

# Реляционная модель данных и основы языка SQL

Семинар 1



# Декартово произведение

Для заданных конечных множеств  $D_1, D_2, \dots, D_n$  (не обязательно различных) **декартовым произведением**  $D_1 * D_2 * \dots * D_n$  называется множество произведений вида:

$d_1 * d_2 * \dots * d_n$ , где  $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$

Пример:  $A = \{a_1, a_2, a_3\}$  и  $B = \{b_1, b_2\}$

$C = A * B$

$= \{a_1 * b_1, a_1 * b_2, a_2 * b_1, a_2 * b_2, a_3 * b_1, a_3 * b_2\}$

# Предикат

**Предикат** ( $n$ -местный, или  $n$ -арный) — это функция с областью значений  $\{0,1\}$ , определённая на  $n$ -й декартовой степени множества  $D$  ( $D = D_1 * D_2 * \dots * D_n$ ). Таким образом, каждую  $n$ -ку элементов  $D$  он характеризует либо как «истинную», либо как «ложную».

Пример:  $x^2 + y^2 \leq 9$  двухместный предикат, заданный над множеством  $R^2$

# Домен

- Домен имеет уникальное имя (в пределах базы данных).
- Домен определен на некотором простом типе данных или на другом домене.
- Домен может иметь некоторое логическое условие, позволяющее описать подмножество данных, допустимых для данного домена.
- Домен несет определенную смысловую

нагрузку.  
Пример:  $D \doteq \{n \in N \mid n \geq 18 \ \& \ n \leq 65\}$

# Отношения, атрибуты, кортежи

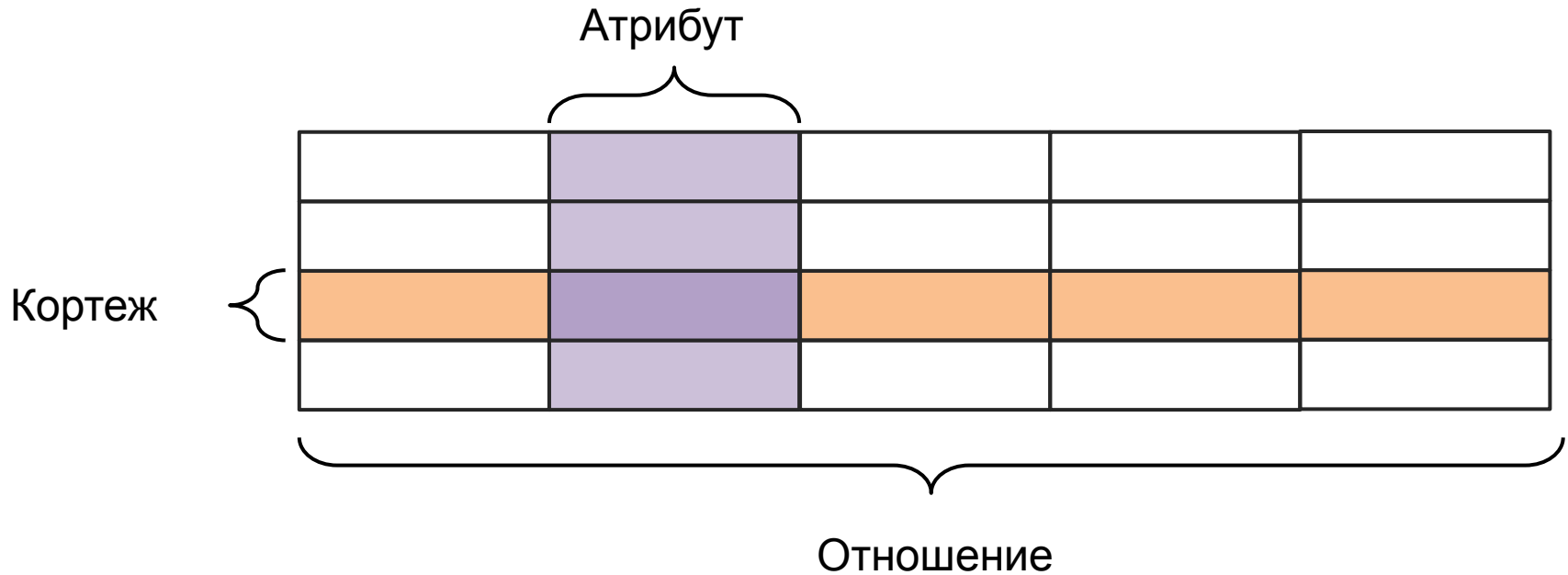
Атрибут отношения есть пара вида  $\langle \text{Имя\_атрибута} : \text{Имя\_домена} \rangle$ .

Отношением  $R$ , определенным на множествах доменов  $D_1, D_2, \dots, D_n$  называется подмножество декартова произведения  $D_1 * D_2 * \dots * D_n$ .

Заголовок отношения содержит фиксированное количество атрибутов отношения.

Тело отношения содержит множество кортежей отношения. Каждый кортеж отношения представляет собой множество пар вида  $\langle \text{Имя\_атрибута} : \text{Значение\_атрибута} \rangle$

# Основные компоненты реляционного отношения



# Основные компоненты реляционного отношения

Отношение	целое	строка		целое	Типы данных	
	номер	ИМЯ	должность	деньги	Домены	
	Табельный номер	Имя	Должность	Оклад	Премия	Атрибуты
	2934	Иванов	инженер	112	40	Кортежи
2935	Петров	вед. инженер	144	50		
2936	Сидоров	бухгалтер	92	35		

↑  
Ключ



# Свойства отношений

- Отсутствие кортежей-дубликатов
- Отсутствие упорядоченности кортежей
- Отсутствие упорядоченности атрибутов
- Атомарность значений атрибутов

# Structured Query Language

- **Data Definition Language (DDL)** – язык определения данных
- **Data Manipulation Language (DML)** – язык манипулирования данными
- **DCL - Data Control Language (DCL)** – язык определения доступа к данным
- **TCL - Transaction Control Language (TCL)** – язык управления транзакциями

# Select

```
SELECT [ALL | DISTINCT] <имя_столбца |  
перечень_столбцов>  
FROM <имя_таблицы>, ...  
[ WHERE <условие> ]  
[ GROUP BY <имя_столбца>,... ]  
[ HAVING <условие> ]  
[ ORDER BY <имя_столбца>,... [ASC | DESC],...  
]
```

# Select

```
SELECT [ALL | DISTINCT] <имя_столбца |  
перечень_столбцов>  
FROM <имя_таблицы>, ...  
[ WHERE <условие> ]  
[ GROUP BY <имя_столбца>,... ]  
[ HAVING <условие> ]  
[ ORDER BY <имя_столбца>,... [ASC | DESC],...  
]
```

# SELECT \* FROM Employees

EMP_ID NUMBER(10)	MANAGER_ID NUMBER(10)	PHONE VARCHAR2(100)	HIRE_DATE DATE	LAST_NAME VARCHAR2(100)	FIRST_NAME VARCHAR2(100)	PATRONYMIC VARCHAR2(100)	POST VARCHAR2(100)	START_DATE DATE	END_DATE DATE
1	(null)	+7(943)3044957	10.03.1995 0:00:00	Баранова	Варвара	Платоновна	Курьер	29.06.2018 0:00:00	(null)
2	12	+7(923)7366278	11.12.2001 0:00:00	Белова	Алиса	Валерьевна	Повар	15.03.2018 0:00:00	(null)
3	12	+7(943)6088737	27.05.1996 0:00:00	Белов	Владимир	Степанович	Повар	05.02.2018 0:00:00	(null)
4	1	+7(923)7862996	17.02.1997 0:00:00	Белов	Игорь	Степанович	Курьер	26.03.2018 0:00:00	(null)
5	1	+7(923)9206719	21.12.1997 0:00:00	Белов	Сергей	Степанович	Курьер	21.06.2018 0:00:00	(null)
6	1	+7(923)7349957	26.12.1996 0:00:00	Беляева	Екатерина	Петровна	Курьер	22.05.2018 0:00:00	(null)
7	13	+7(913)6813761	22.07.1997 0:00:00	Большакова	Елена	Антоновна	Повар	29.06.2018 0:00:00	(null)
8	1	+7(923)9772242	28.01.2001 0:00:00	Волков	Михаил	Борисович	Курьер	16.03.2018 0:00:00	(null)
9	1	+7(951)9087987	06.09.2001 0:00:00	Волкова	Екатерина	Николаевна	Курьер	17.02.2018 0:00:00	(null)
10	(null)	+7(923)3209852	28.11.1997 0:00:00	Голубева	Кристина	Константиновна	Курьер	27.03.2018 0:00:00	(null)
11	13	+7(913)1689062	09.10.1997 0:00:00	Громов	Иван	Иванович	Повар	23.05.2018 0:00:00	(null)
12	(null)	+7(923)1960324	04.12.1990 0:00:00	Гусева	Татьяна	Егоровна	Повар	18.04.2018 0:00:00	(null)
13	(null)	+7(943)0502443	05.10.1991 0:00:00	Давыдов	Дмитрий	Романович	Повар	19.02.2018 0:00:00	(null)
14	10	+7(913)1031303	10.07.2000 0:00:00	Данилова	Полина	Анатольевна	Курьер	28.04.2018 0:00:00	(null)
15	10	+7(923)9693899	28.01.1998 0:00:00	Денисов	Игорь	Алексеевич	Курьер	22.04.2018 0:00:00	(null)
16	18	+7(913)0141661	25.04.2002 0:00:00	Дорофеева	Ева	Матвеевна	Менеджер	08.04.2018 0:00:00	(null)
17	10	(null)	21.01.2001 0:00:00	Егоров	Михаил	Артемович	Курьер	08.06.2018 0:00:00	(null)
18	(null)	+7(951)6728895	23.11.1988 0:00:00	Журавлева	Анастасия	Игоревна	Менеджер	27.02.2018 0:00:00	(null)
19	13	+7(943)8626345	11.05.2000 0:00:00	Зайцев	Максим	Георгиевич	Повар	08.03.2018 0:00:00	(null)
20	26	+7(923)5793944	25.06.1998 0:00:00	Ильин	Александр	Вячеславович	Курьер	26.02.2018 0:00:00	(null)
21	26	+7(913)7158432	05.04.2001 0:00:00	Казачкова	Варвара	Владимировна	Курьер	07.06.2018 0:00:00	(null)
22	12	+7(923)1832229	24.03.2000 0:00:00	Колесникова	Елена	Богдановна	Повар	14.05.2018 0:00:00	(null)
23	18	+7(923)9353161	26.05.1999 0:00:00	Лебедева	Надежда	Станиславовна	Менеджер	08.04.2018 0:00:00	(null)
24	26	+7(923)2212543	24.03.2002 0:00:00	Макарова	Анна	Петровна	Курьер	16.07.2018 0:00:00	(null)
25	13	+7(923)8389157	13.11.1997 0:00:00	Максимов	Иван	Борисович	Повар	13.05.2018 0:00:00	(null)

**SELECT POST**  
**FROM** Employees

POST VARCHAR2(100)
Курьер
Повар
Повар
Курьер
Курьер
Курьер
Повар
Курьер
Курьер
Курьер
Повар
Повар
Повар
Курьер
Курьер
Менеджер

**SELECT DISTINCT POST**  
**FROM** Employees

# Select

```
SELECT [ALL | DISTINCT] <имя_столбца |  
перечень_столбцов>  
FROM <имя_таблицы>, ...  
[ WHERE <условие> ]  
[ GROUP BY <имя_столбца>,... ]  
[ HAVING <условие> ]  
[ ORDER BY <имя_столбца>,... [ASC | DESC],...  
]
```

# Логические операции и Операторы сравнения

= Равно

> Больше чем

< Меньше чем

>= Больше или равно

<= Меньше или равно

<> Не равно

AND – операция И

OR – операция ИЛИ

NOT – операция НЕ

LIKE – Поиск по заданному шаблону

BETWEEN – Принадлежность диапазону

IN – Принадлежность списку



```
SELECT EMP_ID, MANAGER_ID, LAST_NAME, FIRST_NAME
FROM Employees
WHERE MANAGER_ID = 10;
```

EMP_ID NUMBER(10)	MANAGER_ID NUMBER(10)	LAST_NAME VARCHAR2(100)	FIRST_NAME VARCHAR2(100)
14	10	Данилова	Полина
15	10	Денисов	Игорь
17	10	Егоров	Михаил
34	10	Титов	Михаил

```
SELECT EMP_ID, MANAGER_ID, LAST_NAME, FIRST_NAME  
FROM Employees  
WHERE MANAGER_ID = 10 OR EMP_ID = 10;
```

EMP_ID NUMBER(10)	MANAGER_ID NUMBER(10)	LAST_NAME VARCHAR2(100)	FIRST_NAME VARCHAR2(100)
10	<i>(null)</i>	Голубева	Кристина
14	10	Данилова	Полина
15	10	Денисов	Игорь
17	10	Егоров	Михаил
34	10	Титов	Михаил

```
SELECT EMP_ID, MANAGER_ID, LAST_NAME, FIRST_NAME  
FROM Employees  
WHERE MANAGER_ID = 10 OR EMP_ID = 10;
```

EMP_ID NUMBER(10)	MANAGER_ID NUMBER(10)	LAST_NAME VARCHAR2(100)	FIRST_NAME VARCHAR2(100)
10	<i>(null)</i>	Голубева	Кристина
14	10	Данилова	Полина
15	10	Денисов	Игорь
17	10	Егоров	Михаил
34	10	Титов	Михаил

**SELECT PRODUCT\_ID, PRODUCT\_NAME, PRICE FROM PRODUCTS WHERE PRODUCT\_ID IN (1,10,11,8);**

PRODUCT_ID NUMBER(38)	PRODUCT_NAME VARCHAR2(1000)	PRICE NUMB...
1	Пепперони	500
8	Полло	430
10	Вендетта	480
11	Манхэттен	520

**SELECT PRODUCT\_ID, PRODUCT\_NAME, PRICE FROM PRODUCTS WHERE PRICE BETWEEN 480 AND 500;**

PRODUCT_ID NUMBER(38)	PRODUCT_NAME VARCHAR2(1000)	PRICE NUMB...
1	Пепперони	500
3	Карбонара	490
9	Маринара	490
10	Вендетта	480
13	Дьябло	490

**SELECT PRODUCT\_ID, PRODUCT\_NAME, PRICE FROM PRODUCTS WHERE PRICE BETWEEN 480 AND 500 AND PRODUCT ID IN (1,10,11,8);**

PRODUCT_ID NUMBER(38)	PRODUCT_NAME VARCHAR2(1000)	PRICE NUMB...
1	Пепперони	500
10	Вендетта	480

# Поиск по шаблону (**LIKE**)

**%** - любое количество символов (в том числе и 0) подряд

**\_** - любой одиночный символ

**ESCAPE** - поиск специальных символов (% и \_).  
ESCAPE определяет символ после которого любой следующий за ним символ интерпретируется как обычный.

