



SQL Basic

Lecturer: Nataliya Bogushevskaya, Kyiv

E-mail:
Nataliya_Bogushevskaya@epam.com

Agenda

- Понятие соединения (JOIN)
- CROSS JOIN
- INNER JOIN
- LEFT and RIGHT OUTER JOIN
- FULL JOIN



Join

Соединения

Join

Соединение – это процесс, когда две или более таблицы объединяются в одну.

FROM имя_таблицы_1 {вид соединения}

JOIN имя_таблицы_2

ON условие_соединения

Join

CROSS JOIN

- перекрёстное соединение

INNER JOIN

- внутреннее соединение, используется по умолчанию.

LEFT JOIN
[OUTER]

- левое внешнее соединение

RIGHT JOIN
[OUTER]

- правое внешнее соединение

FULL JOIN
[OUTER]

- полное внешнее соединение

CROSS JOIN

- По умолчанию объединённая таблица представляет собой перекрёстное соединение (CROSS JOIN), называемое также декартовым произведением (Cartesian product).

FROM

имя_таблицы_1 **CROSS JOIN** имя_таблицы_2

CROSS JOIN

select * from dep

Dep_ID	Dep_Name
10	Accounting
20	IT
30	Useless
40	Security

select * from emp

Last_Name	Dep_ID
King	10
Brooks	10
Bush	20
Smith	30
Baker	

select
e.dep_id
,e.last_name
,d.dep_id
,d.dep_name
from emp e **cross join** dep d

	Accounting 10	IT 20	Useless 30	Security 40
King 10				
Brooks 10				
Bush 20				
Smith 30				
Baker NULL				

Пример

--Cross join - соединение всех со всеми
select e.LoginID, e.JobTitle, e.BusinessEntityID
 ,dh.BusinessEntityID, dh.DepartmentID
from [HumanResources].[Employee] e **cross join**
 [HumanResources].[EmployeeDepartmentHistory] dh

LoginID	Job Title	BusinessEntityID	BusinessEntityID	DepartmentID
adventure-works\ken0	Chief Executive Officer	1	1	16
adventure-works\ken0	Chief Executive Officer	1	2	1
adventure-works\ken0	Chief Executive Officer	1	3	1
adventure-works\ken0	Chief Executive Officer	1	4	1
adventure-works\ken0	Chief Executive Officer	1	4	2

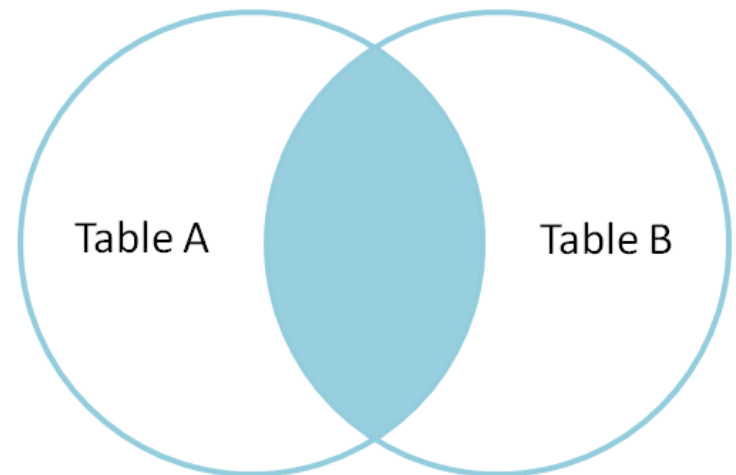
INNER JOIN

При внутреннем естественном соединении группируются только те строки, значения которых по соединяемым столбцам совпадают.

