

Prepare Data

```
In [9]: import pandas as pd
```

```
In [10]: df0 = pd.read_csv("../data/shipping_data_0.csv")
df0.head()
```

Out[10]:

	origin_warehouse	destination_store	product	on_time	product_quantity	driver_identifier
0	d5566b15-b071-4acf-8e8e-c98433083b2d	50d33715-4c77-4dd9-8b9d-ff1ca372a2a2	lotion	True	59	d8da0460-cf3e-4f38-9ff6c9b4e344d8e
1	c42f0de8-b4f0-4167-abd1-ae79e5e18eea	172eb8f3-1033-4fb6-b66b-d0df09df3161	windows	True	28	293ccaec-6592-4f04-aae53e238fe6261
2	b145f396-de9b-42f1-9cc9-f5b52c3a941c	65e4544d-42ae-4751-9580-bdcb90e5fcda	skis	True	63	80988f09-91a3-4e1b-8e6513551c53f31
3	f4372224-759f-43b3-bc83-ca6106bba1af	745bee4e-710c-4538-8df1-5c146e1092a6	bikes	True	47	5f79b402-6554d8e-8ff35ef05870e0a
4	49d0edae-9091-41bb-a08d-ab1c66bd08d5	425b7a1a-b744-4c6b-898e-d424dd8cf18e	candy	False	73	58beb5d3-98f84077-a9641f04f7cb11e

```
In [11]: df0.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110 entries, 0 to 109
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  -
0   origin_warehouse      110 non-null    object
1   destination_store      110 non-null    object
2   product                110 non-null    object
3   on_time                110 non-null    bool
4   product_quantity      110 non-null    int64
5   driver_identifier      110 non-null    object
dtypes: bool(1), int64(1), object(4)
memory usage: 4.5+ KB

In [12]: df1 = pd.read_csv("../data/shipping_data_1.csv")
df1.head()
```

Out[12]:

	shipment_identifier	product	on_time
0	449263b4-6c93-4f19-8b6a-0d99a29fc637	pants	False
1	449263b4-6c93-4f19-8b6a-0d99a29fc637	pants	False
2	449263b4-6c93-4f19-8b6a-0d99a29fc637	pants	False
3	449263b4-6c93-4f19-8b6a-0d99a29fc637	keyboards	False
4	449263b4-6c93-4f19-8b6a-0d99a29fc637	keyboards	False

In [13]: `df2 = pd.read_csv("./data/shipping_data_2.csv")`
`df2.head()`

Out[13]:

	shipment_identifier	origin_warehouse	destination_store	driver_identifier
0	449263b4-6c93-4f19-8b6a-0d99a29fc637	bb75bf7d-c008-4267-bf92-6089cff5fe56	5e9405de-a078-4b00-99c6-96564568b63c	c12025e6-6f9c-4728-8c3c-9f840bde6f1a
1	76e5b84a-9d09-4efb-8b43-a0c932b958bb	372fd2b1-b2a7-4553-b6d7-426a1bc88e56	e34973c8-9ca9-4a06-b497-7a8b49625fc2	85b8d394-a67c-48b6-b1de-55be323ba622
2	b541a47d-89b1-4805-97d0-1988832321f1	469d957f-28ef-4eac-956a-d2a42b06d3ab	fcadc756-61e9-41bb-871b-d3546c5aa981	47bdfc40-f3db-4678-b6a7-43f1e1c2fd32
3	3fc6b63d-27b4-408c-b3b3-e8e94a45b079	cd140190-a53b-4660-a5b4-cc844a6506f0	89ba200c-ca90-443a-b64f-397bce091eae	5ae3e541-2098-45b6-8d94-35d176185606
4	491ee4e8-be80-4f52-802b-d8fe1a6bd487	c6addf8b-eea6-43b8-9040-b5620b1a0d99	7aebe820-8478-4a29-a606-7c59af677e24	1f228b52-7165-4d7f-a731-3f7707aefb2a

In [14]: `df_merged = pd.merge(df1, df2, how="inner", on="shipment_identifier")`
`df_merged.head()`

