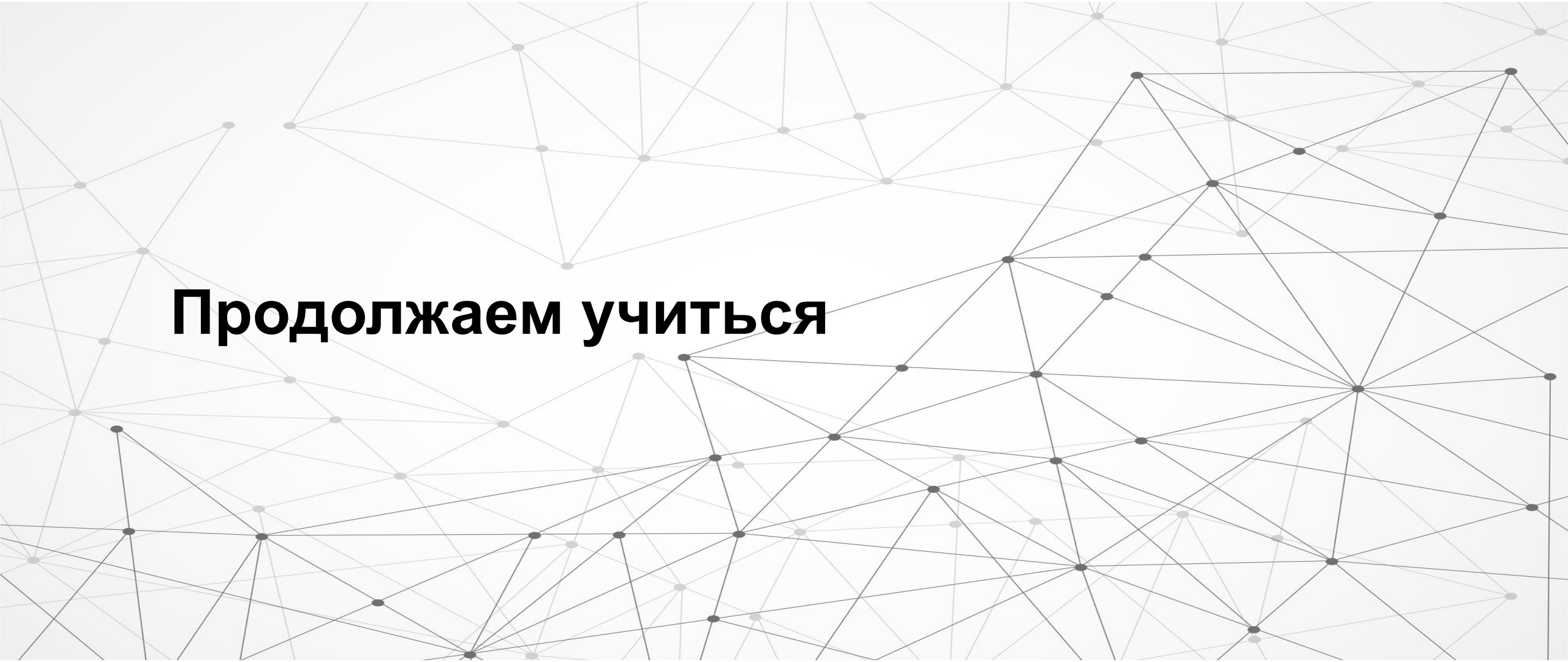
Продолжаем учиться



Data Science Academy

В КОНЦЕ ЗАНЯТИЯ ВЫ ИЗУЧИТЕ :

- Объединения по равенству
- Объединения по неравенству
- Внешние объединения
- Самообъединения
- Естественные объединения
- Полные (или двусторонние) внешние объединения
- UNION
- Виды связей

