicates the data source: 'class.mmcs.sfedu.ru:/fbdata/employee.fdb'. The toolbar contains icons for execution, undo, redo, and other functions. The 'Autolink tables' checkbox is checked.

The main workspace is divided into several sections:

- Table Selection:** Two panes are visible. The left pane shows the 'DEPARTMENT' table with selected fields: 'DEPARTMENT (String)' and 'LOCATION (String)'. The right pane shows the 'EMPLOYEE' table with selected fields: 'SALARY (BCD)' and 'FULL\_NAME (String)'. A line connects the 'LOCATION' field in the DEPARTMENT pane to the 'DEPT\_NO' field in the EMPLOYEE pane.
- Table List:** A vertical list on the right side shows available tables: COUNTRY, CUSTOMER, DEPARTMENT, EMPLOYEE, EMPLOYEE\_PROJECT, IBE\$SCRIPTS, IBE\$VERSION\_HISTORY, JOB, PROJECT, PROJ\_DEPT\_BUDGET, SALARY\_HISTORY, SALES, SALES\_ARH, and PHONE\_LIST.
- Criteria:** At the bottom, a section titled 'Criteria' is active. It shows a radio button for 'All of following are met' and a single criterion: '1. EMPLOYEE.JOB\_COUNTRY = 'USA''.

```
select
    department.department,
    department.location,
    employee.full_name,
    employee.salary
from employee
    inner join department on (employee.dept_no =
department.dept_no)
where
    (employee.job_country = 'USA')
```