

Продолжаем учиться

В КОНЦЕ ЗАНЯТИЯ ВЫ ИЗУЧИТЕ :

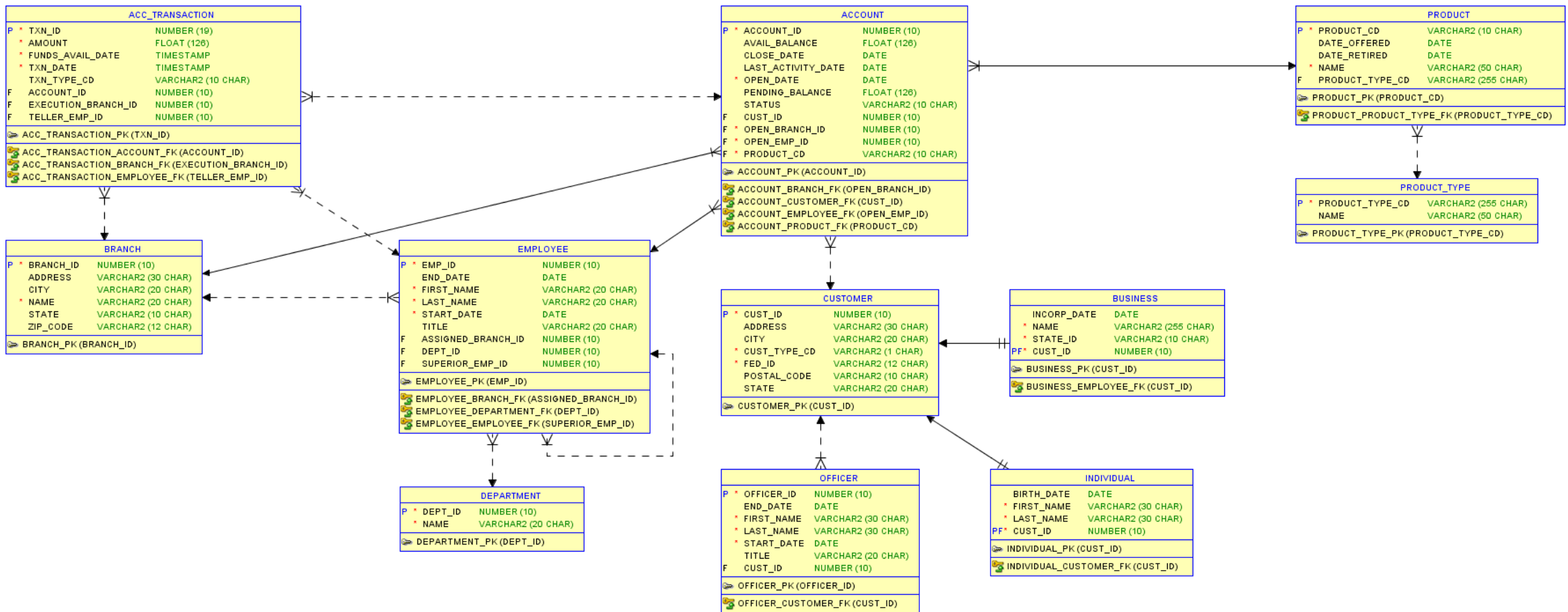
- Объединения по равенству
- Объединения по неравенству
- Внешние объединения
- Самообъединения
- Естественные объединения
- Полные (или двусторонние) внешние объединения

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 Банк



Типы объединений Со стандартом SQL:1999 совместимы следующие объединения:

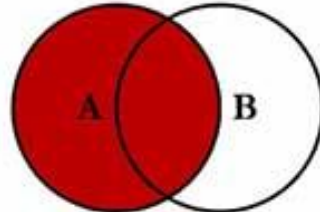
- естественные объединения:
 - предложение NATURAL JOIN
 - предложение USING
 - предложение ON
- внешние объединения:
 - левое внешнее объединение LEFT OUTER JOIN
 - правое внешнее объединение RIGHT OUTER JOIN
 - полное внешнее объединение FULL OUTER JOIN
- перекрестные объединения

Объединение таблиц с помощью синтаксиса SQL:1999

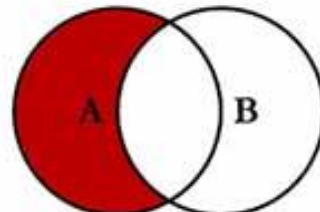
Объединение используется для запроса данных из нескольких таблиц:

```
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];
```

JOIN

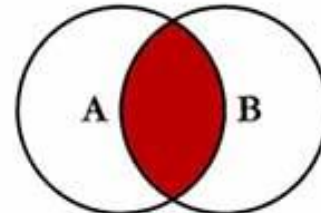


```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```

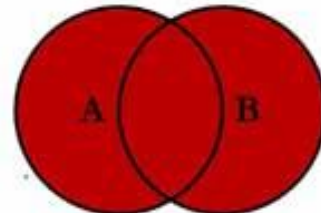


```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```

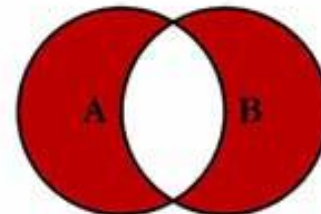
=



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

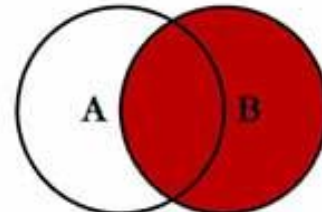


```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```

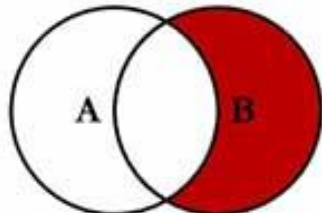


```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

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

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

Создание объединений с предложением USING

- Если у нескольких столбцов совпадают имена, но отличаются типы данных, естественное объединение можно применить, используя 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

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

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

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

Самообъединения 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;
```