



Объединение датафреймов

Константин Башевой
Аналитик-разработчик, Яндекс



Константин Башевой
Аналитик-разработчик
Яндекс

Помогаю аналитикам с инфраструктурой
Собираю инструменты обработки данных
Рассказываю, как это весело

Последние 10 лет:

Rambler&Co

Ростелеком

Яндекс

Что сегодня будет

Программа на сегодня

4



Чем merge
отличается от join,
и их типы в pandas



Дубликаты
как верный спутник
объединений



Оптимизация
хранения данных
с помощью join

Pandas и большие файлы

Большие файлы

6

Количество уникальных ID

user_id

1

1

1

1

2

2

3

9

Данные отсортированы

Большие файлы

7

Количество уникальных ID

user_id

1

1

1

1

2

2

3

9

- Читаем файл построчно
- Текущее VS прошлое значение
- Смена значения = пользователь

Большие файлы

8

Количество уникальных ID

user_id
1
1
1
1
2
2
3
9

- Читаем файл построчно
- Текущее VS прошлое значение
- Смена значения = пользователь
- В памяти 3 числа
- Размер файла не имеет значения

Сквозная аналитика

Сквозная аналитика

10

Склеить лог визитов и лог покупок

	user_id	source
0	11	ad
1	22	yandex
2	55	email
3	11	google
4	77	ad

	user_id	category
0	11	Спорт
1	22	Авто
2	55	Дача
3	11	Дети
4	99	Авто

Сквозная аналитика

11

Склеить лог визитов и лог покупок

	user_id	source
0	11	ad
1	22	yandex
2	55	email
3	11	google
4	77	ad

	user_id	source	category
0	22	yandex	Авто
1	55	email	Дача

+ еще user_id = 11 и 77

	user_id	category
0	11	Спорт
1	22	Авто
2	55	Дача
3	11	Дети
4	99	Авто

Первые проблемы

Проблемы

13

Нет однозначного соответствия

	user_id	source
0	11	ad
1	22	yandex
2	55	email
3	11	google
4	77	ad

	user_id	category
0	11	Спорт
1	22	Авто
2	55	Дача
3	11	Дети
4	99	Авто

Сумма визитов и покупок для user_id

```
visits_grouped = visits.groupby('user_id').count()  
visits_grouped.rename(columns={'source': 'visits'}, inplace=True)  
visits_grouped
```

visits	
user_id	
11	2
22	1
55	1
77	1

Сумма визитов и покупок для user_id

```
purchases_pivot = purchases.pivot_table(index='user_id', columns='category', values='user_id',  
                                         aggfunc='size', fill_value=0)  
purchases_pivot
```

category	Авто	Дача	Спорт
user_id			
11	0	0	2
22	1	0	0
55	0	1	0
99	1	0	0

Сумма визитов и покупок для user_id

```
purchases_pivot = purchases.pivot_table(index='user_id', columns='category', values='user_id',  
                                         aggfunc='size', fill_value=0)  
purchases_pivot
```

category	Авто	Дача	Спорт
user_id			
11	0	0	2
22	1	0	0
55	0	1	0
99	1	0	0

С count не работает

Типы объединений в Pandas

Типы объединений

18

Join – по индексу, merge – по столбцам

	user_id	source
0	11	ad
1	22	yandex
2	55	email
3	11	google
4	77	ad

	user_id	category
0	11	Спорт
1	22	Авто
2	55	Дача
3	11	Дети
4	99	Авто

Типы объединений

В таком варианте merge

```
visits.groupby('user_id').count().reset_index()
```

	user_id	source
0	11	2
1	22	1
2	55	1
3	77	1

user_id сейчас столбец

Типы объединений

В таком варианте merge

```
purchases_pivot.reset_index()
```

category	user_id	Авто	Дача	Спорт
0	11	0	0	2
1	22	1	0	0
2	55	0	1	0
3	99	1	0	0

user_id сейчас столбец

Типы объединений

21

Все параметры по умолчанию

```
visits_grouped.join(purchases_pivot)
```

	visits	Авто	Дача	Спорт
user_id				
11	2	0.0	0.0	2.0
22	1	1.0	0.0	0.0
55	1	0.0	1.0	0.0
77	1	NaN	NaN	NaN

LEFT и RIGHT JOIN

Left Join

23

Каждой строке левой таблицы ищет соответствие в правой

visits	
user_id	
11	2
22	1
55	1
77	1

category	Авто	Дача	Спорт
user_id			
11	0	0	2
22	1	0	0
55	0	1	0
99	1	0	0

Left Join

24

Каждой строке левой таблицы ищет соответствие в правой

visits	
user_id	
11	2
22	1
55	1
77	1

category	Авто	Дача	Спорт
user_id			
11	0	0	2
22	1	0	0
55	0	1	0
99	1	0	0

правая таблица

порядок важен!