– Пример: '%^%%' ESCAPE '^' – поиск строки в которой присутствует символ %

```
SELECT EMPLOYEE_ID, LAST_NAME
FROM EMPLOYEES
WHERE LAST_NAME LIKE '%бе%';
```

EMP_ID NUMBER(10)	LAST_NAME VARCHAR2(100)
2	Белова
3	Белов
4	Белов
5	Белов
6	Беляева
10	Голубева
23	Лебедева

```
SELECT EMPLOYEE_ID, LAST_NAME
FROM EMPLOYEES
WHERE LAST_NAME LIKE '__бе%';
```

EMP_ID NUMBER(10)	LAST_NAME VARCHAR2(100)
23	Лебедева

```
SELECT EMPLOYEE_ID, LAST_NAME
FROM EMPLOYEES
WHERE LAST_NAME LIKE '__%бе%';
```

EMP_ID NUMBER(10)	LAST_NAME VARCHAR2(100)
10	Голубева
23	Лебедева

```
SELECT EMPLOYEE_ID, LAST_NAME
FROM EMPLOYEES
WHERE LAST_NAME LIKE 'бе%';
```

EMP_ID NUMBER(10)	LAST_NAME VARCHAR2(100)
2	Белова
3	Белов
4	Белов
5	Белов
6	Беляева

# NULL

- Сравнение: **IS [NOT] NULL**

```
SELECT EMPLOYEE_ID, LAST_NAME, POST  
FROM EMPLOYEES  
WHERE MANAGER_ID IS NULL;
```

EMP_ID NUMBER(10)	LAST_NAME VARCHAR2(100)	POST VARCHAR2(100)
1	Баранова	Курьер
10	Голубева	Курьер
12	Гусева	Повар
13	Давыдов	Повар
18	Журавлева	Менеджер
26	Медведев	Курьер
33	Сорокин	Курьер

# NULL

- Сравнение: **IS [NOT] NULL**

```
SELECT EMPLOYEE_ID, LAST_NAME, POST  
FROM EMPLOYEES  
WHERE MANAGER_ID IS NULL;
```

EMP_ID NUMBER(10)	LAST_NAME VARCHAR2(100)	POST VARCHAR2(100)
1	Баранова	Курьер
10	Голубева	Курьер
12	Гусева	Повар
13	Давыдов	Повар
18	Журавлева	Менеджер
26	Медведев	Курьер
33	Сорокин	Курьер

В PL/SQL NULL и пустая строка  
эквивалентны



# NULL

- Сравнение: **IS [NOT] NULL**

```
SELECT EMPLOYEE_ID, LAST_NAME, POST  
FROM EMPLOYEES  
WHERE MANAGER_ID IS NULL;
```

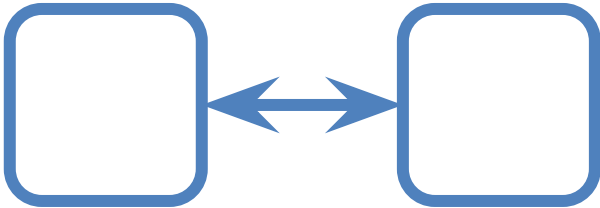
EMP_ID NUMBER(10)	LAST_NAME VARCHAR2(100)	POST VARCHAR2(100)
1	Баранова	Курьер
10	Голубева	Курьер
12	Гусева	Повар
13	Давыдов	Повар
18	Журавлева	Менеджер
26	Медведев	Курьер
33	Сорокин	Курьер

В PL/SQL NULL и пустая строка  
эквивалентны

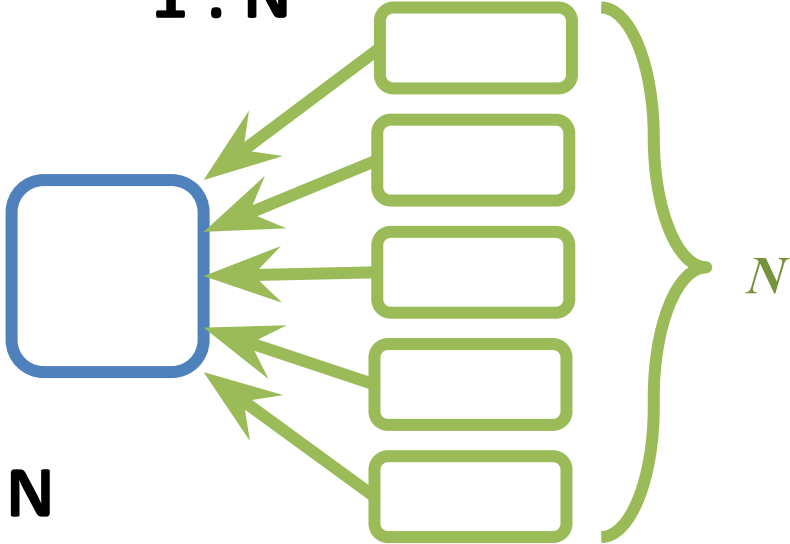
В PL/SQL результат сравнения с NULL через другие  
операции:  
**UNKNOWN**

# СВЯЗИ

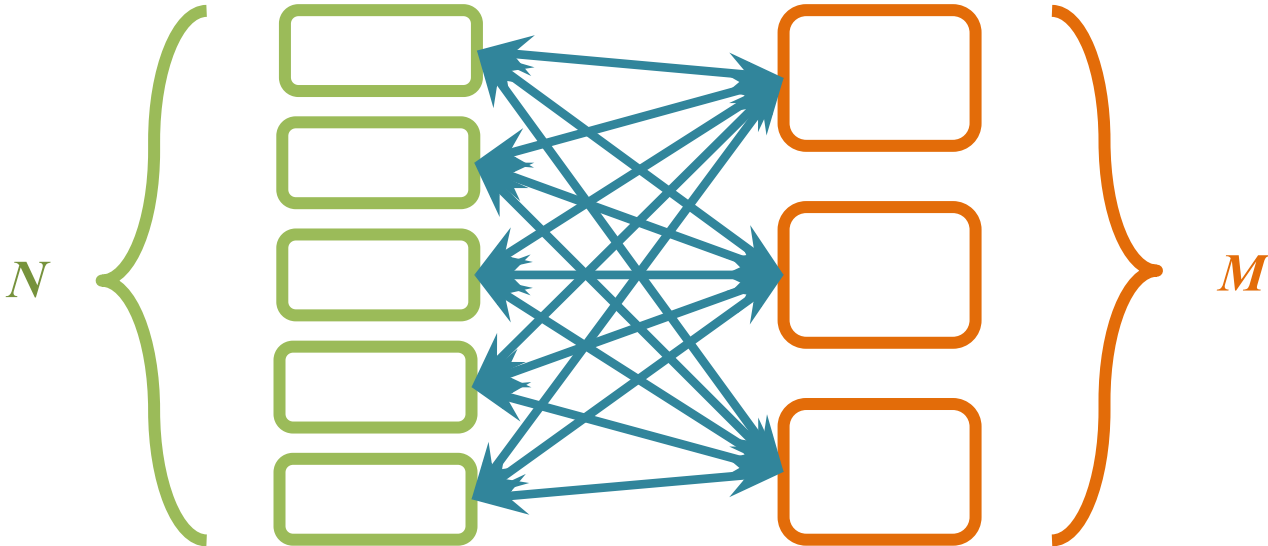
**1 : 1**



**1 : N**



**M : N**



# JOIN

1
2
3

**CROSS  
JOIN**

A
B
C

=

1	A
1	B
1	C
2	A
2	B
2	C
3	A
3	B
3	C

# JOIN

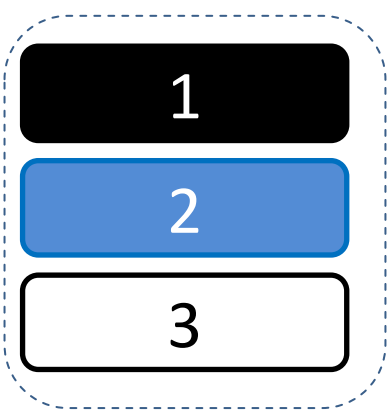
1
2
3

**CROSS  
JOIN**

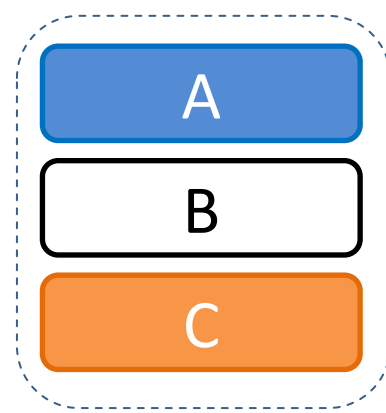
A
B
C

=

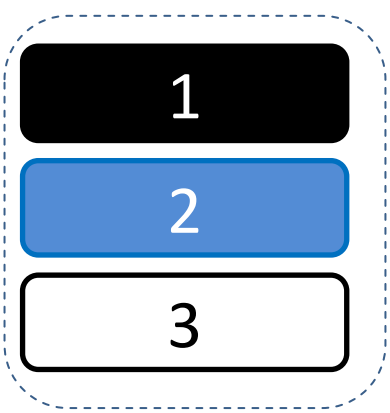
1	A
1	B
1	C
2	A
2	B
2	C
3	A
3	B
3	C



$T_1$

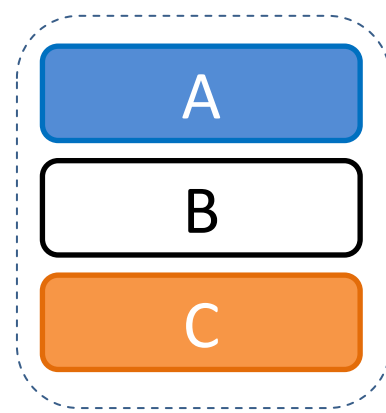


$T_2$

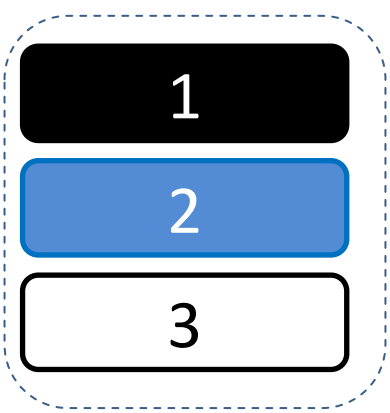


$T_1$

INNER JOIN



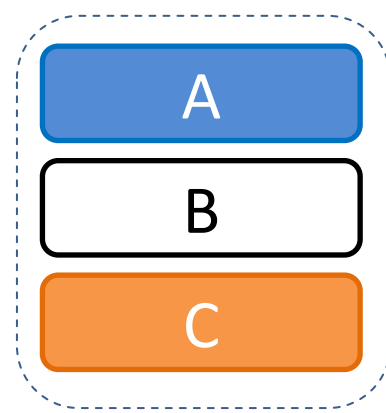
$T_2$



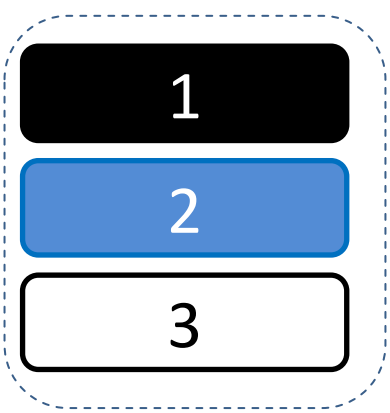
$T_1$

# INNER JOIN

1	A
1	B
1	C
2	A
2	B
2	C
3	A
3	B
3	C

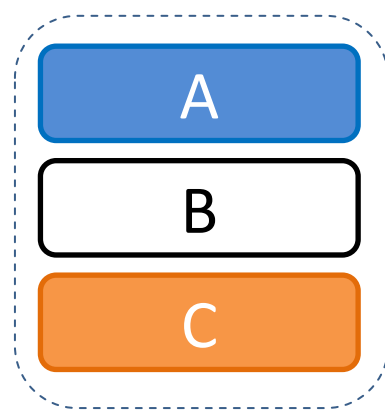
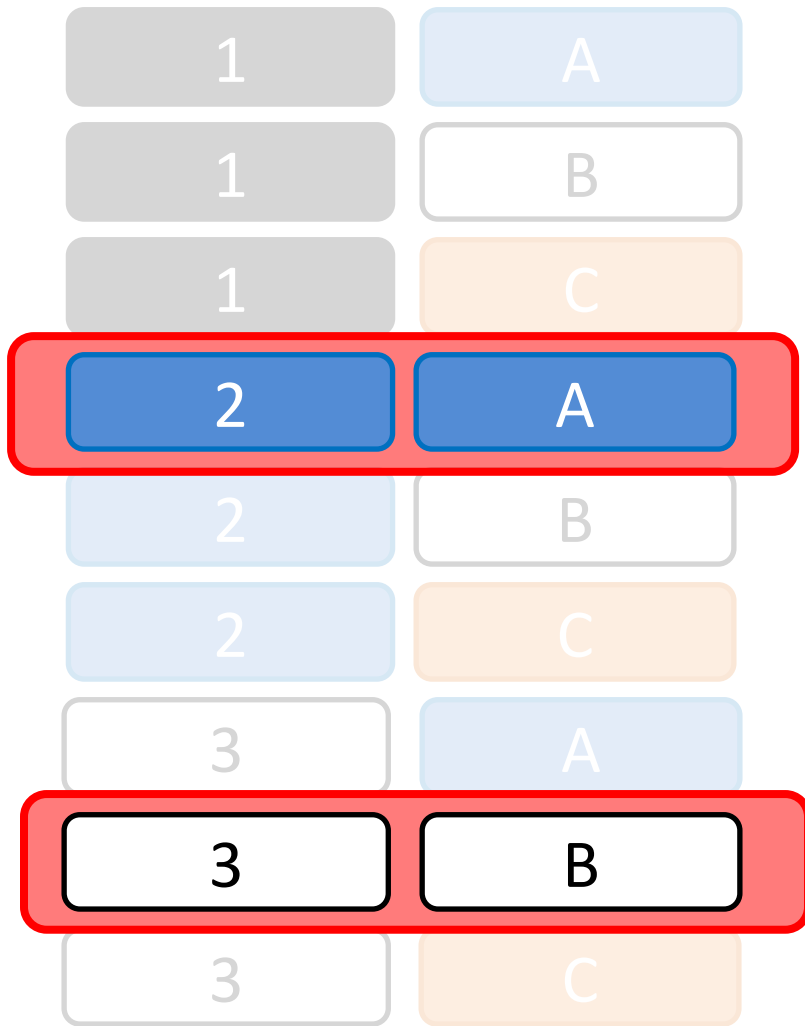


