を表示しています。

それでは、他のデータも読み込んでみましょう。

transactionおよびtransaction_detailのデータは、1もしくは2のいずれかを読み込んでみましょう。

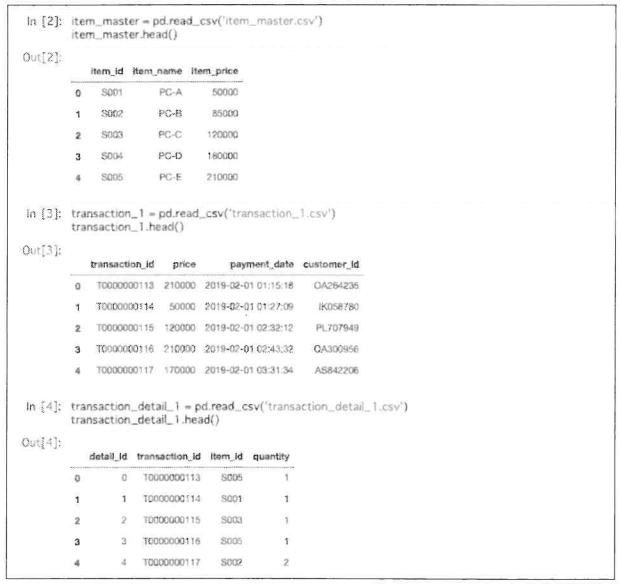
Jupyter-Notebookのセルはデータごとに分けて書きましょう。

```
item_master = pd.read_csv('item_master.csv')
item_master.head()
```

```
transaction_1 = pd.read_csv('transaction_1.csv')
transaction_1.head()
```

```
transaction_detail_1 = pd.read_csv('transaction_detail_1.csv')
transaction_detail_1.head()
```

図1-2：データの読み込み結果



```
In [2]: item_master = pd.read_csv('item_master.csv')
        item_master.head()
```

Out[2]:

	item_id	item_name	item_price
0	S001	PC-A	50000
1	S002	PC-B	85000
2	S003	PC-C	120000
3	S004	PC-D	180000
4	S005	PC-E	210000

```
In [3]: transaction_1 = pd.read_csv('transaction_1.csv')
        transaction_1.head()
```

Out[3]:

	transaction_id	price	payment_date	customer_id
0	T0000000113	210000	2019-02-01 01:15:18	QA264235
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	QA300956
4	T0000000117	170000	2019-02-01 03:31:34	AS842206

```
In [4]: transaction_detail_1 = pd.read_csv('transaction_detail_1.csv')
        transaction_detail_1.head()
```

Out[4]:

	detail_id	transaction_id	item_id	quantity
0	0	T0000000113	S005	1
1	1	T0000000114	S001	1
2	2	T0000000115	S003	1
3	3	T0000000116	S005	1
4	4	T0000000117	S002	2

実行すると、それぞれのデータの先頭5行のデータが確認できます。

全データの先頭5行を表示させることで、どのようなデータ列が存在するのか、それぞれのデータ列の関係性など、データの大枠を掴むことができます。

これまで、機械学習の入門書などでサンプルプログラムを触ってきた方は、機械学習や分析に適したデータが既に準備されているケースが多く、今回のように複数に渡ってデータが存在するケースを取り組んでいる方はいらっしゃらないのではないでしょうか。

しかしながら、実際の現場では、データをかき集めるところから始まり、データの概要を捉え、**分析に適した形に加工**することから始めることが多いです。

それでは、今回のケースに関して、データの大枠を掴んでいきましょう。

customer_masterには、顧客の性別や年齢などの顧客詳細情報が、item_masterには、商品名や商品単価の情報が格納されています。

そして、transactionデータには、いつ誰がいくら買ったのかという情報が、transaction_detailデータには購入した商品や数量などの情報が格納されています。

ではどのデータを使っていくのが良いのでしょうか。

分析業務の目的にも寄りますが、「売上をなんとかしたい」という抽象的なお題の場合でも、「今後の優良顧客を見つけたい」というような具体的なお題の場合でも、まずは**データの全体像を把握**することが重要です。

そのため、なるべくデータの粒度が細かいデータに合わせてデータを作成する必要があります。ここで取り扱っているようなECサイトの場合、当然ながら売上とは切っても切り離せないので、最も粒度が細かい売上関連のデータであるtransaction_detailを主軸に考えていきましょう。

transaction_detailをベースに考える場合、大きく2つのデータ加工を行う必要があります。

1つ目は、transaction_detail_1とtransaction_detail_2やtransaction_1とtransaction_2を縦に結合するユニオンです。

2つ目は、transaction_detailをもとに、transaction、customer_master、item_masterを横に結合するジョインです。

まずは、データユニオンから見ていきましょう。