# Лабораторная работа №3. Оператор JOIN

## Краткие теоретические данные.

Join – оператор объединения. JOIN в SQL используется при построении select выражений. Инструкция Join позволяет объединить колонки из нескольких таблиц в одну. Объединение происходит временное и целостность таблиц не нарушается. Существует три типа join-выражений:

inner join;

outer join

---left,

---right

---full

cross join;

**Внутреннее объединение INNER JOIN**

Внутреннее объединение INNER JOIN (синоним JOIN, слово INNER можно опустить).

Выбираются только совпадающие данные из объединяемых таблиц.

Синтаксис

*SELECT <список выбираемых столбцов>*

*FROM <Table A>*

*INNER JOIN <Table B> ON <условие объединения>*

**Пример 3.1**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Пусть даны:*  *- таблицы t1, содержащая данные о id студента, фамилию, средний балл, id группы, в которой студент учится;*  *- таблицы t2, содержащая данные о id группы, названии группы, факультете*   |  |  |  |  |  | | --- | --- | --- | --- | --- | | t1 | | | |  | | Id | group\_id | Faculty | Name | Exm | | 1 | 1 | ФПК | Ахрем |  | | 2 | 2 | ФПК | Петров |  | | 3 | 3 | ФПИ | Иванов |  |  |  |  |  | | --- | --- | --- | | **Id** | **Name\_gr** | **Faculty** | | 1 | по1701 | ФПК | | 2 | по1705 | ФПК | | 3 | по1708 | ФПИ | | 4 | по1709 | ФПИ |   Необходимо сформировать запрос, результатом которого будет выборка c информацией о студентах, факультетах, группах, оценках  *SELECT \* FROM tbl2*  *JOIN tbl1*  *ON tbl1.id\_gr = tbl2.id*  2) необходимо вывести информацию о студентах учащихся в группе по1701  *SELECT \* from tbl1*  *JOIN tbl1*  *ON tbl1.id\_gr = tbl2.id*  *WHERE tbl2.name\_gr like 'по1701'* |

**LEFT JOIN**

Внутренний SELECT может возвращать не одно а множество значений одного поля. В таких случаях необходимо использовать оператор *IN.*

Выбираются все столбцы левой таблицы, столбцы удовлетворяющие условию объединения из правой таблицы. Если не найдено соответствия в правой таблице, поля итоговой таблицы заполняются NULL.

Синтаксис

SELECT <список выбираемых столбцов>

FROM <Table A>

LEFT JOIN <Table B> ON <условие объединения>

**Пример 3.2**

|  |
| --- |
| На основании таблиц примера 3.1 вывести список всех групп с указанием студентов, которые там учатся  *SELECT \* from tbl2*  *LEFT JOIN tbl1*  *on tbl2.id = tbl1.id\_gr* |

**RIGHT JOIN**

Выбираются все столбцы правой таблицы, столбцы удовлетворяющие условию объединения из левой таблицы. Если не найдено соответствия в левой таблице, поля итоговой таблицы заполняются NULL.

*Синтаксис*

*SELECT <список выбираемых столбцов>*

*FROM <Table A>*

*RIFHT JOIN <Table B> ON <условие объединения>*

**Пример 3.3:**

|  |
| --- |
| На основании таблиц t1 и t2 (см. пример 3.1) необходимо вывести список студентов с указанием факультета и номера группы:  *SELECT \* from tbl2*  *RIGHT JOIN tbl2*  *on tbl2.id = tbl1.id\_gr* |

**FULL JOIN**

Возвращает все объединение левого и правого объединения)

*Синтаксис*

*SELECT <список выбираемых столбцов>*

*FROM <Table A>*

*full JOIN <Table B>*

*on tbl2.f1 = tbl1.id (условие объединения)*

**Пример 3.4:**

|  |
| --- |
| На основании таблиц t1 и t2 (см. пример 3.1) необходимо вывести список студентов с указанием факультета и номера группы:  *SELECT \* from tbl1 (что)*  *full JOIN tbl2 (с чем)*  *tbl2.f1 = tbl1.id (условие объединения)* |

**CROSS JOIN**

Возвращает объединение каждой строки одного отношения с каждой строкой второго (Декартово произведение)