$T_2$



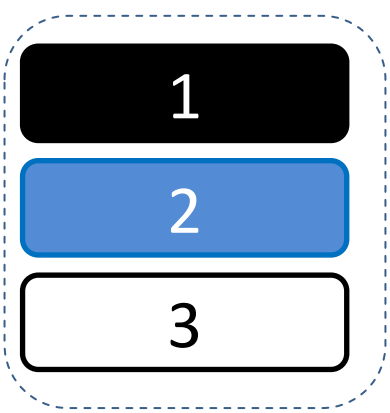
$T_1$

# INNER JOIN



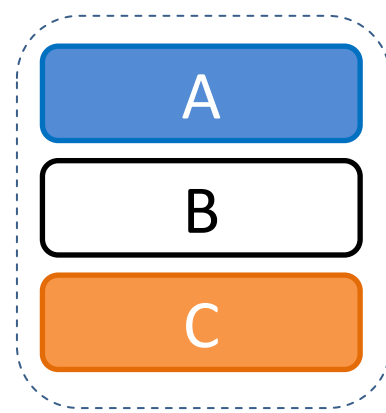
$T_2$



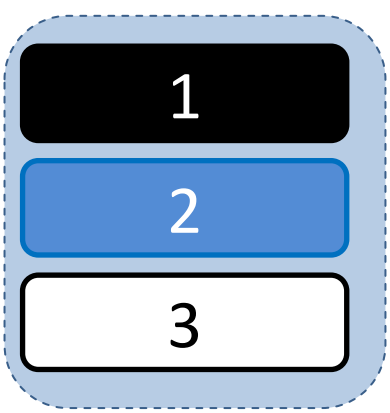


$T_1$

LEFT JOIN

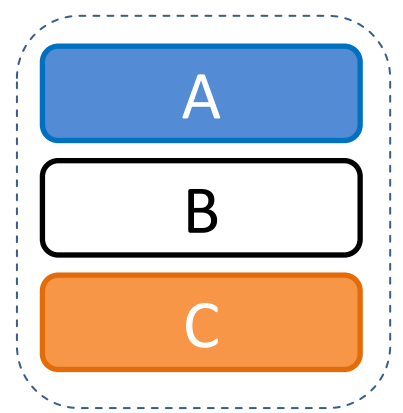


$T_2$



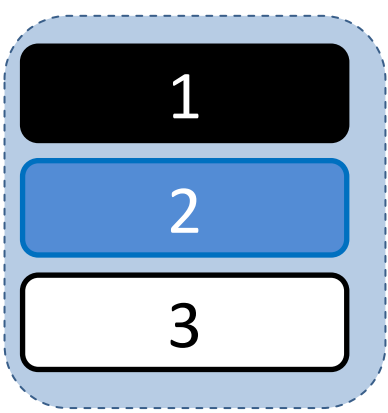
$T_1$

LEFT JOIN

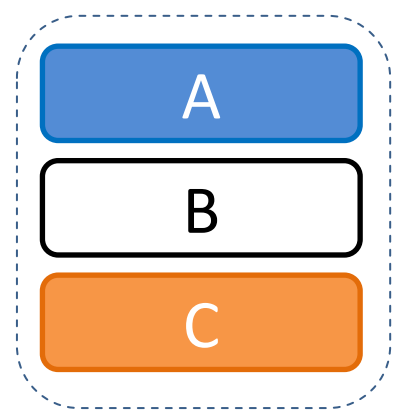


$T_2$

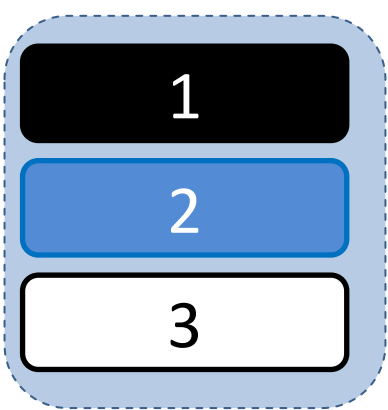
# LEFT JOIN



$T_1$

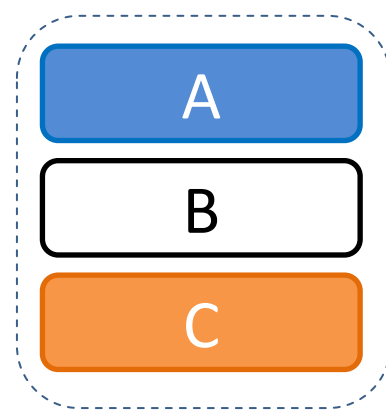
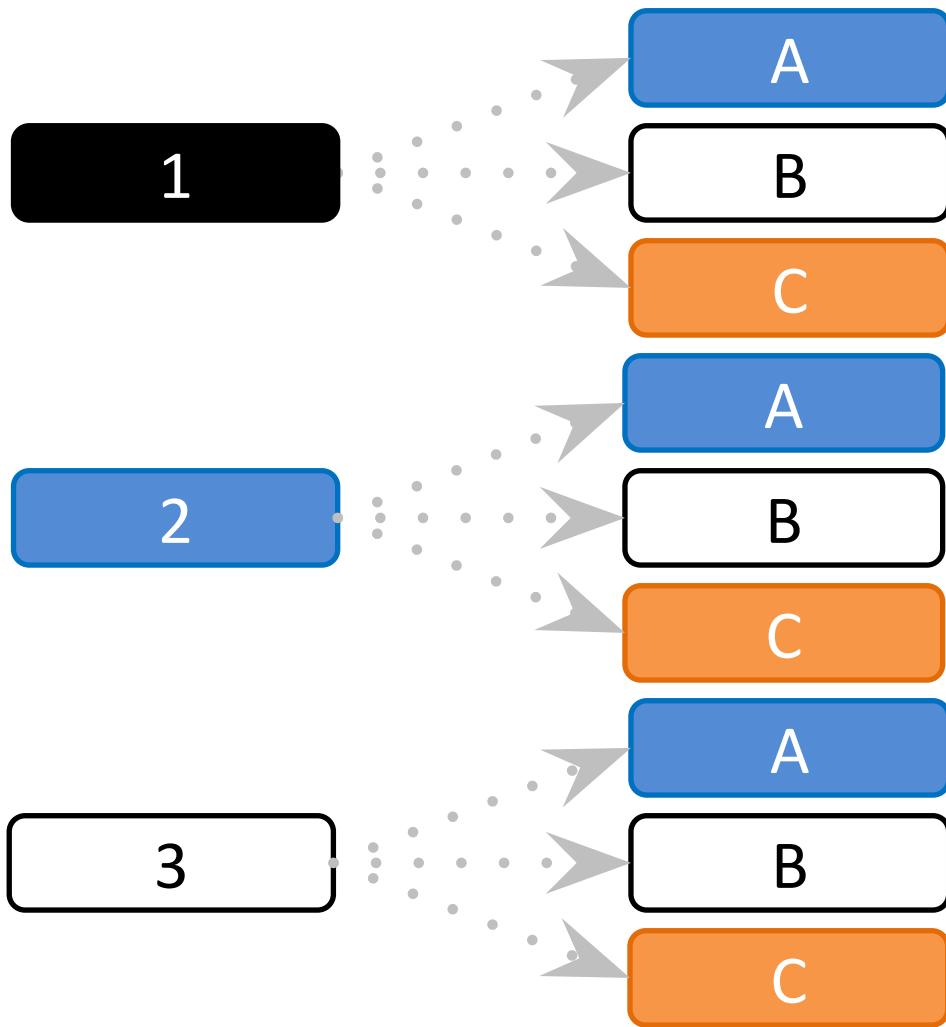


$T_2$

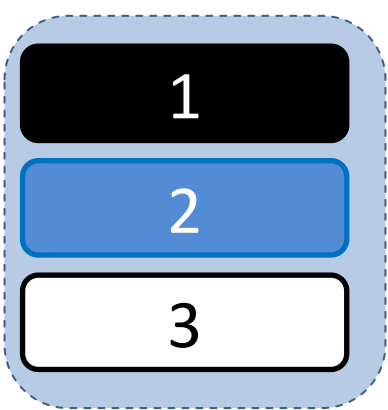


$T_1$

# LEFT JOIN

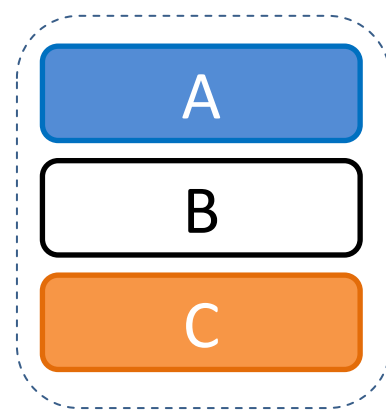
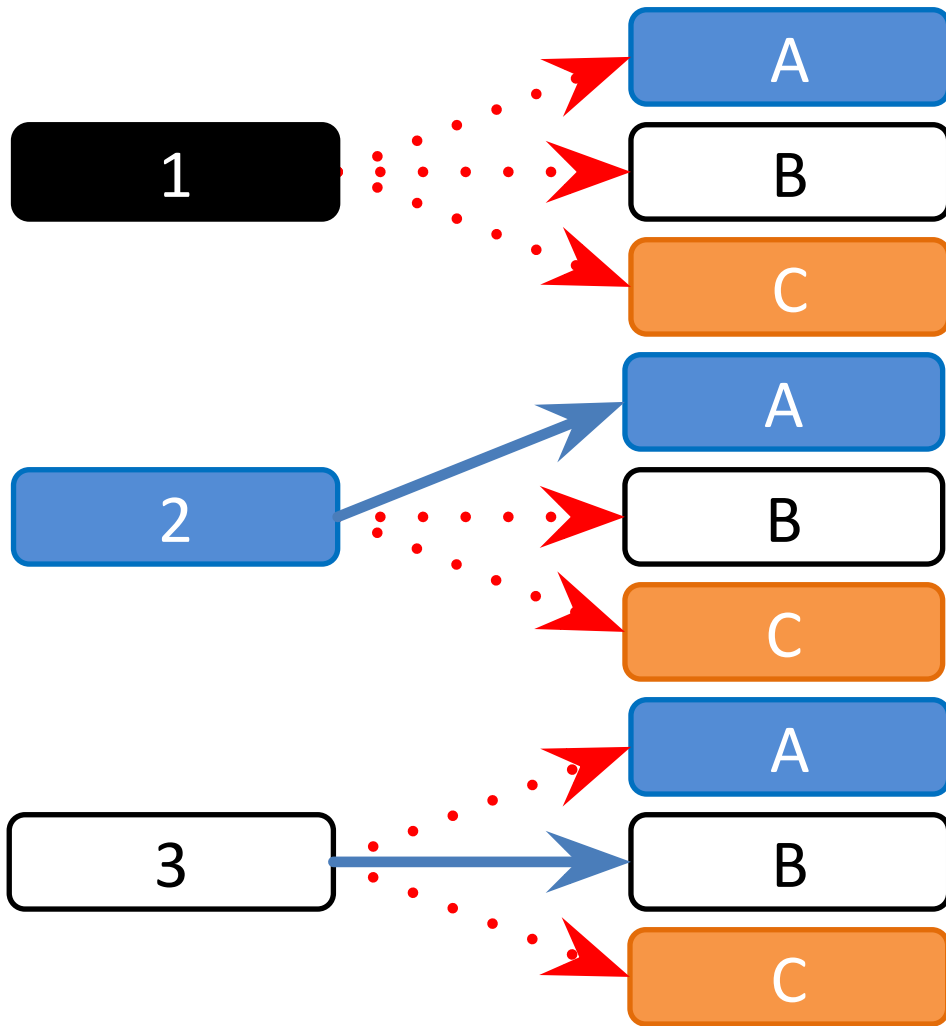