левая таблица

Left Join

25

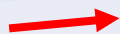
Каждой строке левой таблицы ищет соответствие в правой

visits			category Авто Дача Спорт			
user_id			user_id			
11	2	→	11	0	0	2
22	1	→	22	1	0	0
55	1	→	55	0	1	0
77	1		99	1	0	0

Left Join

Строчки без пары левой таблицы остаются,
Правой - удаляются без результата

visits	
user_id	
11	2
22	1
55	1
77	1



category	Авто	Дача	Спорт
user_id			
11	0	0	2
22	1	0	0
55	0	1	0
99	1	0	0

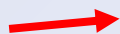
все строчки левой
таблицы останутся


NaN

Left Join

Строчки без пары левой таблицы остаются,
Правой - удаляются без результата

visits	
user_id	
11	2
22	1
55	1
77	1



category	Авто	Дача	Спорт
user_id			
11	0	0	2
22	1	0	0
55	0	1	0
 99	1	0	0

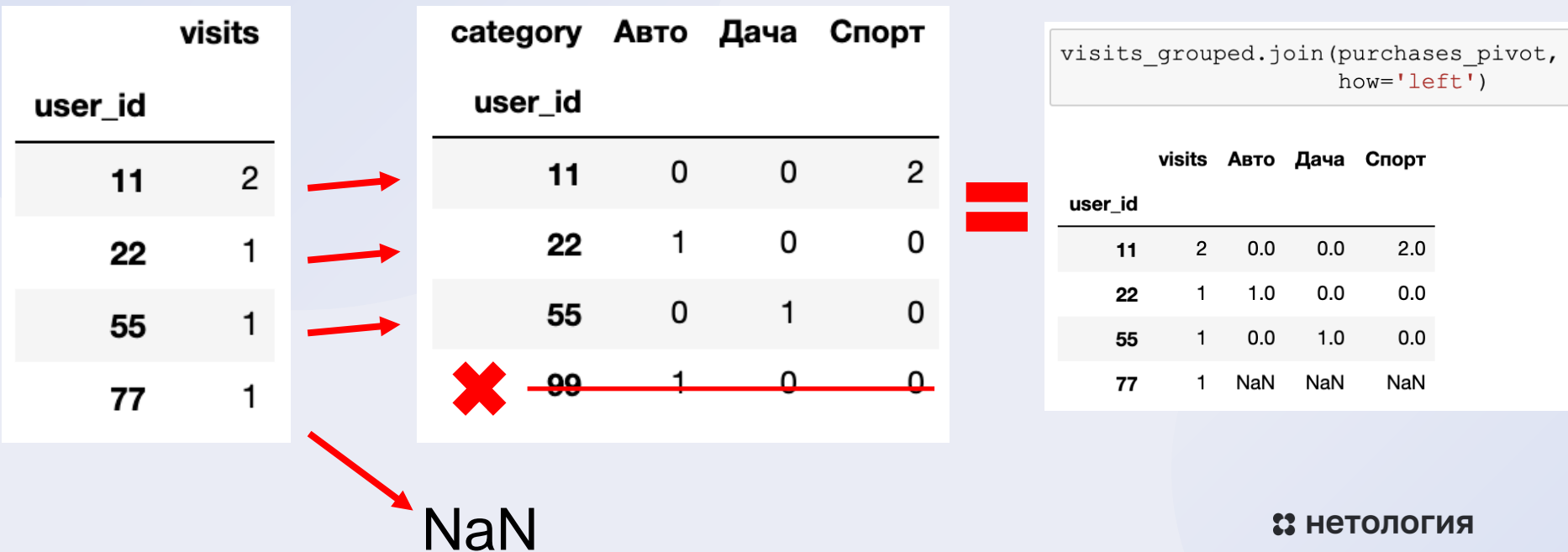
все строчки левой
таблицы останутся

правой - не факт

NaN

Left Join

Строчки без пары левой таблицы остаются,
Правой - удаляются без результата



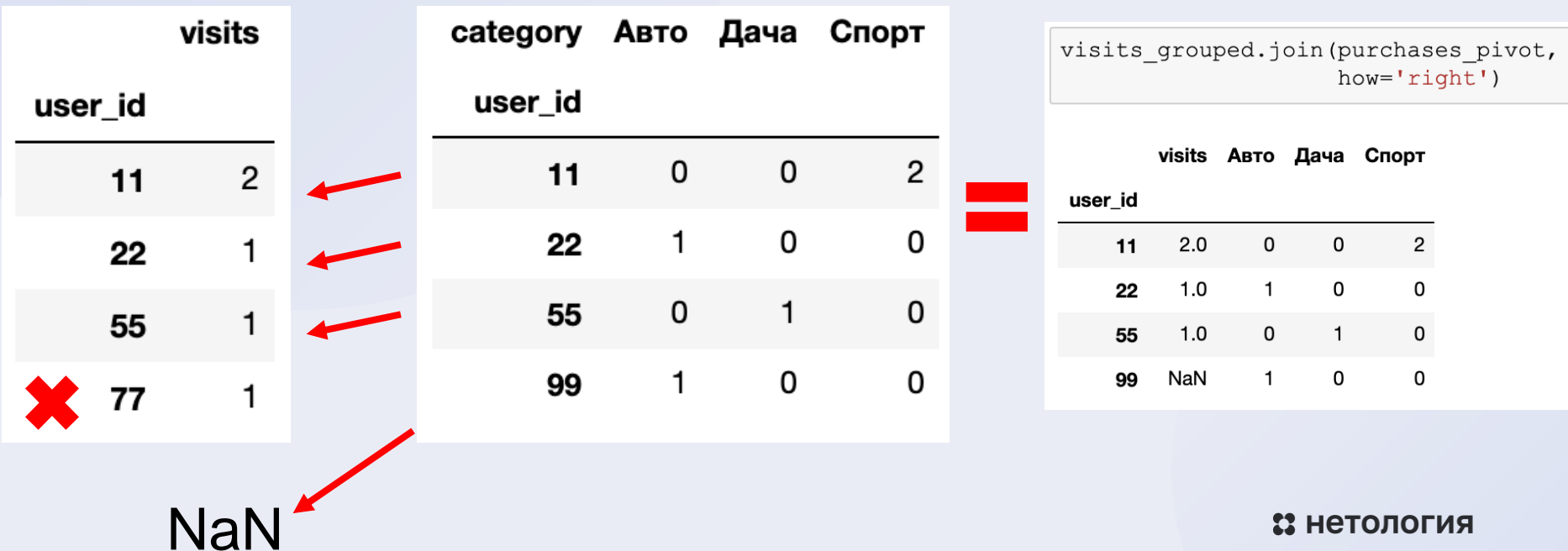
Right Join зеркален Left Join

Не рекомендуется к использованию без
особой на это необходимости

Right Join

30

Каждой строке правой таблицы ищет соответствие в левой