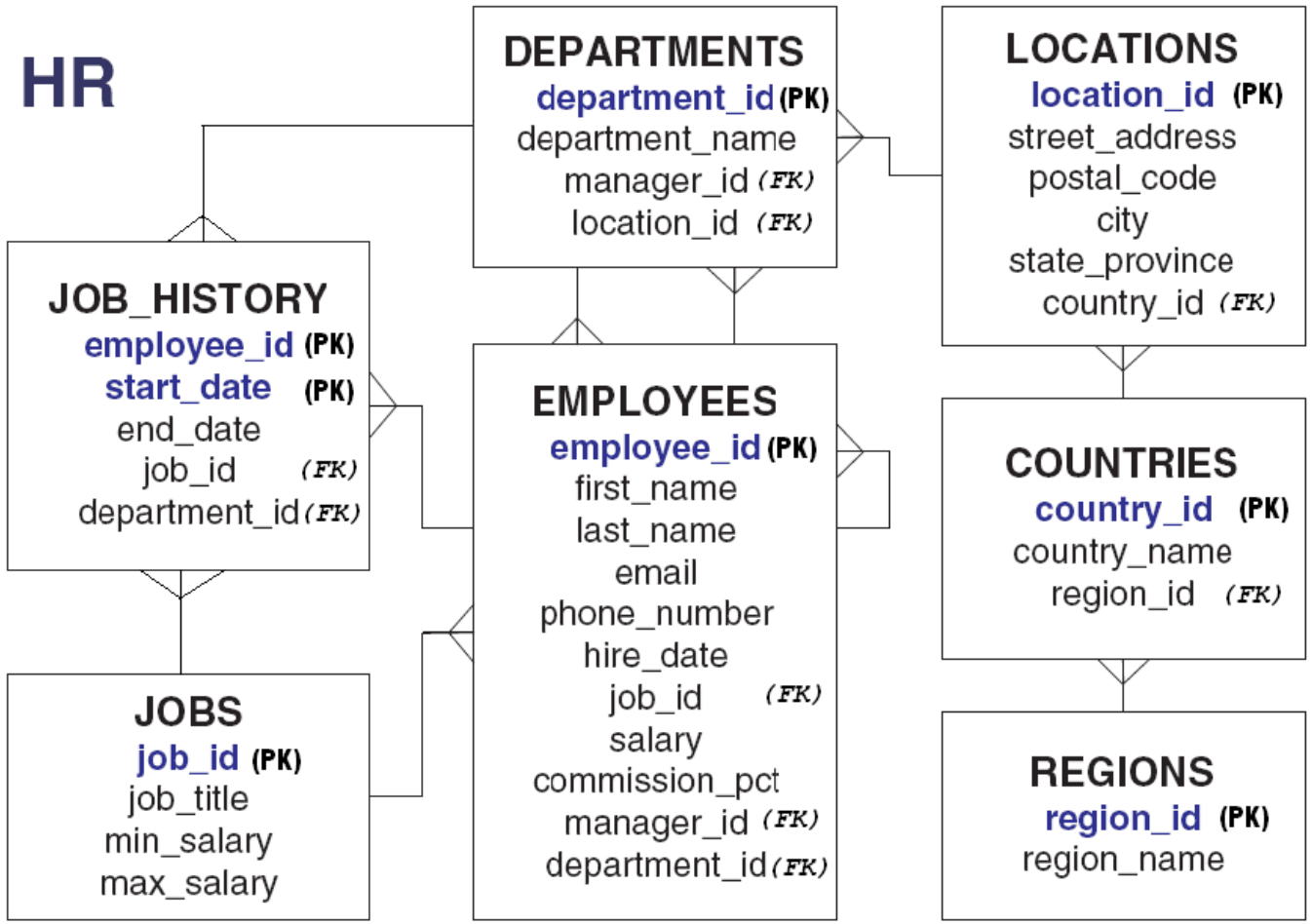
Data Science Academy

Тема 1

Типы объединений JOINS и их синтаксис

- Естественное объединение:
  - предложение USING
  - предложение ON
- Самообъединение
- Внешнее объединение OUTER:
  - левое внешнее объединение LEFT OUTER
  - правое внешнее объединение RIGHT OUTER
  - полное внешнее объединение FULL OUTER
- Декартово произведение – перекрестное объединение

Вспомним схему БД HR



Data Science Academy

Типы объединений Со стандартом SQL:1999 совместимы следующие объединения:

- естественные объединения:
  - предложение NATURAL JOIN
  - предложение USING
  - предложение ON
- внешние объединения:
  - левое внешнее объединение LEFT OUTER JOIN
  - правое внешнее объединение RIGHT OUTER JOIN
  - полное внешнее объединение FULL OUTER JOIN
- перекрестные объединения

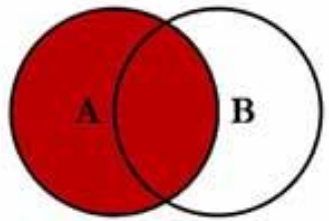
Объединение таблиц с помощью синтаксиса SQL:1999

Объединение используется для запроса данных из нескольких таблиц:

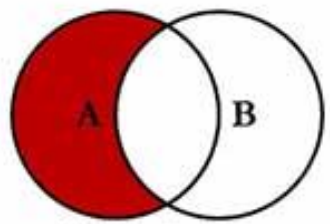
```
SELECT   table1.column, table2.column
FROM     table1
[NATURAL JOIN table2] |
[JOIN table2 USING (column_name)] |
[JOIN table2
  ON (table1.column_name = table2.column_name)]|
[LEFT|RIGHT|FULL OUTER JOIN table2
  ON (table1.column_name = table2.column_name)]|
[CROSS JOIN table2];
```

Data Science Academy

JOIN

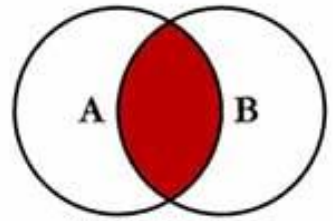



```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```

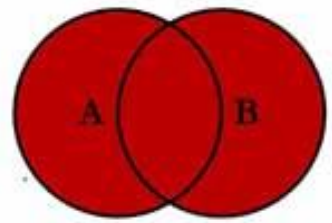



```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```

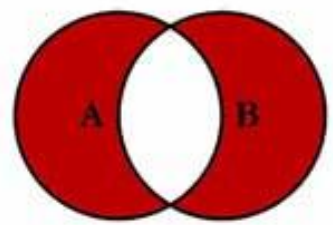
=




```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```

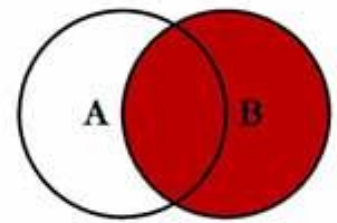



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```

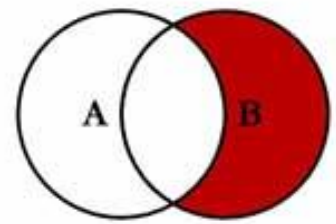



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

Oracle




```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```




```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```

SQL

Data Science Academy

Определение неоднозначных имен столбцов

• Для точного определения имен столбцов, повторяющихся в нескольких таблицах, необходимо использовать префиксы с именами таблиц.
• Префиксы с именами таблиц позволяют повысить производительность.
• В префиксах вместо полного имени таблицы можно использовать псевдонимы.
• Псевдоним позволяет сократить имя таблицы:
  – Сокращается код SQL, используется меньше памяти.
• Псевдонимы столбцов используются для различения столбцов с одинаковыми именами, находящимися в разных таблицах.

Создание естественных объединений

• Предложение NATURAL JOIN при объединении двух таблиц учитывает все столбцы с совпадающими именами.
• Оно выбирает из двух таблиц строки, которые во всех соответствующих столбцах имеют равные значения.
• Если столбцы с совпадающими именами отличаются типом данных, возникает ошибка.

Извлечение записей с помощью естественных объединений

