

ノック2： データを結合(ユニオン)してみよう

ここでは、transaction_1とtransaction_2、transaction_detail_1とtransaction_detail_2のユニオンに挑戦してみます。

基本的には同じ作業となるので、ここではtransaction_1とtransaction_2の説明をします。transaction_1は既に読み込んでいるので、transaction_2を読み込んだ後、ユニオンしたデータをtransactionというデータフレーム型の変数に格納してみます。

また、先ほどと同様に先頭5行を出力しておきましょう。

```
transaction_2 = pd.read_csv('transaction_2.csv')
transaction = pd.concat([transaction_1, transaction_2], ignore_index=True)
transaction.head()
```

■図1-3：データユニオン

ノック2：データを結合(ユニオン)してみよう

```
In [5]: transaction_2 = pd.read_csv('transaction_2.csv')
transaction = pd.concat([transaction_1, transaction_2], ignore_index=True)
transaction.head
```

```
Out[5]:
```

	transaction_id	price	payment_date	customer_id
0	T0000000113	210000	2019-02-01 01:15:18	OA264235
1	T0000000114	50000	2019-02-01 01:27:09	IK058780
2	T0000000115	120000	2019-02-01 02:32:12	PL707949
3	T0000000116	210000	2019-02-01 02:43:32	OA300956
4	T0000000117	170000	2019-02-01 03:31:34	AS842206

2行目のpd.concatでユニオンを行なっています。

先頭5行の表示だと、図1-3のデータと違いが見られず、本当にユニオンができていくかわかりません。

ユニオンは、**データ数を行方向に増やす(縦に結合する)**ことですので、データ件数に変化があるはずで。そこで、データ件数の確認を行い、ユニオンができているか、検証しましょう。

```
print(len(transaction_1))
print(len(transaction_2))
print(len(transaction))
```

実行すると、5000、1737、6737と出力されるはずで。

5000と1737を足して、6737となりますので、ユニオンできていることが確認できました。

同様に、transaction_detailもユニオンしておきましょう。

```
transaction_detail_2 = pd.read_csv('transaction_detail_2.csv')
transaction_detail = pd.concat([transaction_detail_1, transaction_detail_2],
                                ignore_index=True)
transaction_detail.head()
```

これで、縦方向の結合は完了しました。

それでは次に、横方向のジョインに挑戦していきます。

まずは、売上データのジョインをやってみましょう。

ノック3： 売上データ同士を結合(ジョイン)してみよう

ジョインをする場合、しっかりと**主軸になるデータ**を考えつつ、どの列をキーにジョインするかを考えていきます。

主軸に置くデータは最も粒度が細かいtransaction_detailとします。

まずは、売上データをジョインしていきますが、①足りない(付加したい)データ列は何か？ ②共通するデータ列は何か？ を考えます。

今回の場合、付加したいデータはtransactionのpayment_date、customer_idです。transactionのpriceは、一回の購買データごとの合計金額となっており、transaction_detailのquantityとitem_masterのitem_priceから算出されるものです。**transaction_detailの方がデータの粒度が細かい**ため、付加してしまうと、二重計上になってしまうのでpriceを付加してはいけません。