INNER и OUTER JOIN

Inner Join

32

Оставляет строки, в которые есть в обеих таблицах

visits	
user_id	
11	2
22	1
55	1
✗ 77	1



category	Авто	Дача	Спорт
user_id			
11	0	0	2
22	1	0	0
55	0	1	0
✗ 99	1	0	0



```
visits_grouped.join(purchases_pivot,  
                    how='inner')
```

	visits	Авто	Дача	Спорт
user_id				
11	2	0	0	2
22	1	1	0	0
55	1	0	1	0

Outer Join

Оставляет все строки

visits	
user_id	
11	2
22	1
55	1
77	1



category	Авто	Дача	Спорт
user_id			
11	0	0	2
22	1	0	0
55	0	1	0
99	1	0	0



```
visits_grouped.join(purchases_pivot,
                    how='outer')
```

	visits	Авто	Дача	Спорт
user_id				
11	2.0	0.0	0.0	2.0
22	1.0	1.0	0.0	0.0
55	1.0	0.0	1.0	0.0
77	1.0	NaN	NaN	NaN
99	NaN	1.0	0.0	0.0

NaN

NaN

Самое веселое в JOIN

Что полезно проверять (исходя из логики задачи)

- После LEFT-join количество строк не изменилось
- Суммы числовых столбцов не изменились
- Суммы в правой таблице тоже неплохо проверить

JOIN и оптимизация хранения

Как сэкономить место на диске

- Длинные повторяющиеся столбцы переводим в идентификаторы
- Составляем словари
- Логи - отдельно, словари - отдельно