```
select
cust_id,
account_id,
city
from account
natural join customer;
```

CUST_ID	ACCOUNT_ID	CITY
1	61	Lynnfield
1	62	Lynnfield
1	63	Lynnfield

Data Science Academy

Создание объединений с предложением USING

• Если у нескольких столбцов совпадают имена, но отличаются типы данных, естественное объединение можно применить,
использовав предложение USING для указания столбцов, которые должны участвовать в объединении по равенству.

• Предложение USING следует использовать, если при объединении нужно сопоставить только одну из нескольких соответствующих пар столбцов.

• Предложения NATURAL JOIN и USING являются взаимоисключающими.

Извлечение записей с помощью предложения USING

```
select
cust_id,
account_id,
city
from account
join customer
  using (cust_id);
```

CUST_ID	ACCOUNT_ID	CITY
1	61	Lynnfield
1	62	Lynnfield
1	63	Lynnfield
2	64	Woburn
2	65	Woburn

Data Science Academy

Использование псевдонимов таблиц с предложением USING

- Столбец, используемый в предложении USING, уточнять не нужно.
- Если этот же столбец упоминается и в другом месте инструкции SQL, использовать его псевдоним не следует.

```
select
a.account_id,
b.city
from account a
join customer b
  using (cust_id);
```

ACCOUNT_ID	CITY
61	Lynnfield
62	Lynnfield
63	Lynnfield
64	Woburn
65	Woburn

Data Science Academy

Создание объединений с предложением ON

• Условие объединения для естественных объединений
– это в основном объединение по равенству для всех столбцов с одинаковыми именами.
• Для указания произвольных условий или столбцов, участвующих в объединении, следует использовать предложение ON.
• Условие объединения отделено от других условий поиска.
• Предложение ON облегчает понимание кода.

Извлечение записей с помощью предложения ON

```
select
a.cust_id,
a.account_id,
b.city
from account a
join customer b
  on a.cust_id = b.cust_id;
```

CUST_ID	ACCOUNT_ID	CITY
1	61	Lynnfield
1	62	Lynnfield
1	63	Lynnfield
2	64	Woburn
2	65	Woburn

Data Science Academy

Создание трехсторонних объединений с предложением ON

```
select
a.cust_id,
a.account_id,
b.city,
c.birth_date
from account a
join customer b
  on a.cust_id = b.cust_id
join individual c
  on b.cust_id = c.cust_id;
```

CUST_ID	ACCOUNT_ID	CITY	BIRTH_DATE
1	61	Lynnfield	22-APR-72
1	62	Lynnfield	22-APR-72
1	63	Lynnfield	22-APR-72
2	64	Woburn	15-AUG-68
2	65	Woburn	15-AUG-68

Применение к объединению дополнительных условий

Для применения дополнительных условий используйте предложения AND или WHERE:

```
select distinct
a.cust_id,
b.city,
c.first_name,
c.last_name,
c.birth_date
from account a
join customer b
  on a.cust_id = b.cust_id
join individual c
  on b.cust_id = c.cust_id
 and c.birth_date = '22-APR-72';
```

```
select distinct
a.cust_id,
b.city,
c.first_name,
c.last_name,
c.birth_date
from account a
join customer b
  on a.cust_id = b.cust_id
join individual c
  on b.cust_id = c.cust_id
where c.birth_date = '22-APR-72';
```

CUST_ID	CITY	FIRST_NAME	LAST_NAME	BIRTH_DATE
1	Lynnfield	James	Hadley	22-APR-72

Data Science Academy

Само объединения с использованием предложения ON

```
select
a.first_name||' '||a.last_name as emp,
b.first_name||' '||b.last_name as mgr
from employee a
join employee b
  on a.emp_id = b.superior_emp_id;
```

EMP	MGR
Michael Smith	Susan Barker
Michael Smith	Robert Tyler
Robert Tyler	Susan Hawthorne
Susan Hawthorne	John Gooding
Susan Hawthorne	Helen Fleming
Helen Fleming	Chris Tucker
Helen Fleming	Sarah Parker

Сравнение внутренних (INNER) и внешних (OUTER) объединений

• В стандарте SQL:1999 внутренним называется объединение двух таблиц, возвращающее только соответствующие друг другу строки.
• Объединение двух таблиц, возвращающее как результаты внутреннего объединения, так и строки без соответствия из левой (правой) таблицы, называется левым (правым) внешним объединением.
• Объединение двух таблиц, вместе с результатами внутреннего объединения возвращающее результаты левого и правого объединений, называется полным внешним объединением.

Левое внешнее объединение (LEFT OUTER JOIN)

```
select
*
from customer a
left join business b
  on a.cust_id = b.cust_id;
```

CUST_ID	ADDRESS	CITY	CUST_TYPE_CD	FED_ID	POSTAL_CODE	STATE	INCORP_DATE	NAME	STATE_ID	CUST_ID
10	7 Industrial Way	Salem	B	04-1111111	03079	NH	01-MAY-95	Chilton Engineering	12-345-678	10
11	287A Corporate Ave	Wilmington	B	04-2222222	01887	MA	01-JAN-01	Northeast Cooling Inc.	23-456-789	11
12	789 Main St	Salem	B	04-3333333	03079	NH	30-JUN-02	Superior Auto Body	34-567-890	12
13	4772 Presidential Way	Quincy	B	04-4444444	02169	MA	01-MAY-99	AAA Insurance Inc.	45-678-901	13
6	12 Blaylock Ln	Waltham	I	666-66-6666	02451	MA	-	-	-	-
1	47 Mockingbird Ln	Lynnfield	I	111-11-1111	01940	MA	-	-	-	-

Правое внешнее объединение (RIGHT OUTER JOIN)

