

共通するデータ列は、transaction_idとなります。
それでは、ジョインしてみましょう。

```
join_data = pd.merge(transaction_detail, transaction[["transaction_id", "payment_date", "customer_id"]], on="transaction_id", how="left")
join_data.head()
print(len(transaction))
```

■図1-4：売上データのジョイン

ノック3：売上データ同士を結合(ジョイン)してみよう

```
In [8]: join_data = pd.merge(transaction_detail, transaction[["transaction_id", "payment_date", "customer_id"]], on="transaction_id", how="left")
join_data.head()

Out[8]:
```

	detail_id	transaction_id	item_id	quantity	payment_date	customer_id
0	0	T0000000113	S005	1	2019-02-01 01:15:18	OA264235
1	1	T0000000114	S001	1	2019-02-01 01:27:09	IK058780
2	2	T0000000115	S003	1	2019-02-01 02:32:12	PL707949
3	3	T0000000116	S005	1	2019-02-01 02:43:32	OA300956
4	4	T0000000117	S002	2	2019-02-01 03:31:34	AS842206

1行目でpd.mergeを用いてジョインを行っています。

主軸にするtransaction_detail、結合するtransactionデータの必要な列のみを引数に渡して、ジョインキーとしてtransaction_idを、ジョインの種類としてレフトジョインを指定しています。ジョインの種類に関しては、巻末のAppendix①で紹介していますのでご参照ください。

今回は、transaction_detailをメインに、結合しているので、レフトジョインとなります。

先頭5行の出力結果を見ると、transaction_detailのデータ列に、payment_date、customer_idが加えられているのがわかります。

では、データ件数を見てみましょう。

```
print(len(transaction_detail))
print(len(transaction))
print(len(join_data))
```

実行すると、7144、6737、7144と出力されます。

つまり、transaction_detailとjoin_dataは同じ件数となっており、縦にデータが増えたのではなく、横にデータが増えており、ジョインできていることが確

認できました。

ただし、ジョインするデータのジョインキー(今回ではtransactionのtransaction_id)に重複データが存在している(ユニークになっていない)場合、件数が増えることもあるので注意が必要です。

では、次に、顧客や商品情報を付加するために、customer_master、item_masterをジョインしてみましょう。

ノック4： マスターデータを結合(ジョイン)してみよう

今回もノック3と同様にジョインをします。

では、先ほどと同様に、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？ を考えていきます。

今回の場合、付加したいデータは、customer_master、item_masterに含まれるデータ全てになります。共通するデータ列は、それぞれcustomer_idとitem_idで紐付けが可能です。

それでは、ジョインしてみましょう。

```
join_data = pd.merge(join_data, customer_master, on="customer_id", how="left")
join_data = pd.merge(join_data, item_master, on="item_id", how="left")
join_data.head()
```

■図1-5：マスターデータのジョイン

ノック4：マスターデータを結合(ジョイン)してみよう

```
In [10]: join_data = pd.merge(join_data, customer_master, on="customer_id", how="left")
join_data = pd.merge(join_data, item_master, on="item_id", how="left")
join_data.head()

Out[10]:
```

	detail_id	transaction_id	item_id	quantity	payment_date	customer_id	customer_name	registration_date	customer_name_kana	em
0	0	T0000000113	S005	1	2019-02-01 01:15:18	OA264235	大下 遼	2019-01-09 11:46:29	おおした みつる	ooshita_mitsuru@example.cc
1	1	T0000000114	S001	1	2019-02-01 01:27:09	IK058780	新井 勲	2019-01-15 06:52:55	あらいつとむ	arai_toshimu@example.cc
2	2	T0000000115	S003	1	2019-02-01 02:32:12	PL707949	大浦 秀樹	2019-01-27 11:48:45	おおaura ひでき	ooura_hideki@example.cc
3	3	T0000000116	S005	1	2019-02-01 02:43:32	OA300956	国分 翔子	2019-01-23 01:44:11	こくぶん しょうこ	kokubun_shouko@example.cc
4	4	T0000000117	S002	2	2019-02-01 03:31:34	AS842206	小川 雅子	2019-01-18 04:05:43	おがわ ようこ	ogawa_yoko@example.cc