$T_2$

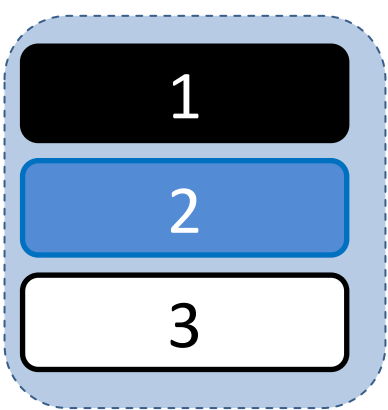


$T_1$

# LEFT JOIN

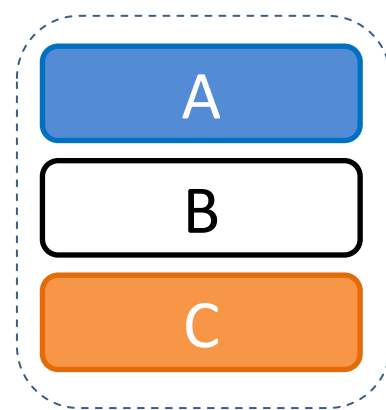
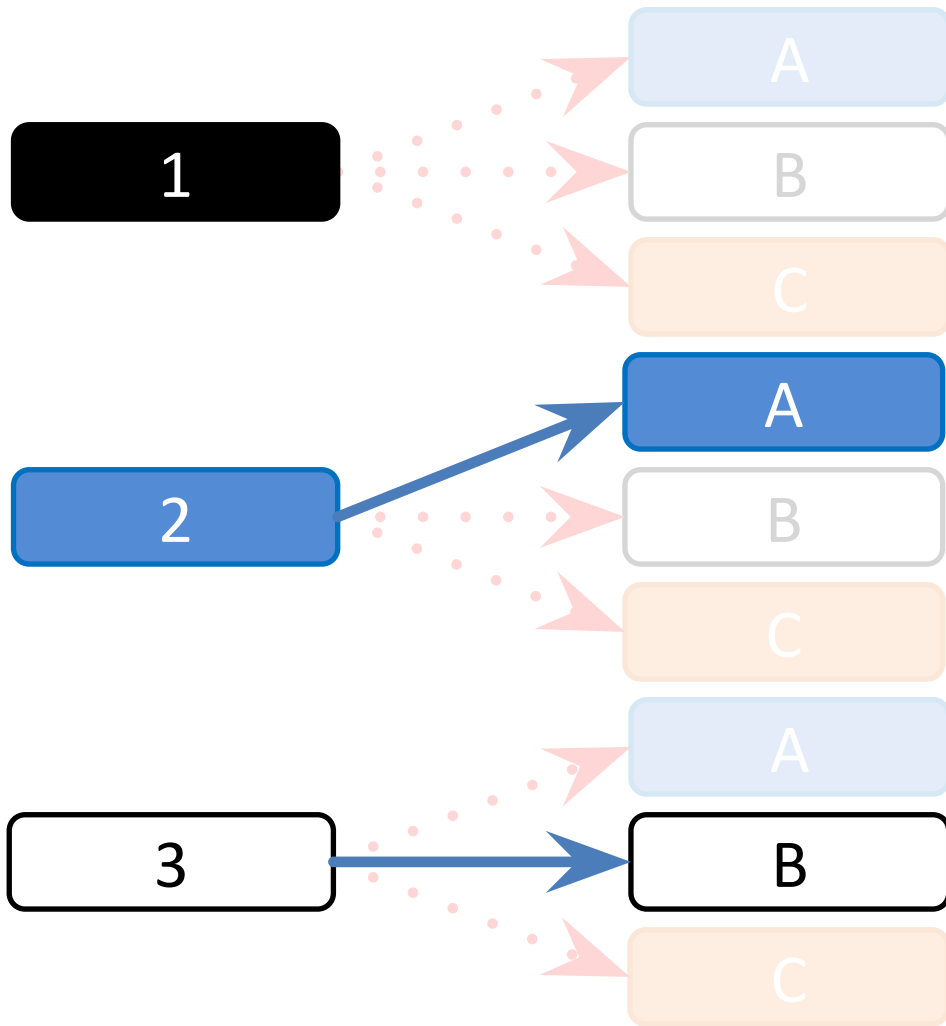


$T_2$



$T_1$

# LEFT JOIN



$T_2$

# RIGHT JOIN

1

2

3

$T_1$

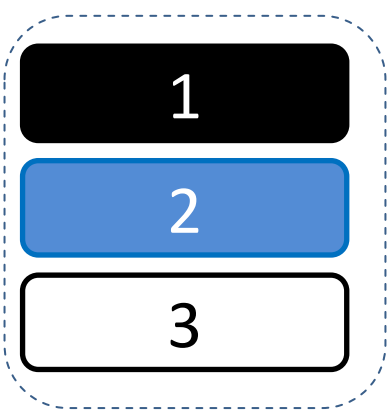
A

B

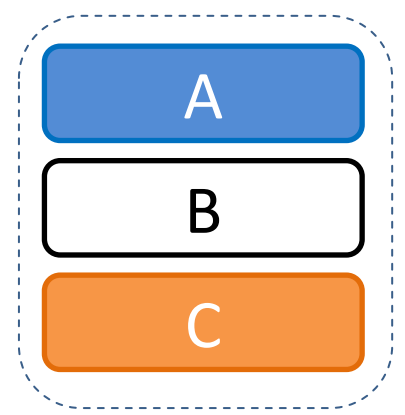
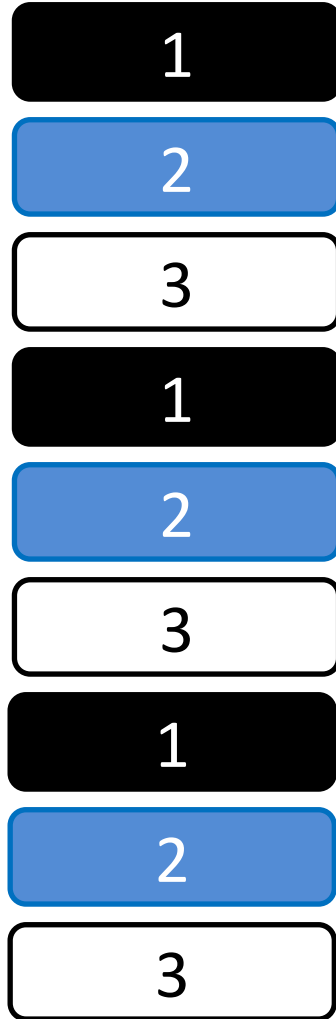
C

$T_2$

# RIGHT JOIN

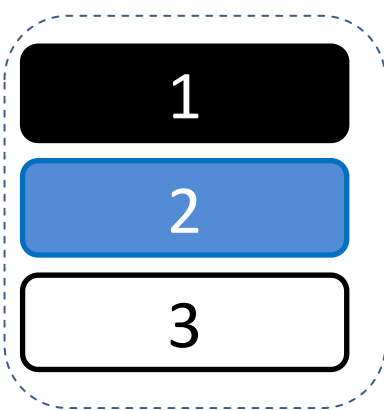


$T_1$



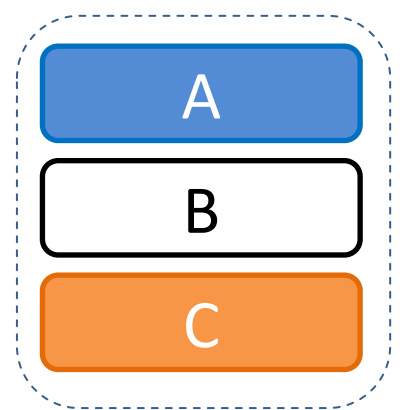
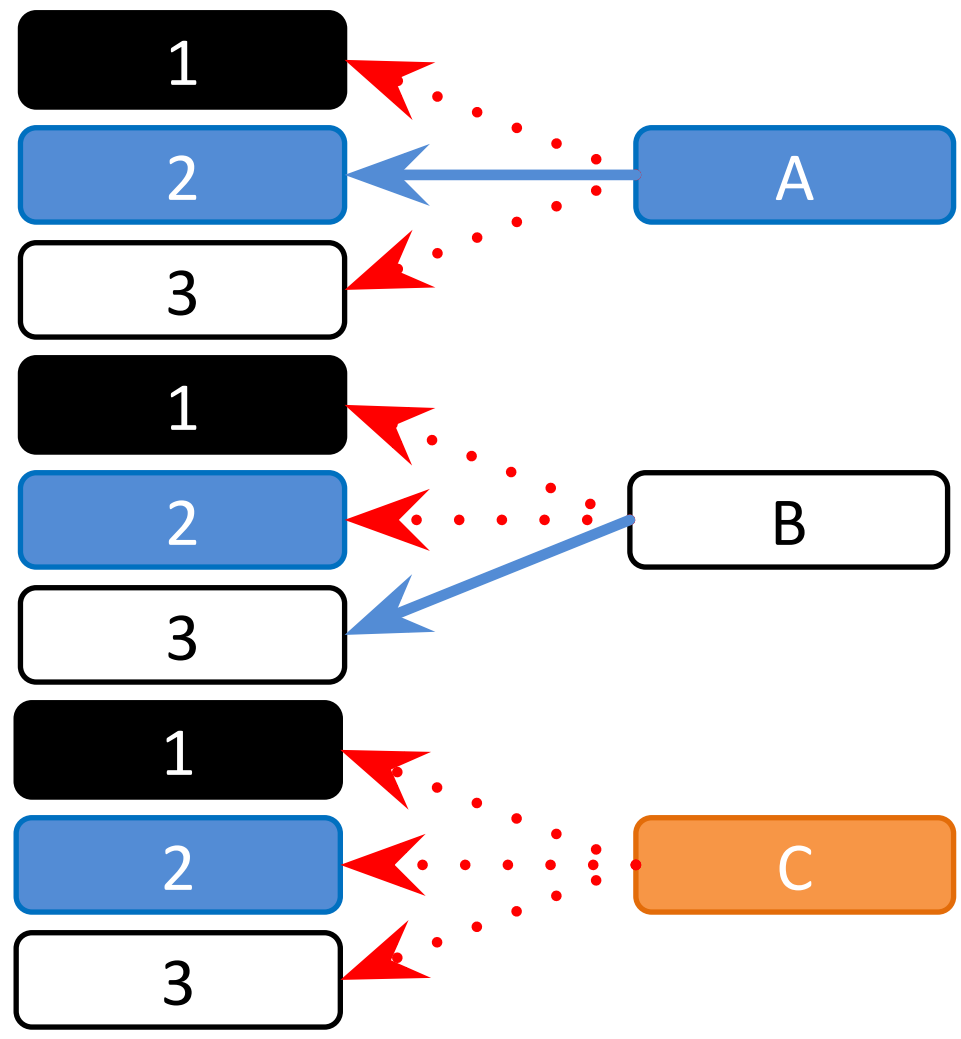
$T_2$



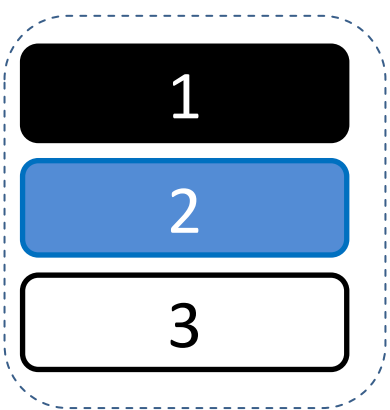


$T_1$

# RIGHT JOIN

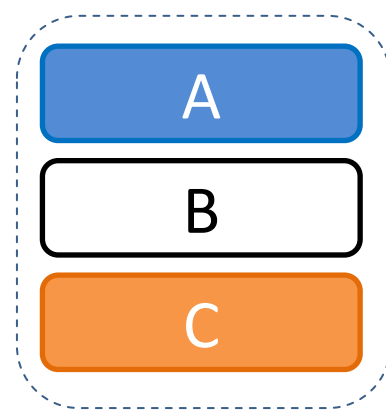
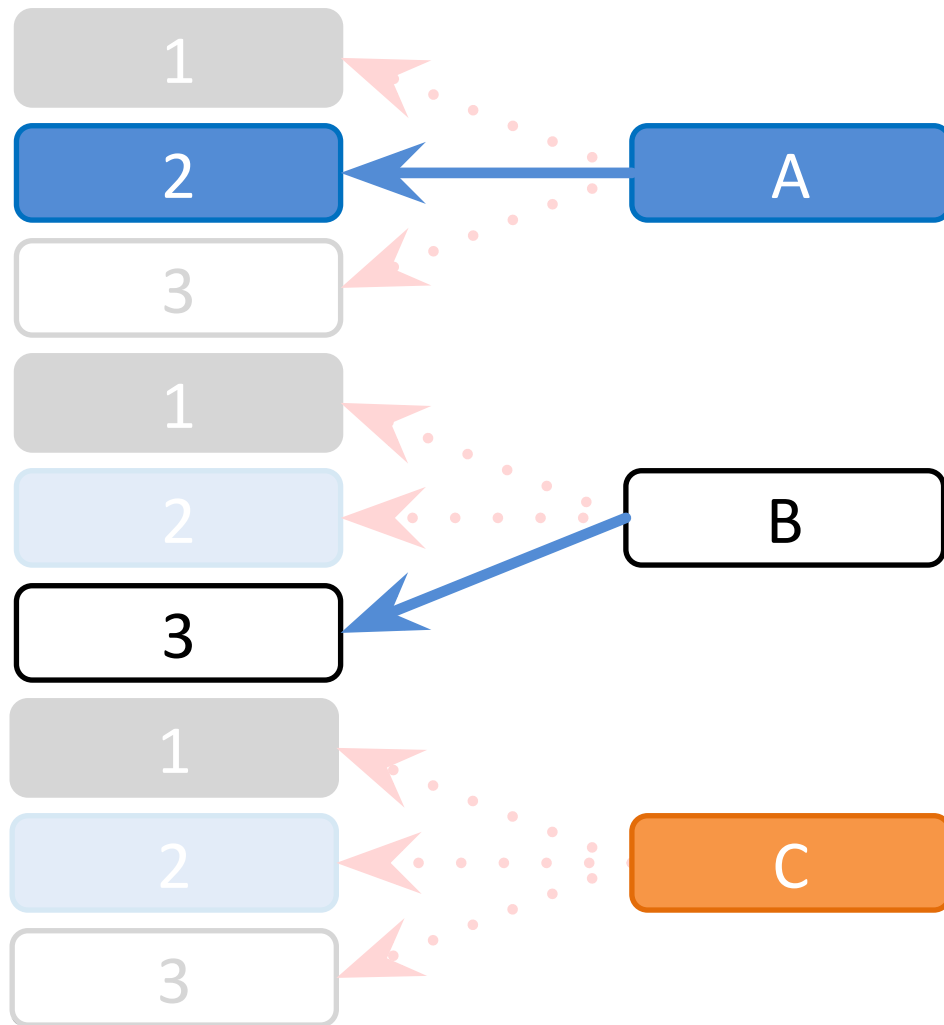