```
select
*
from business a
right join customer b
  on a.cust_id = b.cust_id;
```

INCORP_DATE	NAME	STATE_ID	CUST_ID	CUST_ID	ADDRESS	CITY	CUST_TYPE_CD	FED_ID	POSTAL_CODE	STATE
01-MAY-95	Chilton Engineering	12-345-678	10	10	7 Industrial Way	Salem	B	04-1111111	03079	NH
01-JAN-01	Northeast Cooling Inc.	23-456-789	11	11	287A Corporate Ave	Wilmington	B	04-2222222	01887	MA
30-JUN-02	Superior Auto Body	34-567-890	12	12	789 Main St	Salem	B	04-3333333	03079	NH
01-MAY-99	AAA Insurance Inc.	45-678-901	13	13	4772 Presidential Way	Quincy	B	04-4444444	02169	MA
-	-	-	-	6	12 Blaylock Ln	Waltham	I	666-66-6666	02451	MA
-	-	-	-	1	47 Mockingbird Ln	Lynnfield	I	111-11-1111	01940	MA
-	-	-	-	7	29 Admiral Ln	Wilmington	I	777-77-7777	01887	MA

Полное внешнее объединение (FULL OUTER JOIN)

```
select
*
from business a
full outer join customer b
  on a.cust_id = b.cust_id;
```

INCORP_DATE	NAME	STATE_ID	CUST_ID	CUST_ID	ADDRESS	CITY	CUST_TYPE_CD	FED_ID	POSTAL_CODE	STATE
-	-	-	-	1	47 Mockingbird Ln	Lynnfield	I	111-11-1111	01940	MA
-	-	-	-	2	372 Clearwater Blvd	Woburn	I	222-22-2222	01801	MA
-	-	-	-	3	18 Jessup Rd	Quincy	I	333-33-3333	02169	MA
-	-	-	-	4	12 Buchanan Ln	Waltham	I	444-44-4444	02451	MA
-	-	-	-	5	2341 Main St	Salem	I	555-55-5555	03079	NH
-	-	-	-	6	12 Blaylock Ln	Waltham	I	666-66-6666	02451	MA
-	-	-	-	7	29 Admiral Ln	Wilmington	I	777-77-7777	01887	MA
-	-	-	-	8	472 Freedom Rd	Salem	I	888-88-8888	03079	NH
-	-	-	-	9	29 Maple St	Newton	I	999-99-9999	02458	MA
01-MAY-95	Chilton Engineering	12-345-678	10	10	7 Industrial Way	Salem	B	04-1111111	03079	NH
01-JAN-01	Northeast Cooling Inc.	23-456-789	11	11	287A Corporate Ave	Wilmington	B	04-2222222	01887	MA
30-JUN-02	Superior Auto Body	34-567-890	12	12	789 Main St	Salem	B	04-3333333	03079	NH

Декартово произведение

• Декартово произведение образуется в следующих случаях:
  – условие объединения не указано
  – условие объединения недопустимо
  – все строки в первой таблице объединяются со всеми строками во второй таблице
• Чтобы избежать образования декартова произведения, необходимо всегда указывать допустимое условие объединения.

Создание перекрестных объединений

• При помощи предложения CROSS JOIN создается перекрестное произведение двух таблиц.
• Оно также называется декартовым произведением двух таблиц.
Вернёт всё, в таблице business 4 записи
в таблице customer 13 записей
4 * 13 = 52, вернёт столько записей