FROM

имя_таблицы_1 **INNER JOIN** имя_таблицы_2

ON условие_соединения



INNER JOIN

select * from dep

Dep_ID	Dep_Name
10	Accounting
20	IT
30	Useless
40	Security

select * from emp

Last_Name	Dep_ID
King	10
Brooks	10
Bush	20
Smith	30
Baker	

```
select
e.dep_id
,e.last_name
,d.dep_id
,d.dep_name
from emp e inner join dep d
on e.dep_id = d.dep_id
```

	Accounting 10	IT 20	Useless 30	Security 40
King 10				
Brooks 10				
Bush 20				
Smith 30				
Baker NULL				

Пример

--inner join - внутреннее соединение

```
select e.LoginID, e.JobTitle, e.BusinessEntityID
      ,dh.BusinessEntityID, dh.DepartmentID
from [HumanResources].[Employee] e inner join
     [HumanResources].[EmployeeDepartmentHistory] dh
on e.BusinessEntityID = dh.BusinessEntityID
```

--неявный inner join

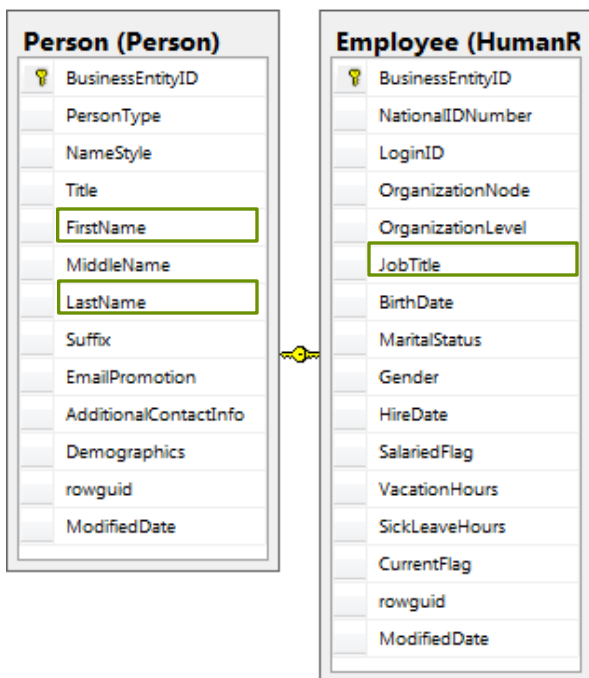
```
select e.LoginID, e.JobTitle, e.BusinessEntityID
      ,dh.BusinessEntityID, dh.DepartmentID
from [HumanResources].[Employee] e ,
     [HumanResources].[EmployeeDepartmentHistory] dh
where e.BusinessEntityID = dh.BusinessEntityID
```

LoginID	JobTitle	BusinessEntityID	BusinessEntityID	DepartmentID
adventure-works\ken0	Chief Executive Officer	1	1	16
adventure-works\teri0	Vice President of Engineering	2	2	1
adventure-works\roberto0	Engineering Manager	3	3	1
adventure-works\rob0	Senior Tool Designer	4	4	1