$T_2$



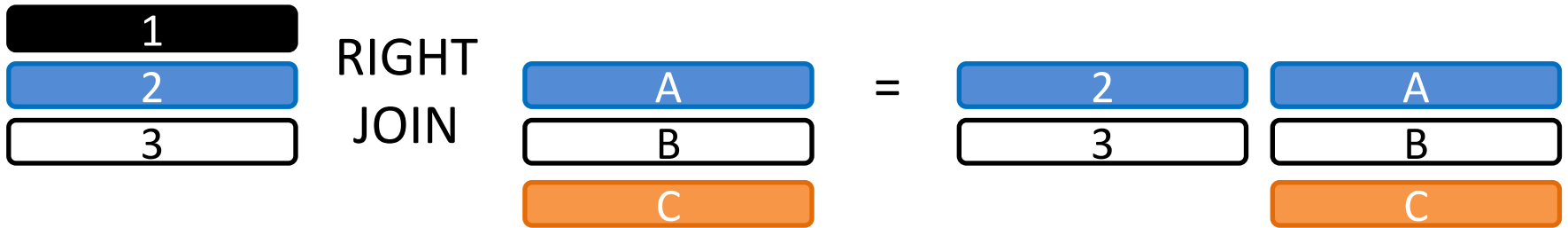
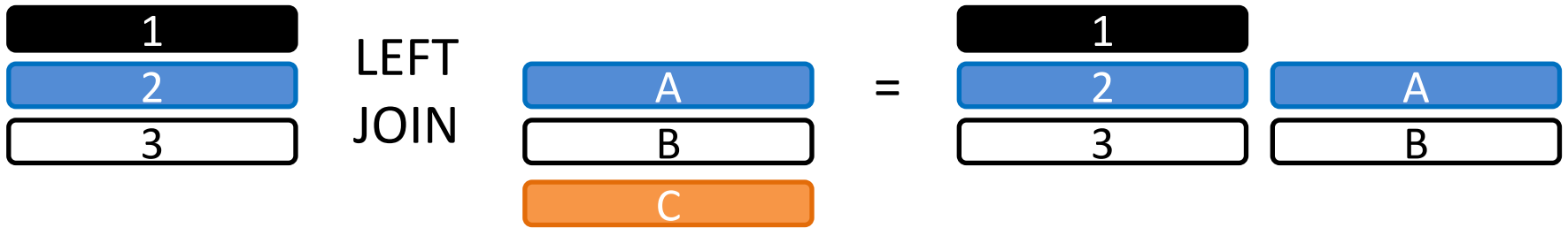
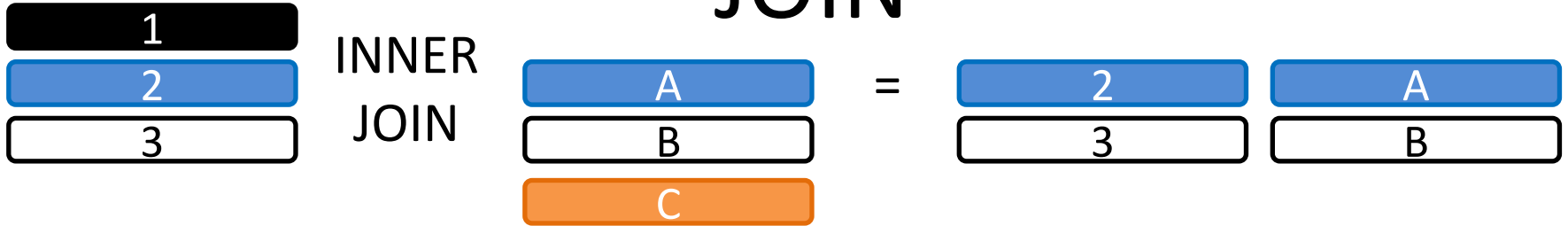
$T_1$

# RIGHT JOIN

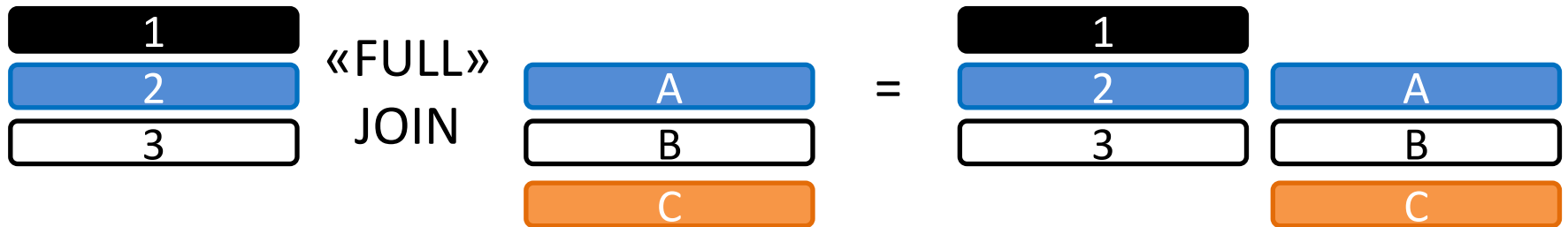
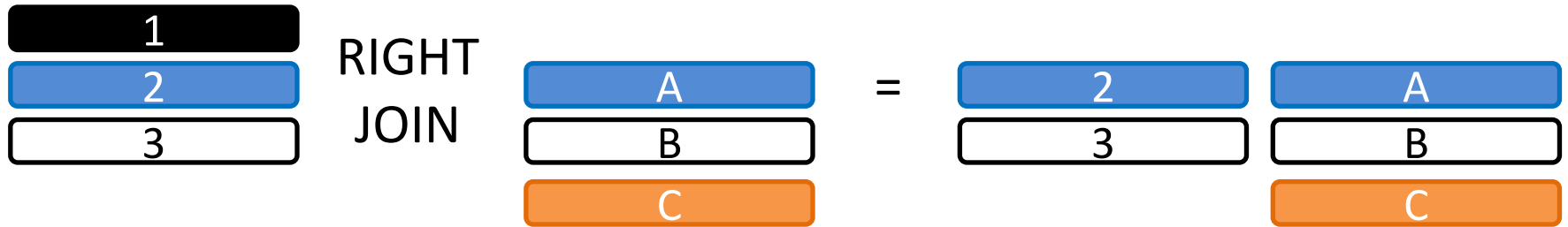
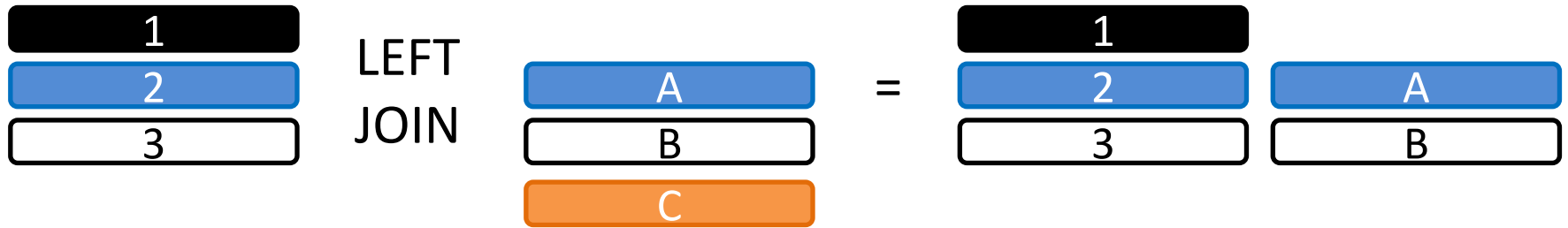
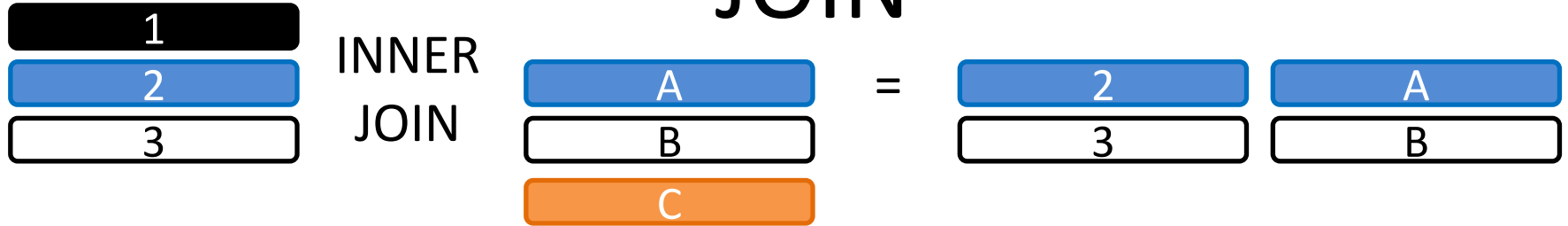


$T_2$

# JOIN



# JOIN



1

2

3

$T_1$

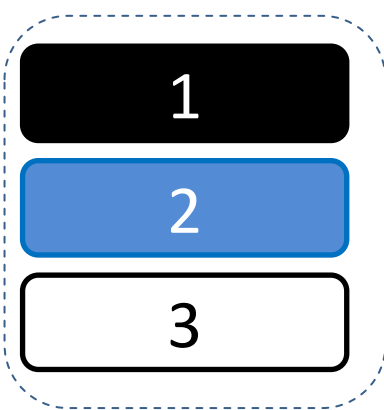
INNER JOIN

A

B

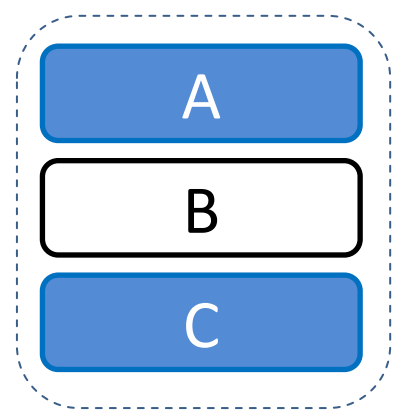
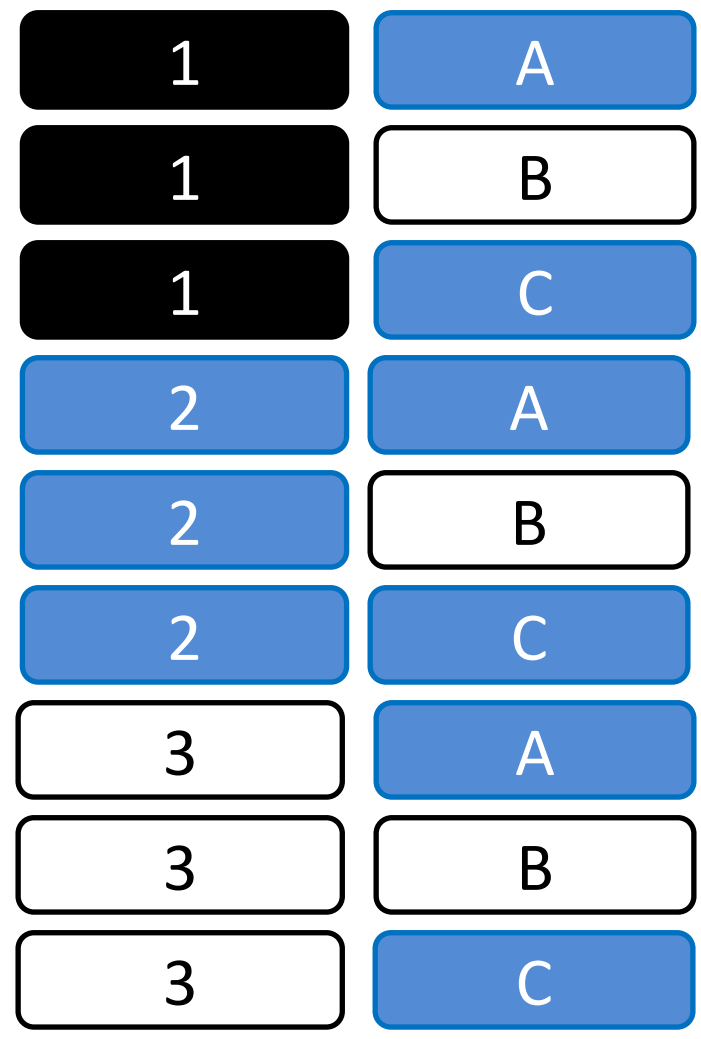
C

$T_2$

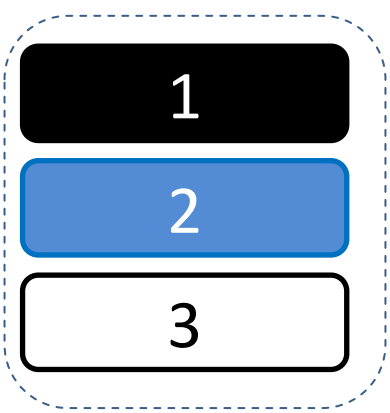


$T_1$

# INNER JOIN



$T_2$

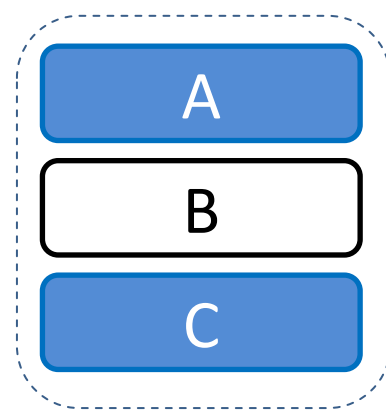


$T_1$

# INNER JOIN

Cartesian product of  $T_1$  and  $T_2$  with 9 rows and 2 columns:

1	A
1	B
1	C
2	A
2	B
2	C
3	A
3	B
3	C



$T_2$

1

2

3

INNER

JOIN

A

B

C

D



1
2
3

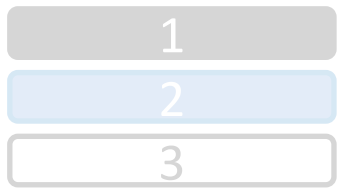
INNER  
JOIN

A
B
C
D

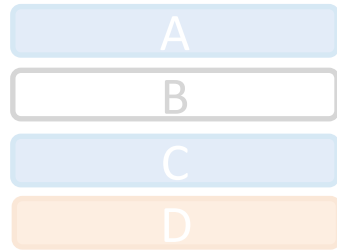
=

2
2
3

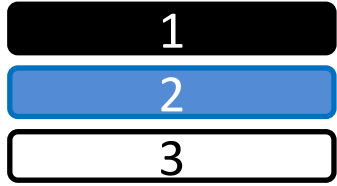
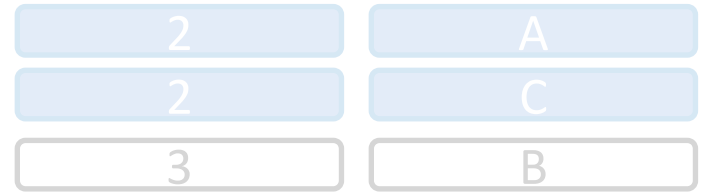
A
C
B