```
select
*
from business a
cross join customer b;
```

Data Science Academy

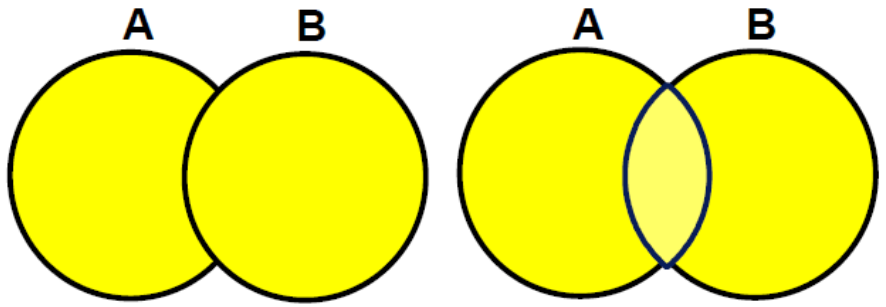
Использование операторов для работы с наборами

Операторы для работы с наборами: типы и указания

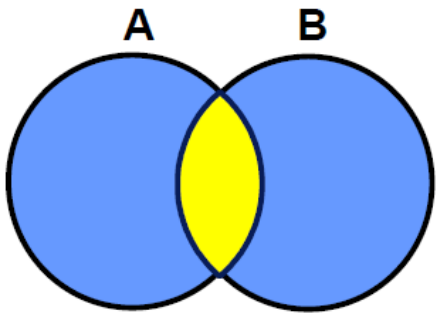
- Таблицы, используемые на этом занятии
- Оператор UNION и UNION ALL
- Оператор INTERSECT
- Оператор MINUS
- Согласование инструкций SELECT
- Использование предложения ORDER BY в операторах для работы с наборами

Data Science Academy

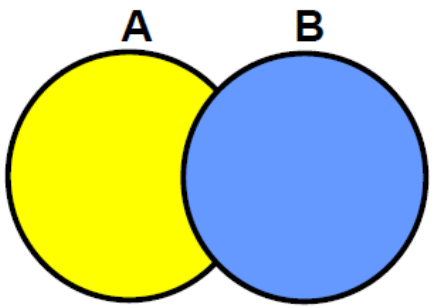
Операторы для работы с наборами



UNION/UNION ALL



INTERSECT



MINUS

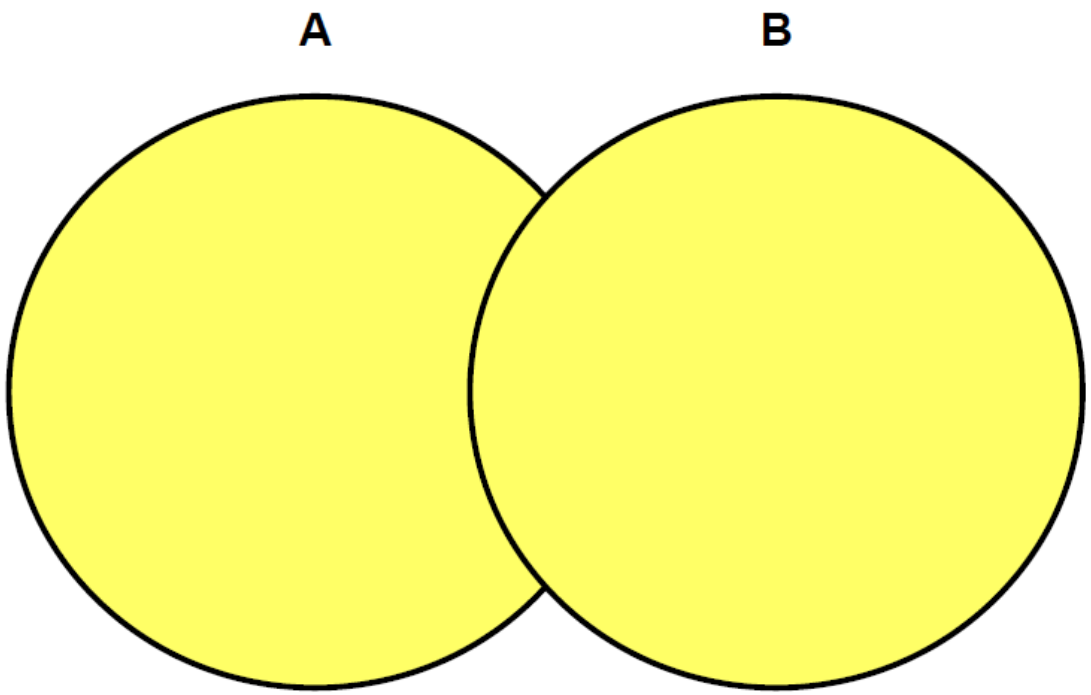
Data Science Academy

Указания по операторам для работы с наборами

• Число выражений в списках SELECT должно быть одинаковым.
• Тип данных каждого столбца второго запроса должен соответствовать типу данных соответствующего ему столбца первого запроса.
• Для изменения последовательности выполнения можно использовать скобки
• Предложение ORDER BY можно помещать только в самом конце инструкции.

Data Science Academy

Оператор `UNION`



Оператор `UNION` возвращает строки из обоих запросов после исключения дубликатов.

Data Science Academy

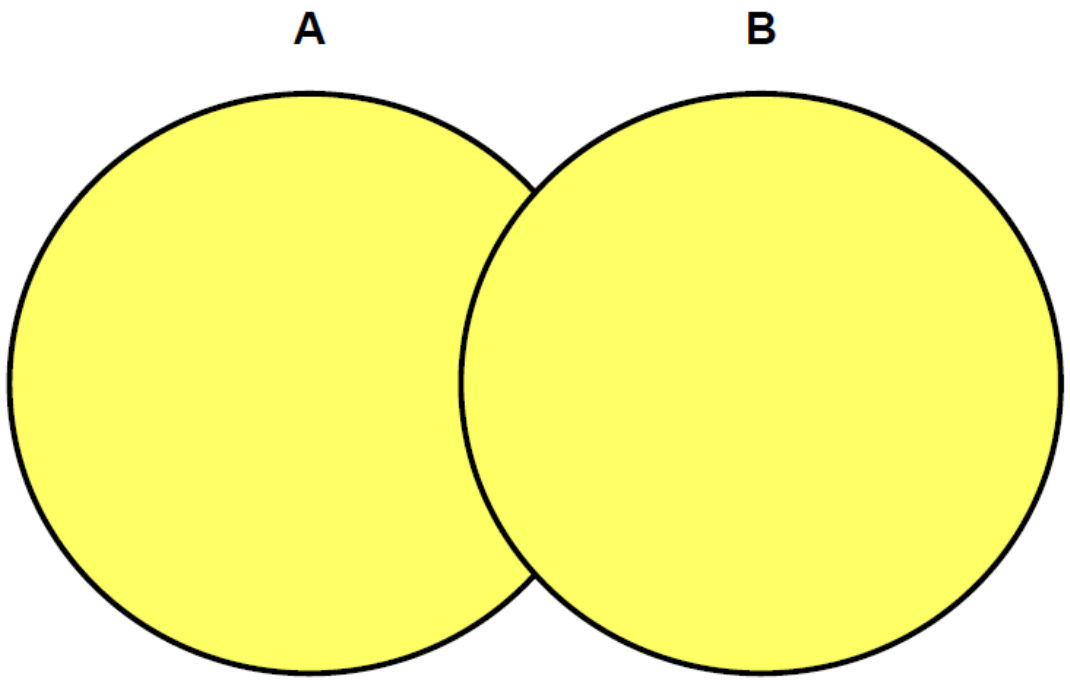
Использование оператора UNION