Пример

Вывести

- фамилию и имя сотрудника,
- должность, которую он занимает



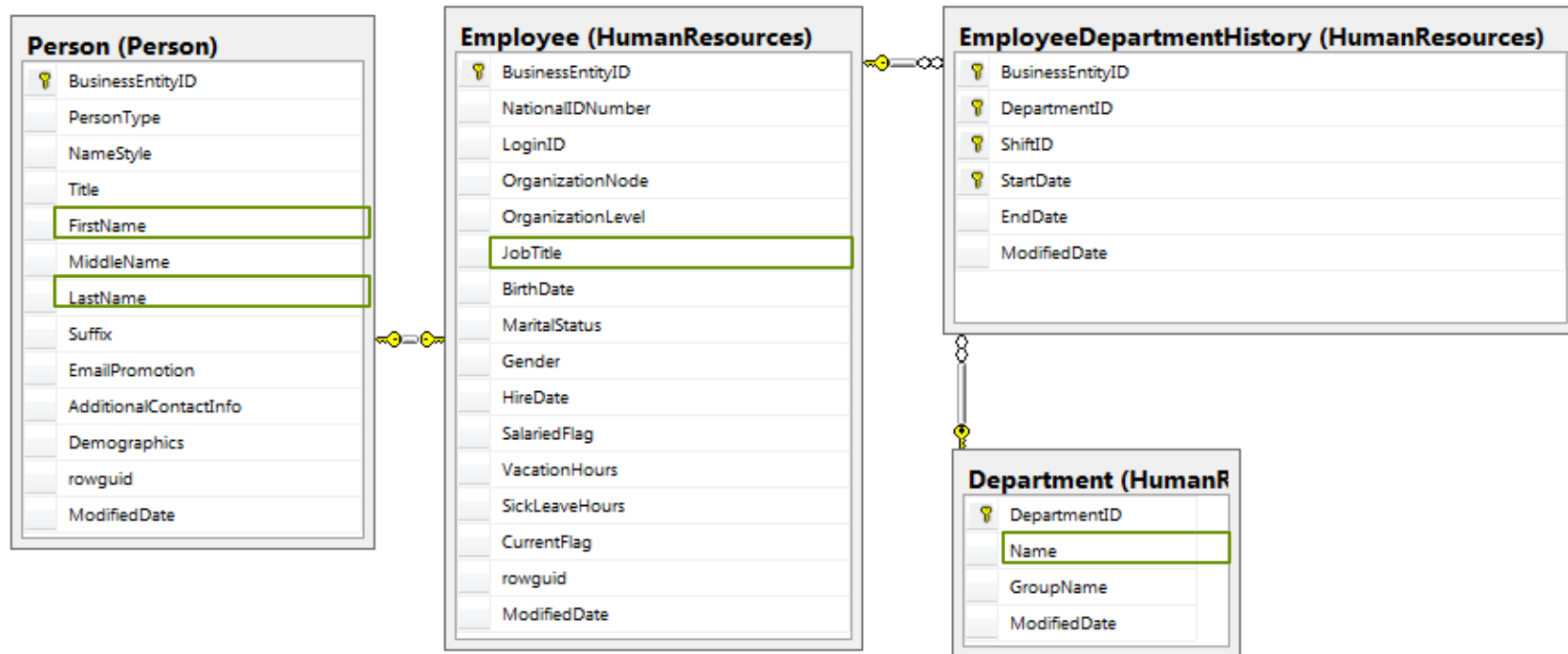
```
select p.LastName, p.FirstName, e.JobTitle  
from HumanResources.Employee e  
inner join Person.Person p  
on e.BusinessEntityID = p.BusinessEntityID  
order by p.LastName
```

Last Name	First Name	Job Title
Abbas	Syed	Pacific Sales Manager
Abercrombie	Kim	Production Technician - WC60
Abolrous	Hazem	Quality Assurance Manager
Ackerman	Pilar	Shipping and Receiving Supervisor
Adams	Jay	Production Technician - WC60

Задания

1. Вывести

- фамилию и имя сотрудника,
- должность,
- название департамента, в котором он работает



- Подсчитать число сотрудников в каждом департаменте, упорядочить по убыванию количества сотрудников.
- Найти количество сотрудников каждой должности в каждом департаменте

Внешнее соединение

Dep_ID	Dep_Name
10	Accounting
20	IT
30	Useless
40	Security

Last_Name	Dep_ID
King	10
Brooks	10
Bush	20
Smith	30
Baker	

INNER JOIN

Dep_ID	Dep_Name	Last_Name
10	Accounting	King
10	Accounting	Brooks
20	IT	Bush
30	Useless	Smith