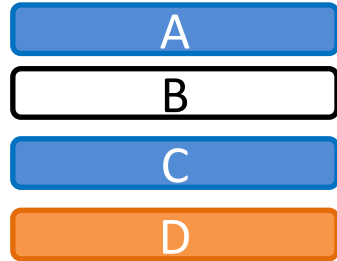
INNER  
JOIN

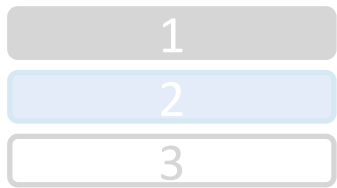


=

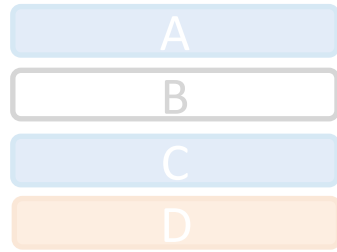


LEFT  
JOIN

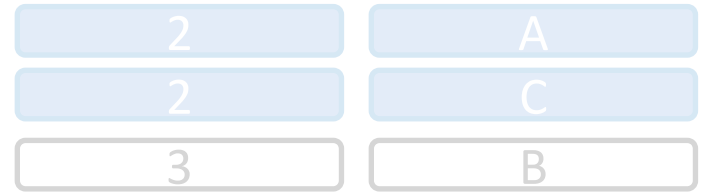




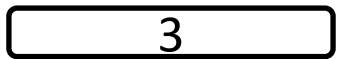
INNER  
JOIN

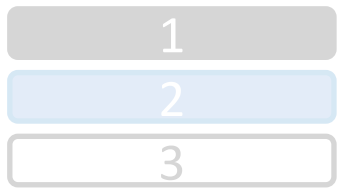


=

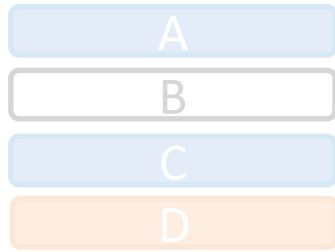


LEFT JOIN

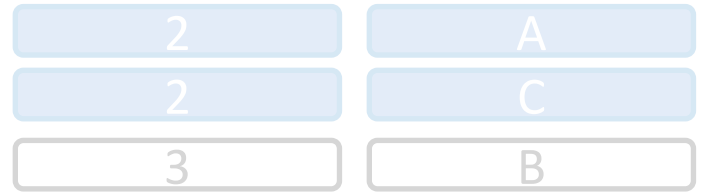




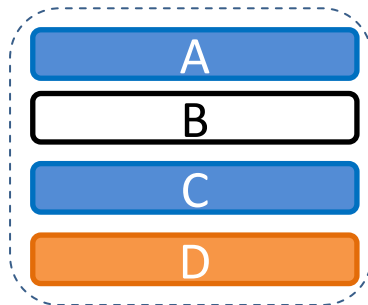
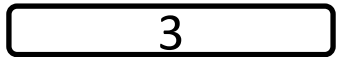
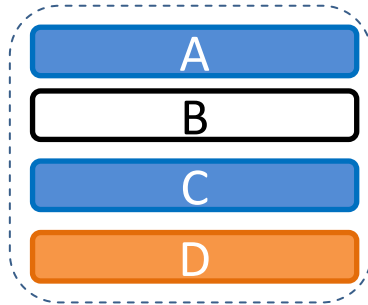
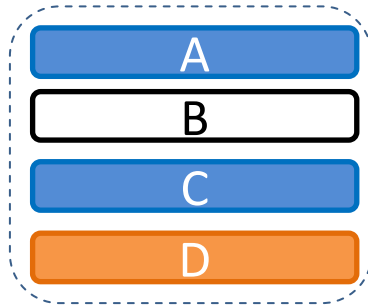
INNER  
JOIN

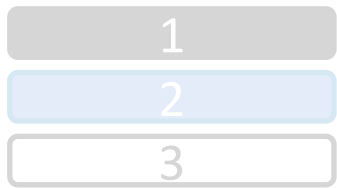


=

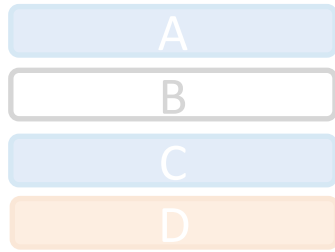


LEFT JOIN

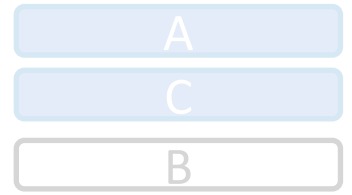
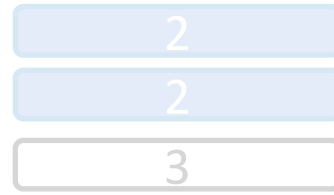




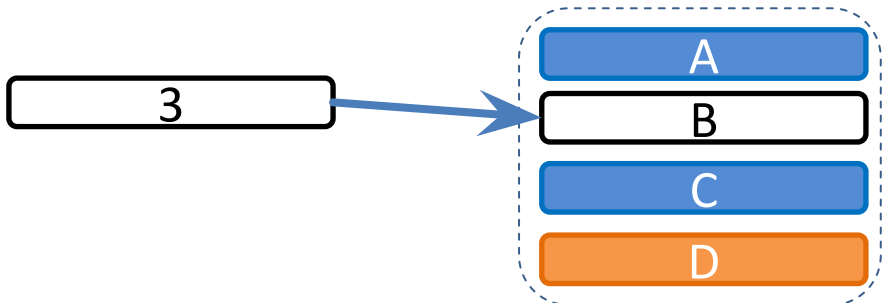
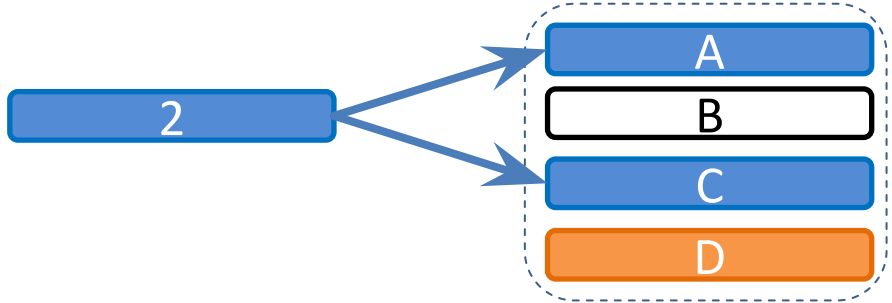
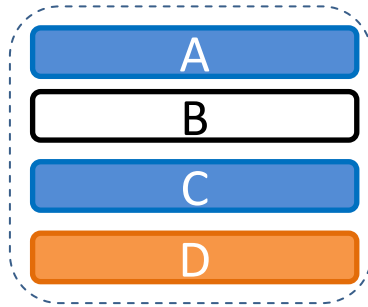
INNER  
JOIN

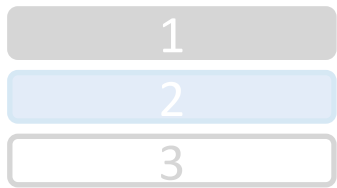


=

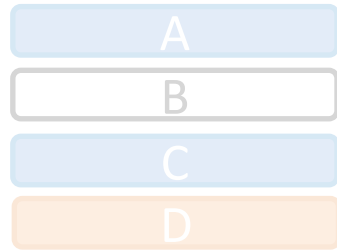


LEFT JOIN

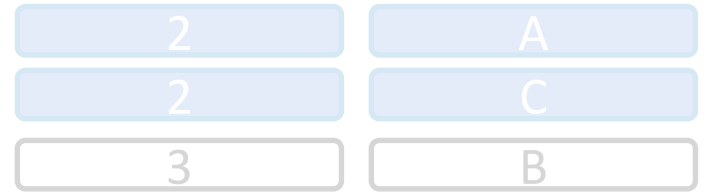




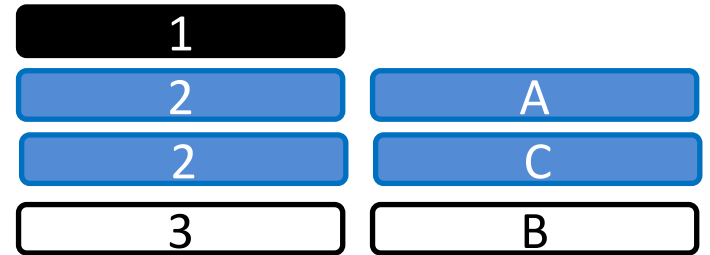
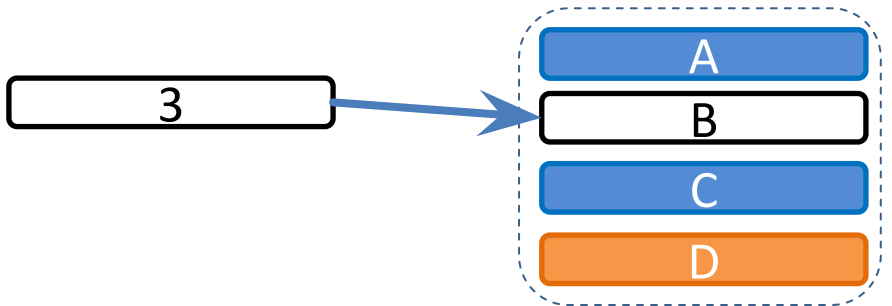
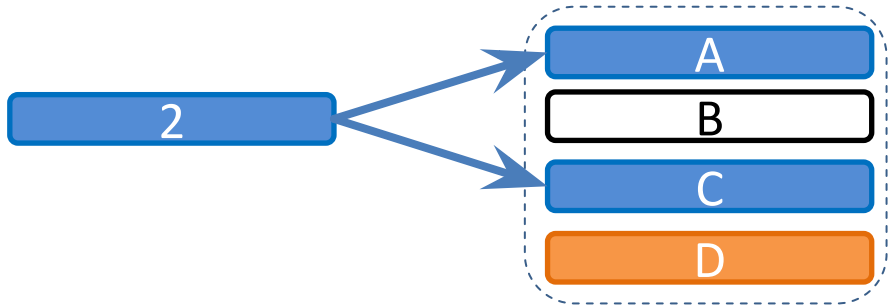
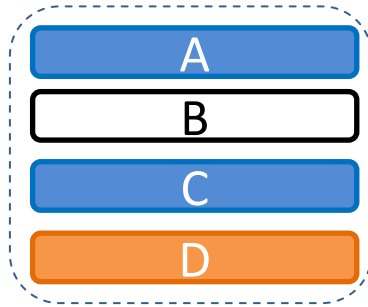
INNER  
JOIN

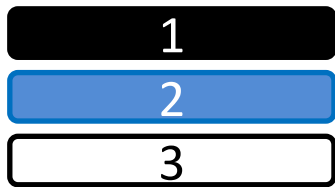


=

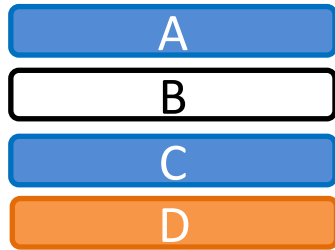


LEFT JOIN

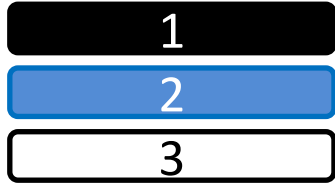
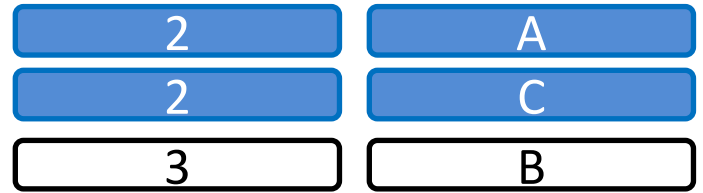




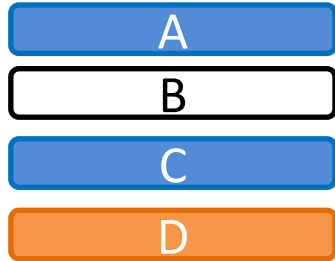
INNER  
JOIN



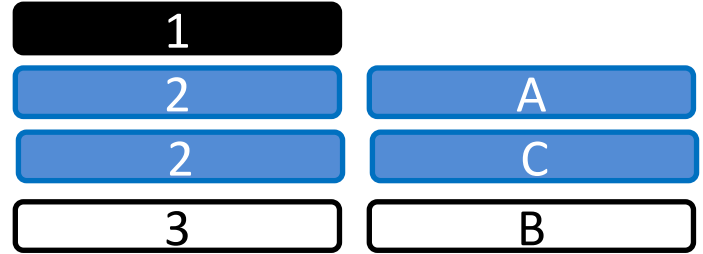
=

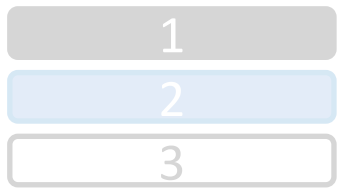


LEFT  
JOIN

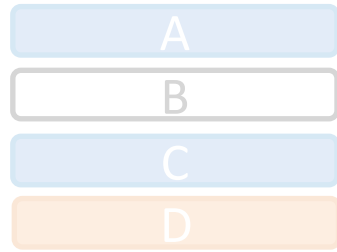


=

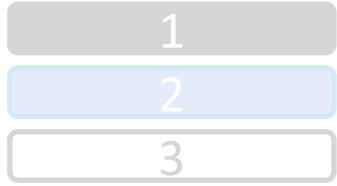
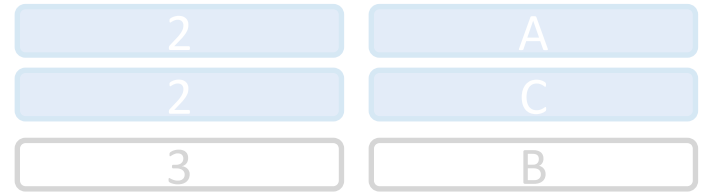