```
select
account_id,
avail_balance
from account
where 1 = 1
and account_id = 1

UNION

select
account_id,
avail_balance
from account
where 1 = 1
and account_id = 1;
```

ACCOUNT_ID	AVAIL_BALANCE
1	1057.75

Data Science Academy

Оператор `UNION ALL`



Оператор `UNION ALL` возвращает строки из обоих запросов, включая все дубликаты.

Data Science Academy

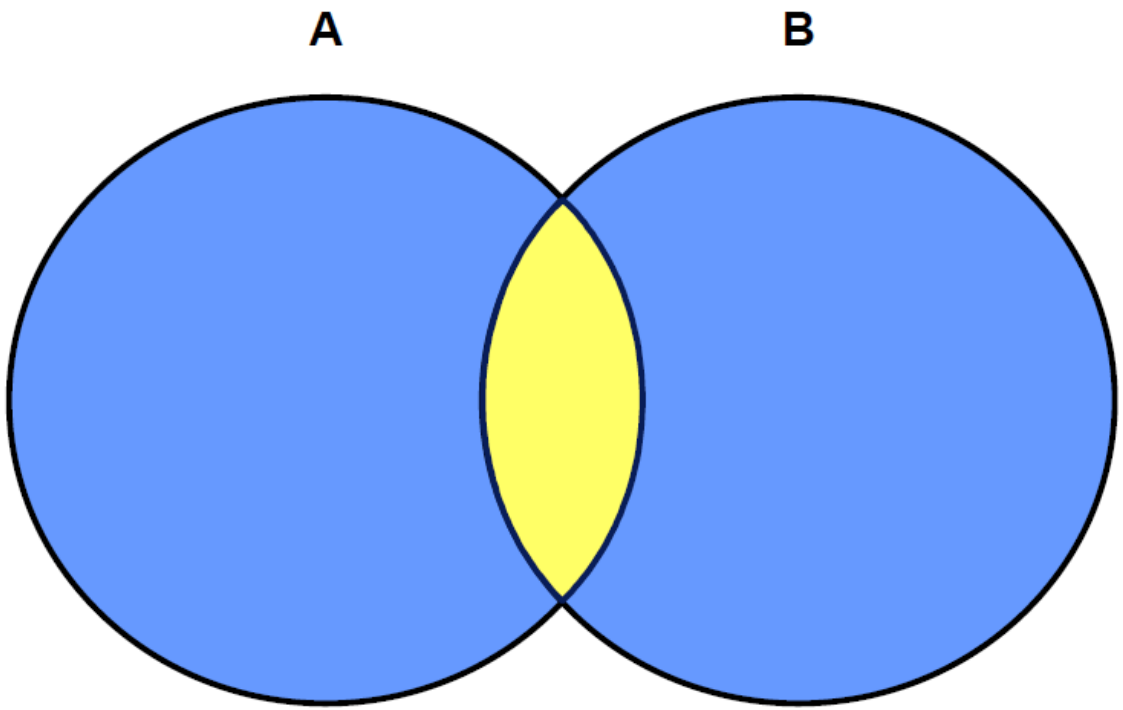
Использование оператора UNION ALL

```
select
account_id,
avail_balance
from account
where 1 = 1
and account_id = 1

UNION ALL

select
account_id,
avail_balance
from account
where 1 = 1
and account_id = 1;
```

ACCOUNT_ID	AVAIL_BALANCE
1	1057.75
1	1057.75

Data Science Academy

Оператор INTERSECT



Оператор INTERSECT возвращает строки, общие для двух запросов.

Data Science Academy

Использование оператора INTERSECT

Вернёт все значения

```
select
account_id,
avail_balance
from account

INTERSECT

select
account_id,
avail_balance
from account;
```

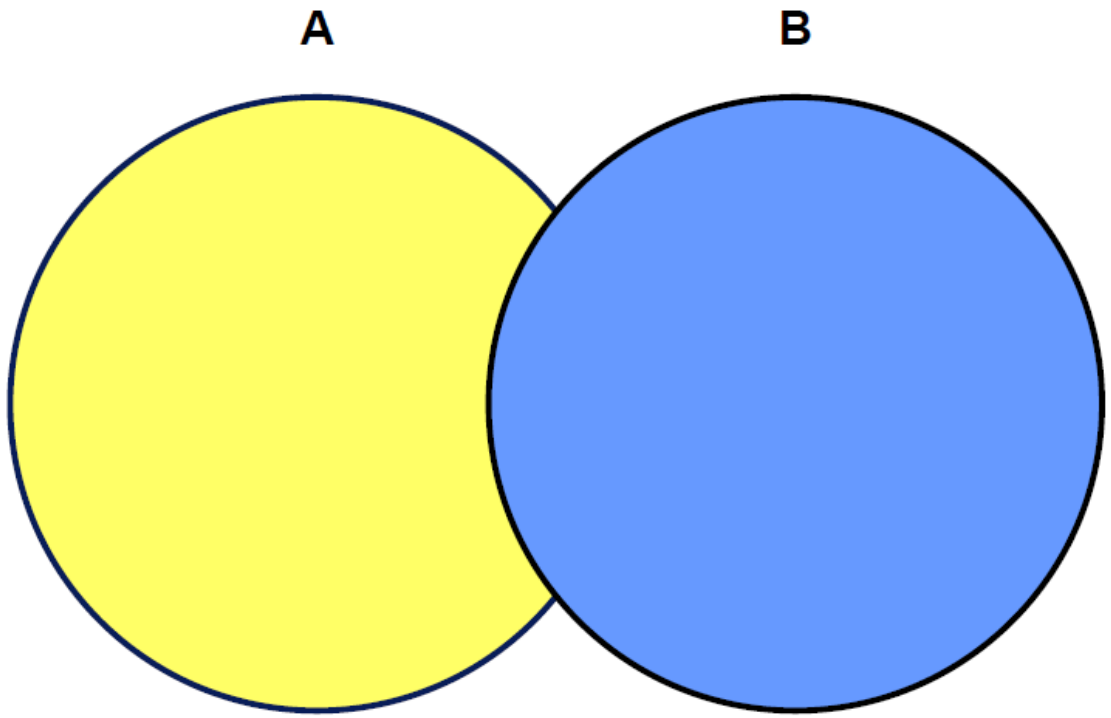
Отображение идентификаторов сотрудника и должности для тех сотрудников, текущая должность которых совпадает с одной из прежних (т. е. ранее они сменили должность, но затем вернулись на нее снова).