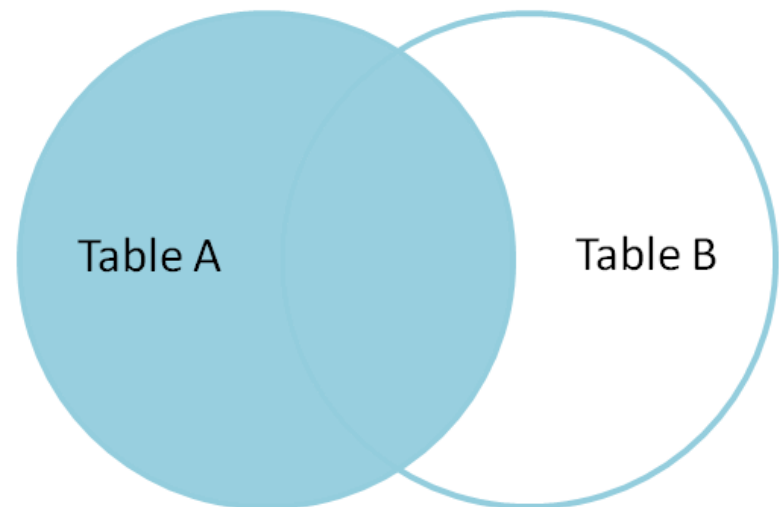
LEFT JOIN

При внешнем левом соединении в результирующий набор будут выбраны все строки из левой таблицы

FROM

имя_таблицы_1 **LEFT JOIN** имя_таблицы_2

ON условие_соединения



LEFT JOIN

select * from dep

Dep_ID	Dep_Name
10	Accounting
20	IT
30	Useless
40	Security

select * from emp

Last_Name	Dep_ID
King	10
Brooks	10
Bush	20
Smith	30
Baker	

```
select
e.dep_id
,e.last_name
,d.dep_id
,d.dep_name
from emp e left join dep d
on e.dep_id = d.dep_id
```

	Accounting 10	IT 20	Useless 30	Security 40
King 10				
Brooks 10				
Bush 20				
Smith 30				
Baker NULL				

Пример

-- Число сотрудников в каждом департаменте
select d.Name, count(e.DepartmentID) as qty
from HumanResources.Department d **left outer join**
HumanResources.EmployeeDepartmentHistory e
on d.DepartmentID = e.DepartmentID
group by d.DepartmentID, d.Name
order by qty ;

Таблица слева от оператора join
берётся вся

RIGHT JOIN

При внешнем левом соединении в результирующий набор будут выбраны все строки из правой таблицы

FROM

имя_таблицы_1 **RIGHT JOIN** имя_таблицы_2

ON условие_соединения

RIGHT JOIN

select * from dep

Dep_ID	Dep_Name
10	Accounting
20	IT
30	Useless
40	Security

select * from emp

Last_Name	Dep_ID
King	10
Brooks	10
Bush	20
Smith	30
Baker	

select
e.dep_id
,e.last_name
,d.dep_id
,d.dep_name
from dep d **right join** emp e
on e.dep_id = d.dep_id

	Accounting 10	IT 20	Useless 30	Security 40
King 10				
Brooks 10				
Bush 20				
Smith 30				
Baker NULL				

Пример

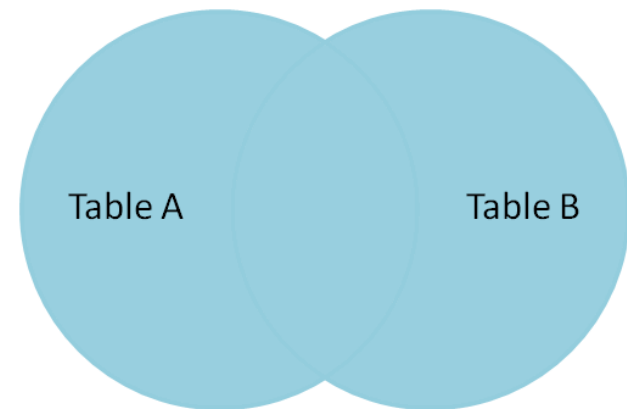
-- Число сотрудников в каждом департаменте
select d.Name, count(e.DepartmentID) as qty
from HumanResources.EmployeeDepartmentHistory e
right outer join HumanResources.Department d
on d.DepartmentID = e.DepartmentID
group by d.DepartmentID, d.Name
order by qty ;

Таблица справа от оператора
join берётся вся

FULL JOIN

При полном внешнем соединении в результирующий набор будут выбраны все строки – как из правой, так и из левой таблицы. При совпадении значений по соединяемым столбцам строка содержит значения как из левой, так и из правой таблицы.