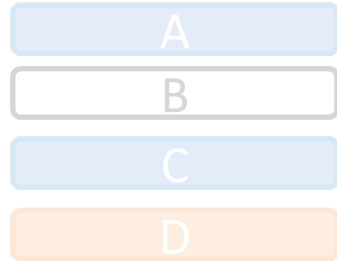
INNER  
JOIN



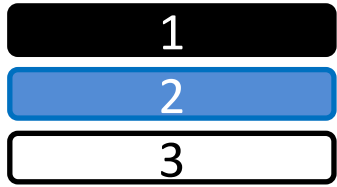
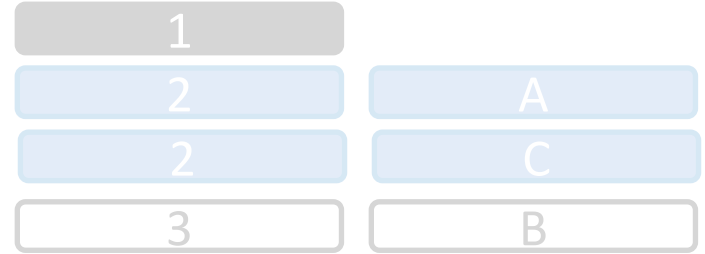
=



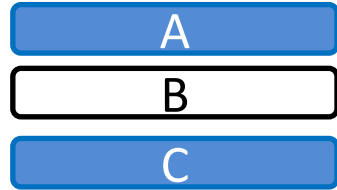
LEFT  
JOIN



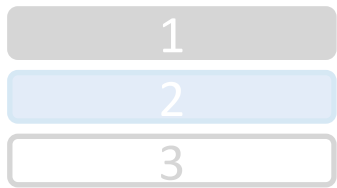
=



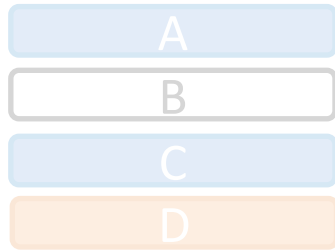
RIGHT  
JOIN



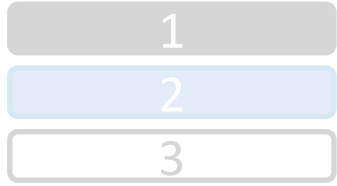
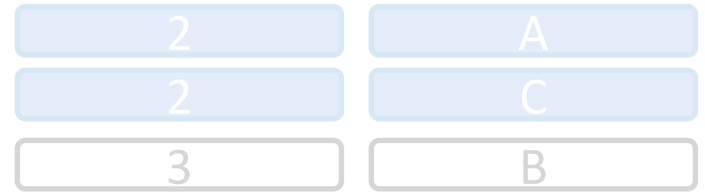




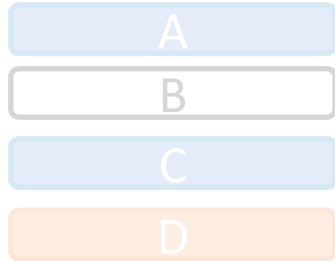
INNER  
JOIN



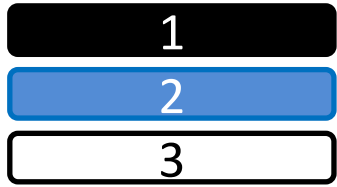
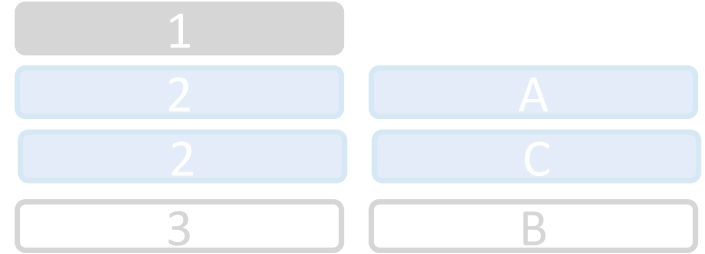
=



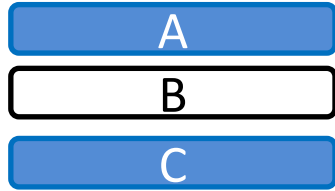
LEFT  
JOIN



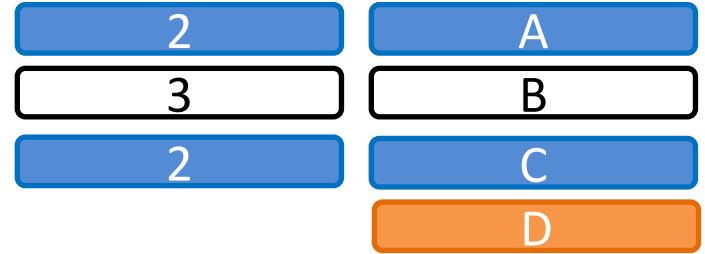
=

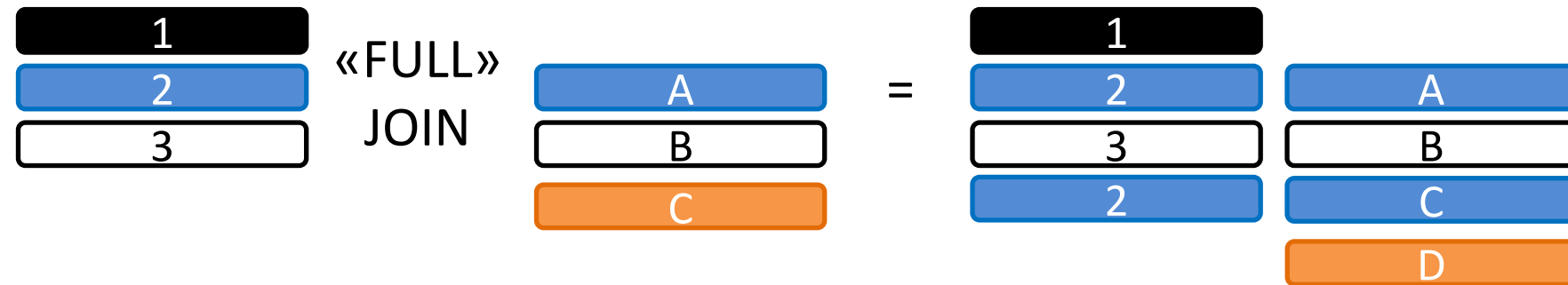
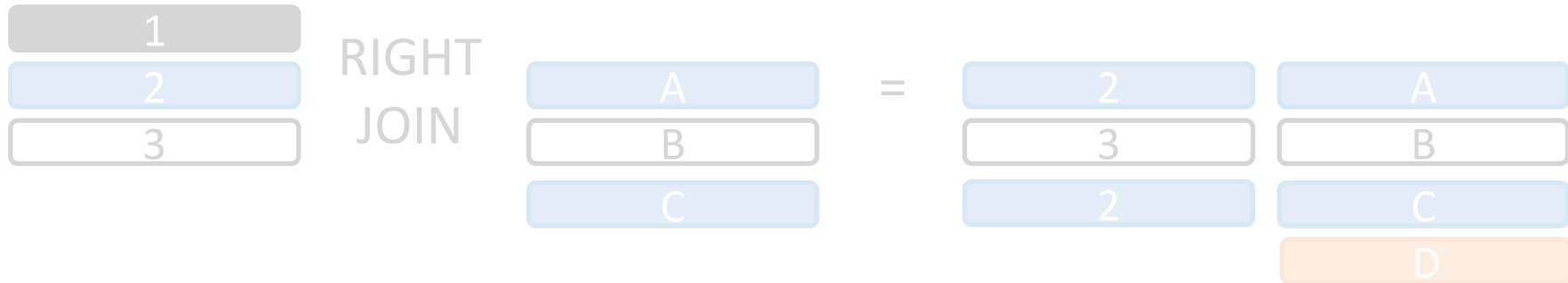
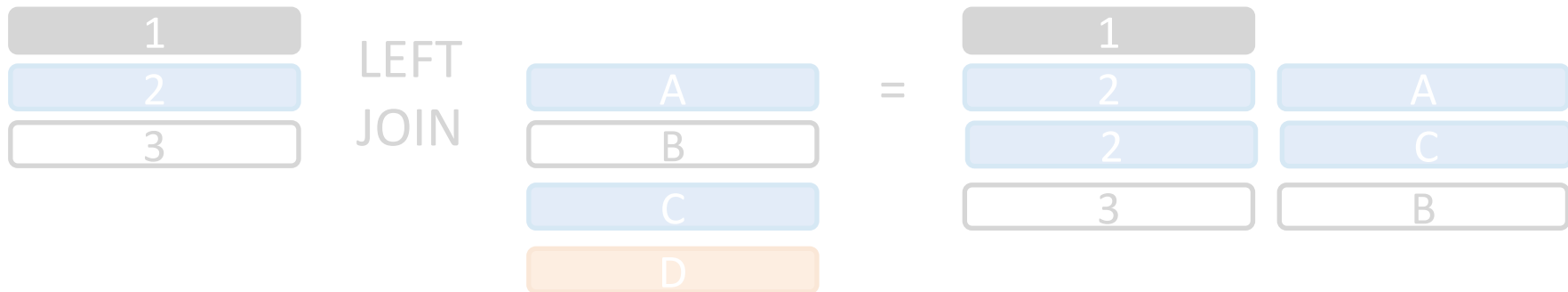
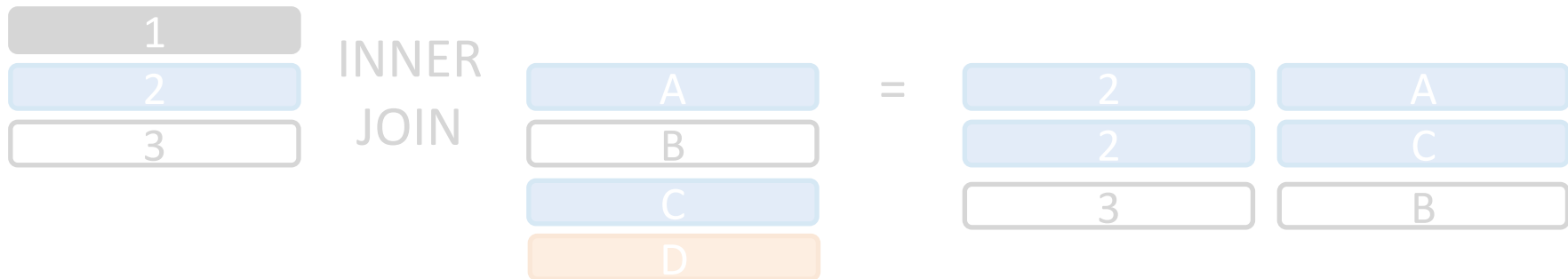


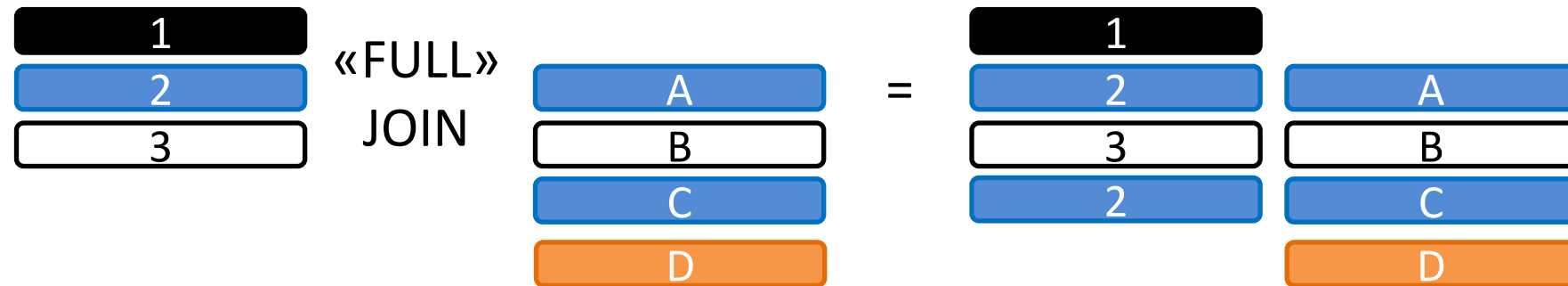
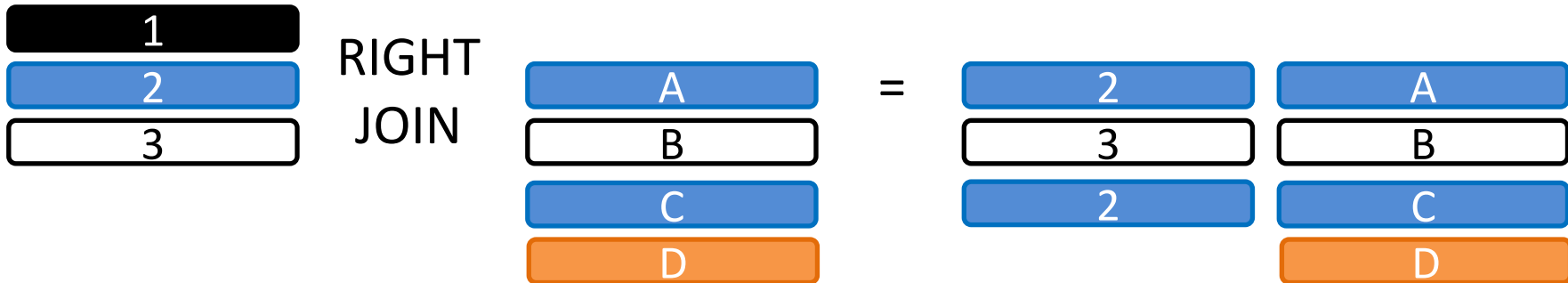
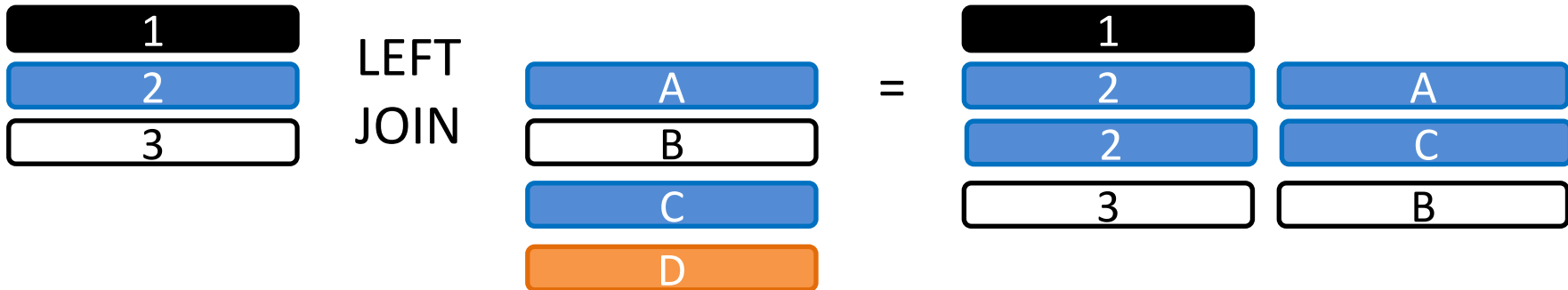
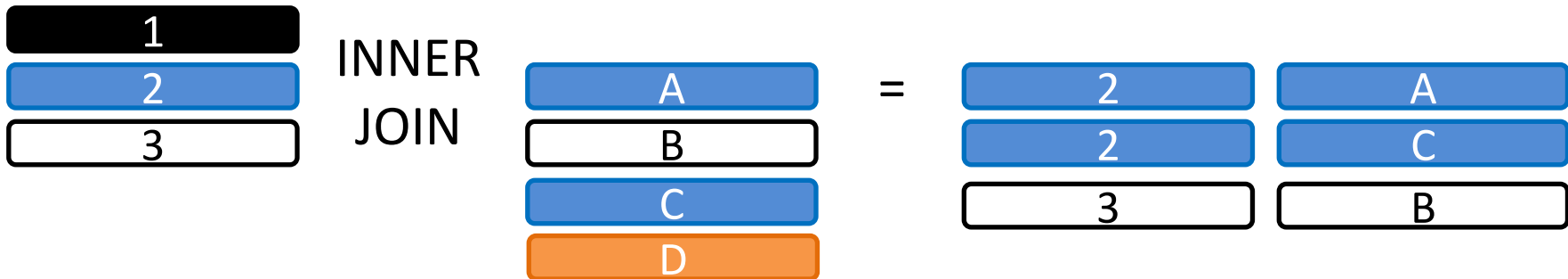
RIGHT  
JOIN