```
SELECT employee_id, job_id
FROM hr.employees

INTERSECT

SELECT employee_id, job_id
FROM hr.job_history;
```

EMPLOYEE_ID	JOB_ID
120	ST_MAN
176	SA_REP
200	AD_ASST

Data Science Academy

Оператор MINUS



Оператор MINUS возвращает все уникальные строки, которые выбраны первым запросом, но отсутствуют в наборе результатов второго запроса.

Data Science Academy

Использование оператора MINUS

```
select
account_id,
product_cd
from account

MINUS

select
account_id,
product_cd
from account
where product_cd = 'CHK';
```

ACCOUNT_ID	PRODUCT_CD
2	SAV
3	CD
5	SAV
7	MM

Data Science Academy

Использование предложения ORDER BY в операторах для работы с наборами

- Предложение ORDER BY можно использовать только один раз, причем в конце составного запроса.
- Компоненты запроса не могут иметь свои собственные предложения ORDER BY.
- Предложение ORDER BY распознает только столбцы первого запроса SELECT.
- По умолчанию результаты сортируются в порядке возрастания, при этом используются значения первого столбца первого запроса SELECT.

Онтология предметной области

Data Science Academy

Описание объектов рассматриваемой области и связей между ними называется **онтологией** предметной области.

онтологию хорошо знают эксперты соответствующей области

в бухгалтерии — бухгалтер

в обучении — преподаватель

И т.д.

Но, в отличие от программистов, они часто представляют её на интуитивном уровне, неформально.

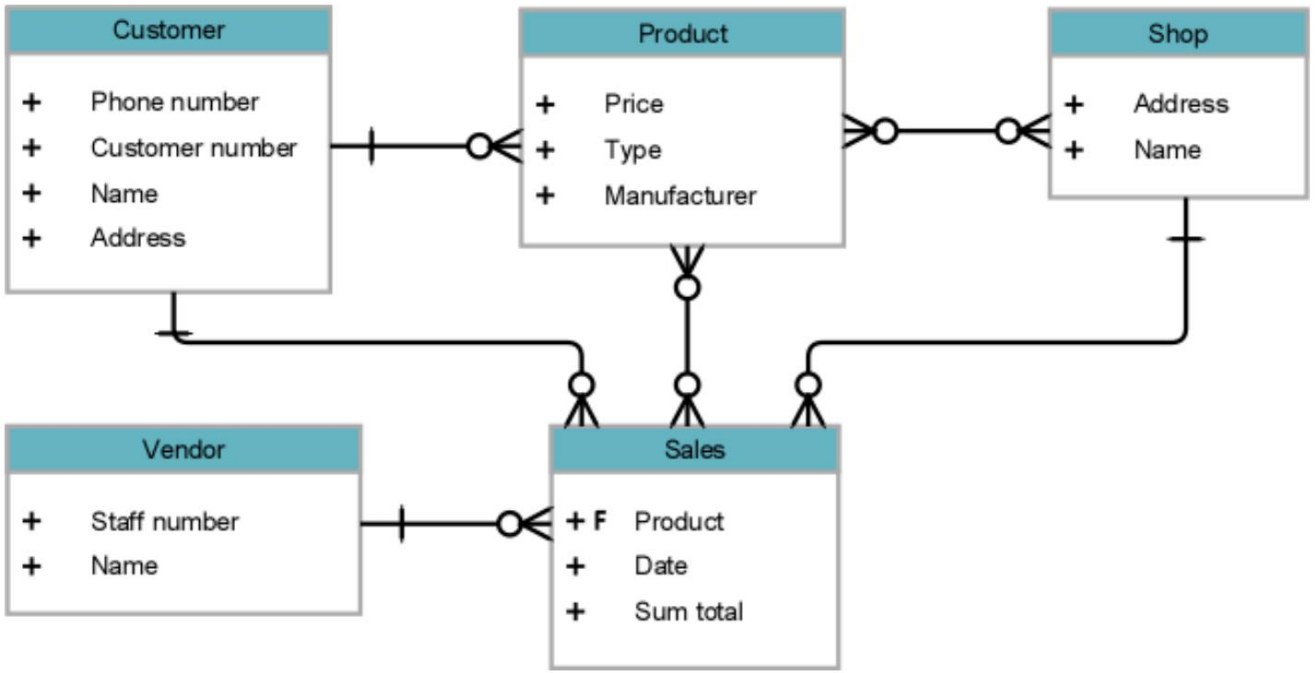
На практике программисты (или бизнес-аналитики и менеджеры) общаются с заказчиками, которые могут сами выступать в роли экспертов и строят вместе с ними формальную онтологию (этот процесс происходит постоянно в процессе развития проекта и не выделяется в отдельный этап проектирования). То есть выделяют конкретные термины, договариваются о том, что они означают и как связаны друг с другом. Затем, используя ER-модель, программист формирует необходимую модель данных.

Data Science Academy

Каждая сущность в реляционной базе данных представлена таблицей, а связи между сущностями реализуются через внешние ключи.

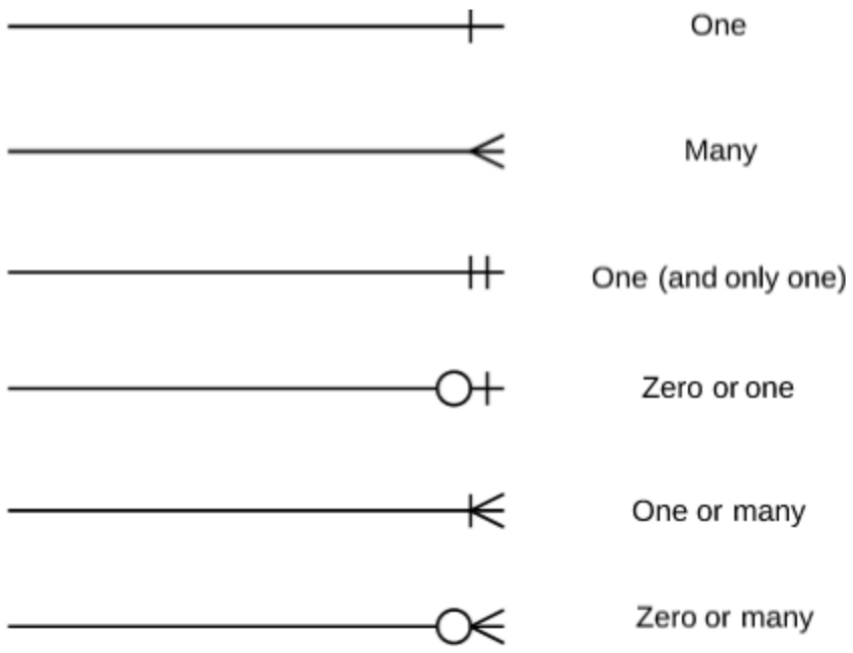
ПРИМЕР СХЕМЫ БД

Entity-Relationship Diagram



Data Science Academy

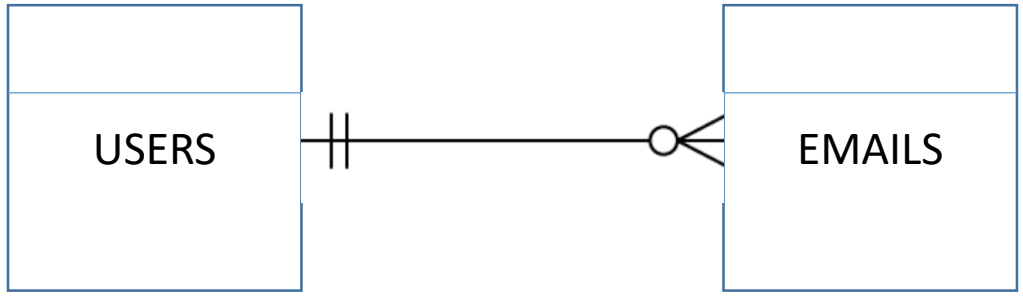
В ERD, каждая сущность представлена блоком, в котором перечисляются поля. Между блоками рисуются линии, имеющие некоторые заранее определенные концы, они определяют тип связи между сущностями.



ВИДЫ СВЯЗЕЙ:
1) Один ко Многим (one2many)
2) Один к Одному (one2one)
3) Многие ко многим (many2many)

Data Science Academy

Один ко Многим (one2many)



Если мы хотим узнать все емейлы, которые есть у пользователя с идентификатором 1, то нужно выполнить такой запрос:

```sql
SELECT * FROM emails WHERE user_id = 1;
```

users

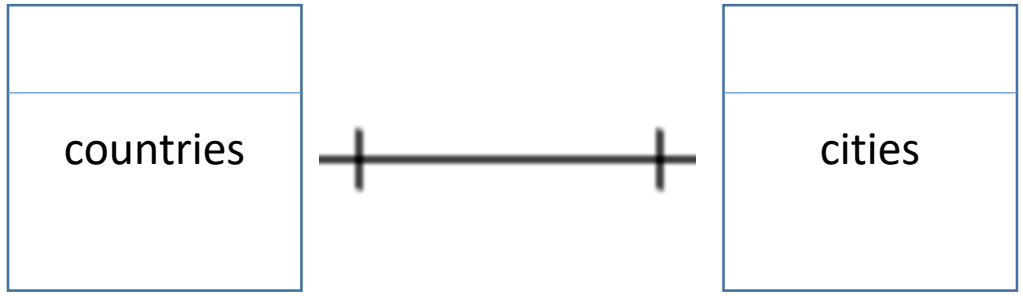
id	first_name	last_name	created_at
1	Сергей	Петров	11.10.2005
38	Иван	Носов	03.08.2000
22	Виктор	Пирогов	23.12.2011

emails

id	user_id	email
1	1	serj@gmail.com
2	1	petrov@mail.ru
10	38	ivan@yahoo.com
22	22	vkurg@indbox.com

Data Science Academy

Один к Одному(o2o)



Связь o2o обычно существует не сама по себе, а внутри связи o2m. То есть у каждой страны есть города, но только один из них столица.

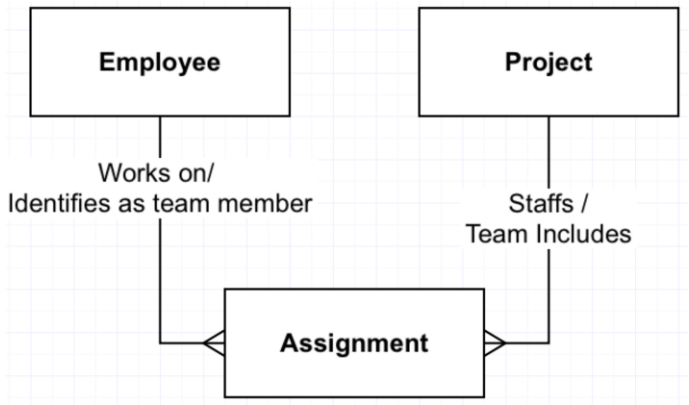
```
select *
from cities
where country_id = 88
and capital = 'TRUE'
```

countries		
id	name	created_date
88	Kazakhstan	16.12.1991
1	USA	04.07.1776
8	Germany	18.01.1871
2	Russia	24.08.1991
91	Canada	01.07.1867

cities				
id	name	country_id	capital	created_date
1	Nur-Sultan	88	TRUE	11.10.2005
2	New York	1		20.08.1790
3	Almaty	88		11.10.1918
4	Moscow	2	TRUE	11.10.1811
5	Karagandy	88		11.10.1961

Data Science Academy

Многие ко многим (many2many)



Если мы захотим узнать все курсы, которые проходит определенный пользователь, то выполним такой запрос:

```
SELECT course_id FROM course_members WHERE user_id = 3;
```

users

id	first_name	created_at
2	Сергей	11.10.2005
38	Иван	03.08.2000
22	Виктор	23.12.2011

courses

id	name	created_at
8	PHP basics	11.10.2005
55	Python basics	03.08.2000
22	Ruby basics	23.12.2011

course_members

id	user_id	course_id	created_at
34	2	8	11.10.2005
33	38	55	03.08.2000
99	22	22	23.12.2011
4	22	8	23.12.2011
5	38	22	23.12.2011

Data Science Academy