*Синтаксис*

*SELECT <список выбираемых столбцов>*

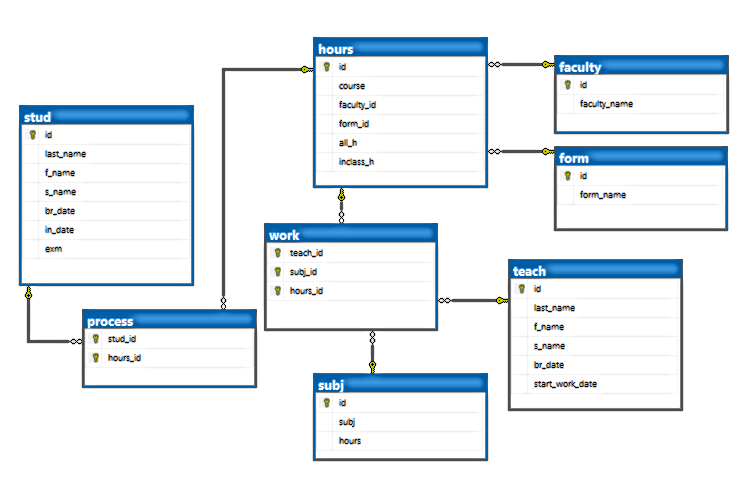
*FROM <Table A>*

*full JOIN <Table B>*

**Пример 3.5:**

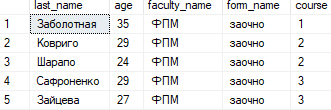
|  |
| --- |
| *SELECT \* from tbl1*  *CROSS JOIN tbl2* |

## Схема БД UNIVER2



## Задание к лабораторной работе №3

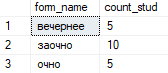
1. ***Вывести информацию студентах (факультет, курс) заочной формы обучения младше 37 лет***
2. *SELECT stud.last\_name, (YEAR(getdate()) - YEAR(stud.br\_date)) AS age, faculty.faculty\_name,form.form\_name, hours.course from stud*
3. *inner JOIN process ON stud.id=process.stud\_id*
4. *inner JOIN hours ON hours.id=process.hours\_id*
5. *inner JOIN form ON hours.form\_id=form.id*
6. *inner JOIN faculty ON hours.faculty\_id=faculty.id*
7. *WHERE (YEAR(getdate()) - YEAR(stud.br\_date)) < 37*
8. *AND form.form\_name = 'заочно'*



1. ***Определить количество студентов на каждом факультете***
2. *SELECT faculty.faculty\_name, count(stud.id) AS count\_stud*
3. *FROM stud*
4. *inner JOIN process ON stud.id=process.stud\_id*
5. *inner JOIN hours ON hours.id=process.hours\_id*
6. *inner JOIN form ON hours.form\_id=form.id*
7. *inner JOIN faculty ON hours.faculty\_id=faculty.id*
8. *GROUP BY faculty.faculty\_name*



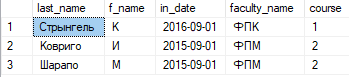
1. ***Определить количество студентов, учащихся на каждой форме обучения***
2. *SELECT form.form\_name, COUNT(stud.id) AS count\_stud*
3. *FROM stud*
4. *inner JOIN process ON stud.id=process.stud\_id*
5. *inner JOIN hours ON hours.id=process.hours\_id*
6. *inner JOIN form ON hours.form\_id=form.id*
7. *inner JOIN faculty ON hours.faculty\_id=faculty.id*
8. *GROUP BY form.form\_name*



1. ***Определить средний возраст студентов для каждого факультета на конец года (лет)***
2. *SELECT faculty.faculty\_name,*
3. *AVG(YEAR('2021-12-31') - YEAR(stud.br\_date)) AS avg\_age*
4. *FROM faculty*
5. *INNER JOIN hours ON hours.faculty\_id = faculty.id*
6. *INNER JOIN process ON process.hours\_id = hours.id*
7. *INNER JOIN stud ON stud.id = process.stud\_id*
8. *GROUP BY faculty.faculty\_name*



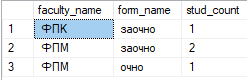
1. ***Найти всех студентов-иностранцев (без отчества), вывести сведения о них (дата поступления, факультет, курс, форма обучения)***
2. *SELECT stud.last\_name, stud.f\_name, stud.in\_date,*
3. *faculty.faculty\_name, hours.course*
4. *FROM stud*
5. *JOIN process on stud.id = process.stud\_id and stud.s\_name is null*
6. *JOIN hours on hours.id = process.hours\_id*
7. *JOIN faculty on faculty.id = hours.faculty\_id*
8. *JOIN form on hours.form\_id=form.id*



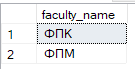
1. ***Определить, на какой факультет поступило наибольшее количество человек в 2015 году***
2. *select top 1 faculty\_name,*
3. *count(stud.id) as stud\_count from faculty*
4. *inner JOIN hours on hours.faculty\_id=faculty.id*
5. *inner JOIN process on hours.id=process.hours\_id*
6. *inner JOIN stud on stud.id=process.stud\_id*
7. *where YEAR(stud.in\_date)=2015*
8. *group by faculty\_name*



1. ***Определить число студентов, поступивших на каждый факультет и форму обучения в 2014 году***
2. ***select faculty\_name, form.form\_name,***
3. ***count(stud.id) as stud\_count from faculty***
4. ***inner JOIN hours on hours.faculty\_id=faculty.id***
5. ***inner JOIN process on hours.id=process.hours\_id***
6. ***inner JOIN stud on stud.id=process.stud\_id***
7. ***inner JOIN form on hours.form\_id=form.id***
8. ***where YEAR(stud.in\_date)=2014***