В противном случае, вместо отсутствующих значений в столбцы таблицы (левой или правой) заносится значение NULL.



FULL JOIN

select * from dep

Dep_ID	Dep_Name
10	Accounting
20	IT
30	Useless
40	Security

select * from emp

Last_Name	Dep_ID
King	10
Brooks	10
Bush	20
Smith	30
Baker	

select
e.dep_id
,e.last_name
,d.dep_id
,d.dep_name
from dep d **full join** emp e
on e.dep_id = d.dep_id

	Accounting 10	IT 20	Useless 30	Security 40
King 10				
Brooks 10				
Bush 20				
Smith 30				
Baker NULL				

FULL JOIN

T_RESOURCES

Name	Type
T_ID	NUMBER(11)
T_NAME	VARCHAR2(16)
T_USERID	NUMBER(11)

T_USERS

Name	Type
T_ID	NUMBER(11)
T_NICK	VARCHAR2(16)

```
SELECT T_RESOURCES.T_NAME,  
       T_USERS.T_NICK  
FROM T_RESOURCES FULL OUTER JOIN T_USERS  
ON T_USERS.T_ID = T_RESOURCES.T_USERID
```

	T_NAME	T_NICK
1	res 2	user 1
2	res 5	user 3
3	res 1	user 3
4	res 3	
5		user 4

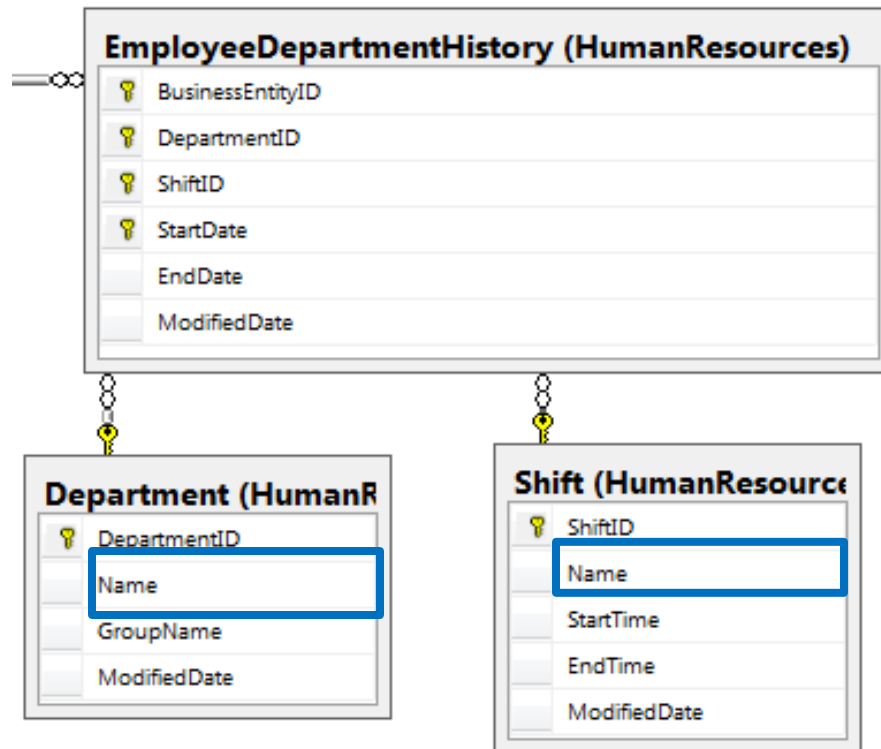
INNER JOIN

LEFT JOIN

RIGHT JOIN

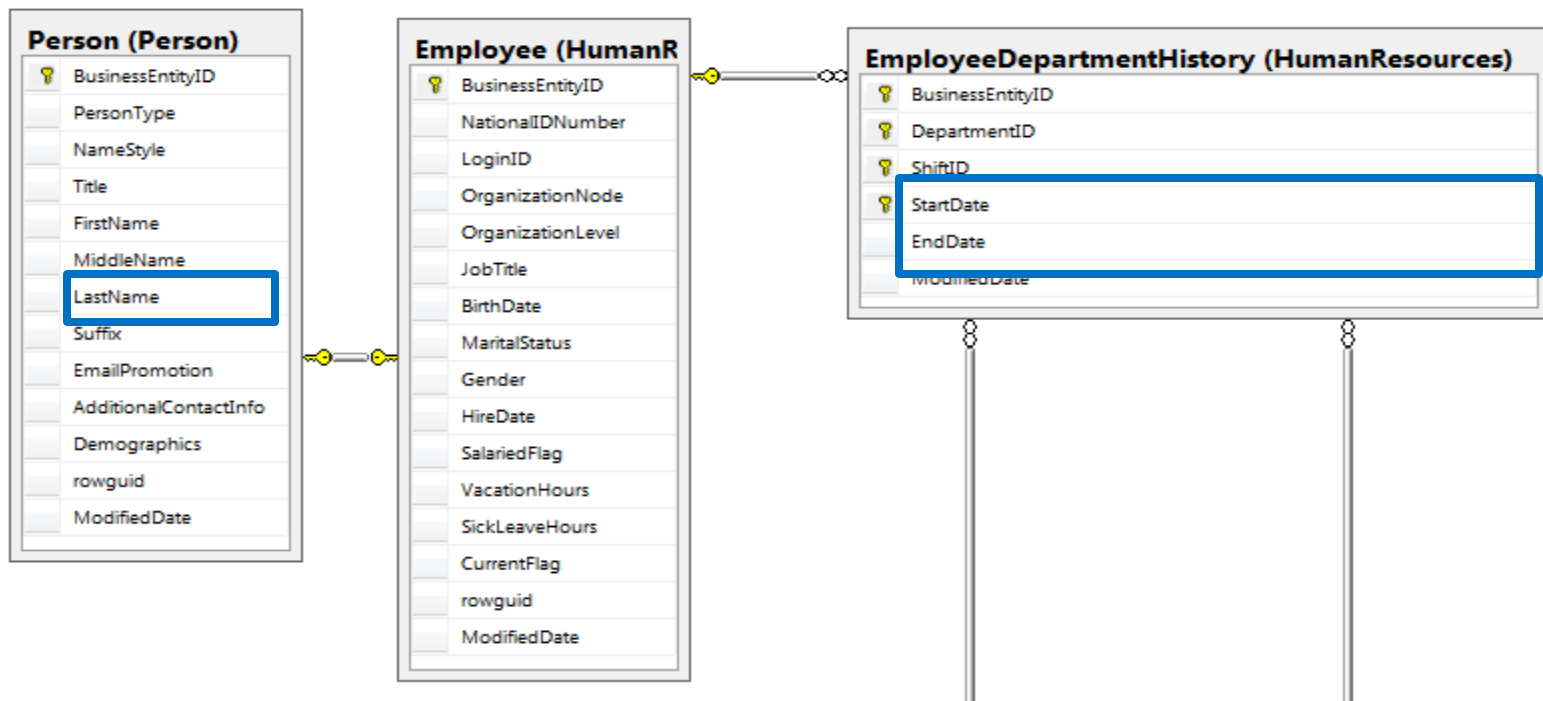
Задания

1. Вывести число сотрудников, работающих на *каждой* смене в *каждом* департаменте.



Задания

2. Вывести фамилии сотрудников, сменивших работу в 2004 году.
3. Вывести фамилии сотрудников, сменивших по крайней мере 2 должности.



Задания

4. Определить сумму отпускных и больничных в каждом отделе. Первыми отображать отделы с максимальными общими отпусками (т.е. отпускные + больничные)