=







# Синтаксис JOIN

```
SELECT [ALL | DISTINCT] <имя_столбца |  
перечень_столбцов>  
FROM <имя_таблицы>, ...  
[ WHERE <условие> ]  
[ GROUP BY <имя_столбца>,... ]  
[ HAVING <условие> ]  
[ ORDER BY <имя_столбца>,... [ASC | DESC],...  
]
```

# Синтаксис JOIN

```
SELECT [ALL | DISTINCT] <имя_столбца |  
перечень_столбцов>  
FROM <имя_таблицы>, ...  
    JOIN <имя_таблицы_2> ON <условие>  
[ WHERE <условие> ]  
[ GROUP BY <имя_столбца>,... ]  
[ HAVING <условие> ]  
[ ORDER BY <имя_столбца>,... [ASC | DESC],...  
]
```

```
SELECT CATEGORY_NAME, PRODUCT_NAME, PRICE
FROM PRODUCTS
```

```
INNER JOIN CATEGORIES ON CATEGORIES. CATEGORY_ID
= PRODUCTS. CATEGORY_ID;
```

CATEGORY_ID NUMBER(10)	PRODUCT_NAME VARCHAR2(100)	PRICE NUMBER(10, 4)
1	Пепперони	502
1	Гавайская	510
(null)	Карбонара	490
1	Марио	450
(null)	Сицилия	510
3	Медовый	150
3	Тирамису	150
3	Наполеон	140

CATEGORY_ID NUMBER(10)	CATEGORY_NAME VARCHAR2(100)
1	Пицца
2	Напитки
3	Десерты

**SELECT** CATEGORY\_NAME, PRODUCT\_NAME, PRICE  
**FROM** PRODUCTS

**INNER JOIN** CATEGORIES **ON** CATEGORIES. **CATEGORY\_ID**  
**=** PRODUCTS. **CATEGORY\_ID**;

CATEGORY_ID NUMBER(10)	PRODUCT_NAME VARCHAR2(100)	PRICE NUMBER(10, 4)
1	Пепперони	502
1	Гавайская	510
(null)	Карбонара	490
1	Марио	450
(null)	Сицилия	510
3	Медовый	150
3	Тирамису	150
3	Наполеон	140

CATEGORY_NAME VARCHAR2(100)	PRODUCT_NAME VARCHAR2(100)	PRICE NUMBER(10, 4)
Пицца	Пепперони	502
Пицца	Гавайская	510
Пицца	Марио	450
Десерты	Наполеон	140
Десерты	Медовый	150
Десерты	Тирамису	150

CATEGORY_ID NUMBER(10)	CATEGORY_NAME VARCHAR2(100)
1	Пицца
2	Напитки
3	Десерты

**SELECT** CATEGORY\_NAME, PRODUCT\_NAME, PRICE  
**FROM** PRODUCTS

**LEFT JOIN** CATEGORIES **ON** CATEGORIES. **CATEGORY\_ID**

**=** PRODUCTS. **CATEGORY\_ID**;

CATEGORY_ID NUMBER(10)	PRODUCT_NAME VARCHAR2(100)	PRICE NUMBER(10, 4)
1	Пепперони	502
1	Гавайская	510
(null)	Карбонара	490
1	Марио	450
(null)	Сицилия	510
3	Медовый	150
3	Тирамису	150
3	Наполеон	140

CATEGORY_NAME VARCHAR2(100)	PRODUCT_NAME VARCHAR2(100)	PRICE NUMBER(10, 4)
Пицца	Марио	450
Пицца	Гавайская	510
Пицца	Пепперони	502
Десерты	Наполеон	140
Десерты	Тирамису	150
Десерты	Медовый	150
(null)	Сицилия	510
(null)	Карбонара	490

CATEGORY_ID NUMBER(10)	CATEGORY_NAME VARCHAR2(100)
1	Пицца
2	Напитки
3	Десерты



**SELECT** CATEGORY\_NAME, PRODUCT\_NAME, PRICE  
**FROM** PRODUCTS

**RIGHT JOIN** CATEGORIES **ON** CATEGORIES. **CATEGORY\_ID**  
**=** PRODUCTS. **CATEGORY\_ID**;

CATEGORY_ID NUMBER(10)	PRODUCT_NAME VARCHAR2(100)	PRICE NUMBER(10, 4)
1	Пепперони	502
1	Гавайская	510
(null)	Карбонара	490
1	Марио	450
(null)	Сицилия	510
3	Медовый	150
3	Тирамису	150
3	Наполеон	140

CATEGORY_NAME VARCHAR2(100)	PRODUCT_NAME VARCHAR2(100)	PRICE NUMBER(10, 4)
Пицца	Пепперони	502
Пицца	Гавайская	510
Пицца	Марио	450
Десерты	Наполеон	140
Десерты	Медовый	150
Десерты	Тирамису	150

CATEGORY_ID NUMBER(10)	CATEGORY_NAME VARCHAR2(100)
1	Пицца
2	Напитки
3	Десерты

**SELECT** CATEGORY\_NAME, PRODUCT\_NAME, PRICE  
**FROM** PRODUCTS

**FULL JOIN** CATEGORIES **ON** CATEGORIES. **CATEGORY\_ID**

**=** PRODUCTS. **CATEGORY\_ID**;

CATEGORY_ID NUMBER(10)	PRODUCT_NAME VARCHAR2(100)	PRICE NUMBER(10, 4)
1	Пепперони	502
1	Гавайская	510
(null)	Карбонара	490
1	Марио	450
(null)	Сицилия	510
3	Медовый	150
3	Тирамису	150
3	Наполеон	140

CATEGORY_NAME VARCHAR2(100)	PRODUCT_NAME VARCHAR2(100)	PRICE NUMBER(10, 4)
Пицца	Марио	450
Пицца	Гавайская	510
Пицца	Пепперони	502
Десерты	Наполеон	140
Десерты	Тирамису	150
Десерты	Медовый	150
(null)	Сицилия	510
(null)	Карбонара	490

CATEGORY_ID NUMBER(10)	CATEGORY_NAME VARCHAR2(100)
1	Пицца
2	Напитки
3	Десерты

**SELECT CATEGORY\_NAME, PRODUCT\_NAME, PRICE**

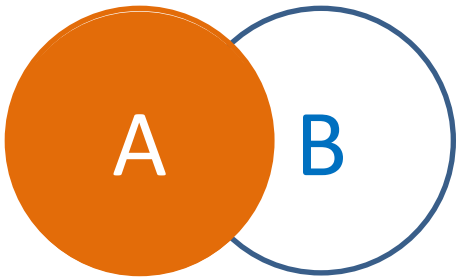
**FROM PRODUCTS**

**CROSS JOIN CATEGORIES;**

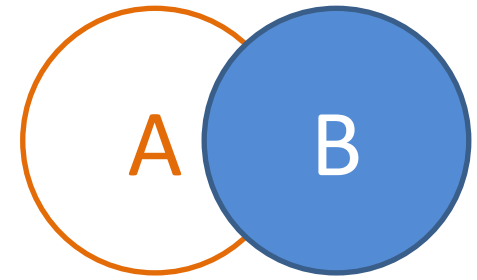
CATEGORY_ID NUMBER(10)	PRODUCT_NAME VARCHAR2(100)	PRICE NUMBER(10, 4)
1	Пепперони	502
1	Гавайская	510
(null)	Карбонара	490
1	Марио	450
(null)	Сицилия	510
3	Медовый	150
3	Тирамису	150
3	Наполеон	140

CATEGORY_ID NUMBER(10)	CATEGORY_NAME VARCHAR2(100)
1	Пицца
2	Напитки
3	Десерты

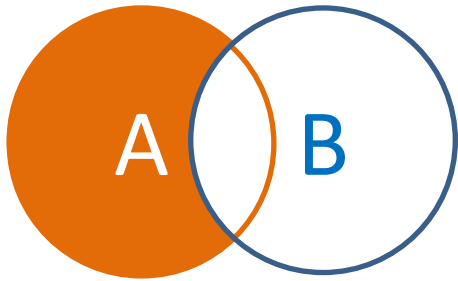
CATEGORY_NAME VARCHAR2(100)	PRODUCT_NAME VARCHAR2(100)	PRICE NUMBER(10, 4)
Пицца	Пепперони	502
Пицца	Гавайская	510
Пицца	Карбонара	490
Пицца	Марио	450
Пицца	Сицилия	510
Пицца	Медовый	150
Пицца	Тирамису	150
Пицца	Наполеон	140
Напитки	Пепперони	502
Напитки	Гавайская	510
Напитки	Карбонара	490
Напитки	Марио	450
Напитки	Сицилия	510
Напитки	Медовый	150
Напитки	Тирамису	150
Напитки	Наполеон	140
Десерты	Пепперони	502
Десерты	Гавайская	510
Десерты	Карбонара	490
Десерты	Марио	450
Десерты	Сицилия	510
Десерты	Медовый	150
Десерты	Тирамису	150
Десерты	Наполеон	140



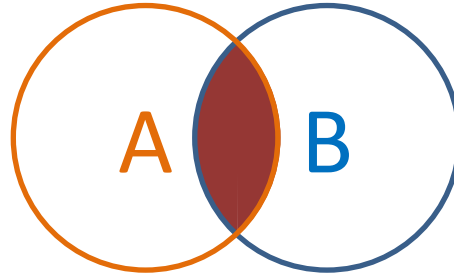
SELECT <...>  
 FROM A  
**LEFT JOIN** B ON A.KEY = B.KEY



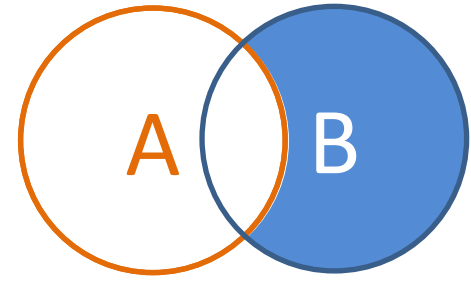
SELECT <...>  
 FROM A  
**RIGHT JOIN** B ON A.KEY = B.KEY



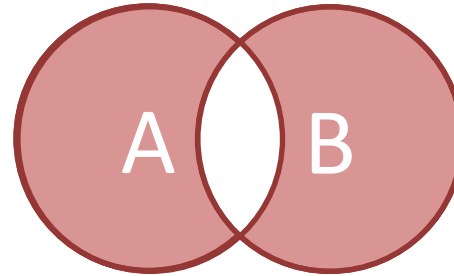
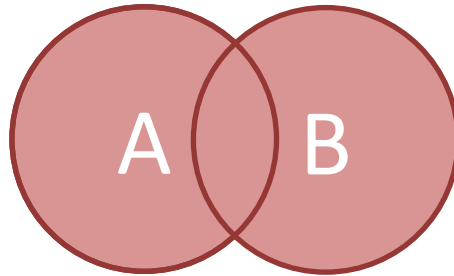
SELECT <...>  
 FROM A  
**LEFT JOIN** B ON A.KEY = B.KEY  
 WHERE B.KEY IS NULL



SELECT <...>  
 FROM A  
**INNER JOIN** B ON A.KEY = B.KEY



SELECT <...>  
 FROM A  
**RIGHT JOIN** B ON A.KEY = B.KEY  
 WHERE A.KEY IS NULL



SELECT <...>  
 FROM A  
**FULL OUTER RIGHT JOIN** B  
 ON A.KEY = B.KEY

SELECT <...>  
 FROM A  
**FULL OUTER RIGHT JOIN** B  
 ON A.KEY = B.KEY  
 WHERE A.KEY IS NULL  
 OR B.KEY IS NULL