1. ***Вывести список всех факультетов, на которых доступна заочная форма обучения.***
2. select DISTINCT faculty.faculty\_name from hours
3. inner JOIN faculty on hours.faculty\_id = faculty.id
4. inner JOIN form on hours.form\_id = form.id and form\_name = 'заочно'



1. ***Вывести список факультетов, для каждого факультета вывести список форм обучения и номера курсов, (\*)***

select distinct faculty\_name, form.form\_name, hours.course

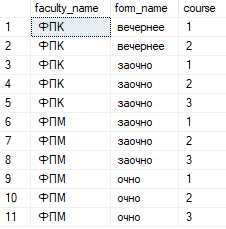
from faculty

inner JOIN hours on hours.faculty\_id=faculty.id

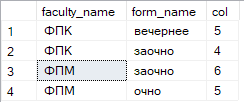
inner JOIN process on hours.id=process.hours\_id

inner JOIN stud on stud.id=process.stud\_id

inner JOIN form on hours.form\_id=form.id



1. ***Вывести список факультетов, с указанием числа студентов обучающихся на каждой форме обучения***
2. select faculty.faculty\_name, form.form\_name, count(stud.last\_name) as col from stud
3. inner JOIN process on stud.id = process.stud\_id
4. inner JOIN hours on hours.id = process.hours\_id
5. inner JOIN faculty on faculty.id = hours.faculty\_id
6. inner JOIN form on form.id = hours.form\_id
7. GROUP BY faculty.faculty\_name,form.form\_name



1. ***Вывести число всех студентов первого и третьего курсов для всех факультетов и форм***

select distinct faculty\_name, form.form\_name, hours.course

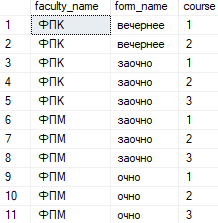
from faculty

inner JOIN hours on hours.faculty\_id=faculty.id

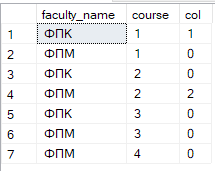
inner JOIN process on hours.id=process.hours\_id

inner JOIN stud on stud.id=process.stud\_id

inner JOIN form on hours.form\_id=form.id



1. ***Для всех факультетов и курсов найти число иностранцев, которое на них учится(\*)***
2. select faculty.faculty\_name,hours.course, count(stud.last\_name) as col from stud
3. left JOIN process on stud.id = process.stud\_id and stud.s\_name is null
4. right JOIN hours on hours.id = process.hours\_id
5. right JOIN faculty on faculty.id = hours.faculty\_id
6. GROUP BY faculty.faculty\_name, hours.course



1. ***Для каждого курса с учетом факультета найти количество студентов, средний балл которых больше 7,5(\*)***

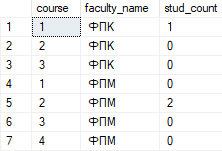
select hours.course, faculty.faculty\_name, count(stud.id) as stud\_count from stud

left JOIN process on stud.id = process.stud\_id and stud.s\_name is null

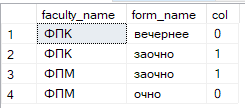
right JOIN hours on hours.id = process.hours\_id

right JOIN faculty on faculty.id = hours.faculty\_id

GROUP BY hours.course, faculty.faculty\_name



1. ***Вывести список всех факультетов с учетом формы с указанием числа студентов старше 45(\*)***
2. select faculty.faculty\_name, form.form\_name, count(stud.last\_name) as col from stud
3. left JOIN process on stud.id = process.stud\_id and (year(getdate()) - year(stud.br\_date ) > 45)
4. RIGHT JOIN hours on hours.id = process.hours\_id
5. right JOIN faculty on faculty.id = hours.faculty\_id
6. right JOIN form on form.id = hours.form\_id
7. GROUP BY faculty.faculty\_name,form.form\_name



1. ***Вывести информацию для каждого факультета, формы, курса о количестве студентов младше 27(\*)***

select faculty.faculty\_name, form.form\_name, hours.course, count(stud.last\_name) as stud\_count from stud

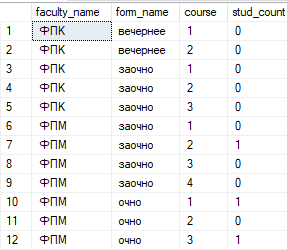
left JOIN process on stud.id = process.stud\_id and DATEDIFF(YEAR,stud.br\_date,GETDATE())<27

right JOIN hours on hours.id = process.hours\_id

right JOIN faculty on faculty.id = hours.faculty\_id

right JOIN form on hours.form\_id=form.id

GROUP BY faculty.faculty\_name, form.form\_name,hours.course



1. ***Определить сколько студентов с фамилиями на C на каждом факультете (\*)***

***\* - left/right***

1. select faculty.faculty\_name, count(stud.last\_name) as col from stud
2. left JOIN process on stud.id = process.stud\_id and stud.last\_name like 'С%'
3. left JOIN hours on hours.id = process.hours\_id
4. right JOIN faculty on faculty.id = hours.faculty\_id
5. GROUP BY faculty.faculty\_name