Out[14]:

	shipment_identifier	product	on_time	origin_warehouse	destination_store	driver_iden
0	449263b4-6c93-4f19-8b6a-0d99a29fc637	pants	False	bb75bf7d-c008-4267-bf92-6089cff5fe56	5e9405de-a078-4b00-99c6-96564568b63c	c12025e6-4728-9f840bd
1	449263b4-6c93-4f19-8b6a-0d99a29fc637	pants	False	bb75bf7d-c008-4267-bf92-6089cff5fe56	5e9405de-a078-4b00-99c6-96564568b63c	c12025e6-4728-9f840bd
2	449263b4-6c93-4f19-8b6a-0d99a29fc637	pants	False	bb75bf7d-c008-4267-bf92-6089cff5fe56	5e9405de-a078-4b00-99c6-96564568b63c	c12025e6-4728-9f840bd
3	449263b4-6c93-4f19-8b6a-0d99a29fc637	keyboards	False	bb75bf7d-c008-4267-bf92-6089cff5fe56	5e9405de-a078-4b00-99c6-96564568b63c	c12025e6-4728-9f840bd
4	449263b4-6c93-4f19-8b6a-0d99a29fc637	keyboards	False	bb75bf7d-c008-4267-bf92-6089cff5fe56	5e9405de-a078-4b00-99c6-96564568b63c	c12025e6-4728-9f840bd



```
In [15]: df_final = df_merged[['origin_warehouse', 'destination_store', 'product',
df_final.head()
```

Out[15]:

	origin_warehouse	destination_store	product	on_time	driver_identifier
0	bb75bf7d-c008-4267-bf92-6089cff5fe56	5e9405de-a078-4b00-99c6-96564568b63c	pants	False	c12025e6-6f9c-4728-8c3c-9f840bde6f1a
1	bb75bf7d-c008-4267-bf92-6089cff5fe56	5e9405de-a078-4b00-99c6-96564568b63c	pants	False	c12025e6-6f9c-4728-8c3c-9f840bde6f1a
2	bb75bf7d-c008-4267-bf92-6089cff5fe56	5e9405de-a078-4b00-99c6-96564568b63c	pants	False	c12025e6-6f9c-4728-8c3c-9f840bde6f1a
3	bb75bf7d-c008-4267-bf92-6089cff5fe56	5e9405de-a078-4b00-99c6-96564568b63c	keyboards	False	c12025e6-6f9c-4728-8c3c-9f840bde6f1a
4	bb75bf7d-c008-4267-bf92-6089cff5fe56	5e9405de-a078-4b00-99c6-96564568b63c	keyboards	False	c12025e6-6f9c-4728-8c3c-9f840bde6f1a

```
In [16]: df_final.duplicated().sum()
```

```
Out[16]: np.int64(66)
```

```
In [17]: df_final_Product_qunantity = df_final.groupby(['origin_warehouse', 'destinatio
df_final_Product_qunantity.head()
```

Out[17]:

	origin_warehouse	destination_store	product	on_time	driver_identifier	product_quantit
0	026ffd7c-cff2-4daa-ae76-6768d3283861	d33a7b5d-15e5-4ba6-93ef-136bbcbf4946	pants	True	6e905a32-46a8-4555-8434-a6b16580e873	
1	026ffd7c-cff2-4daa-ae76-6768d3283861	d33a7b5d-15e5-4ba6-93ef-136bbcbf4946	pencils	True	6e905a32-46a8-4555-8434-a6b16580e873	
2	20efa3c2-c498-4908-8af4-f81c76781912	0167b0c3-60fd-4fbd-b378-fba0ceac39ff	eggs	False	45a23a59-09cb-4906-8293-6014b72de249	
3	20efa3c2-c498-4908-8af4-f81c76781912	0167b0c3-60fd-4fbd-b378-fba0ceac39ff	toys	False	45a23a59-09cb-4906-8293-6014b72de249	
4	372fd2b1-b2a7-4553-b6d7-426a1bc88e56	e34973c8-9ca9-4a06-b497-7a8b49625fc2	baby food	False	85b8d394-a67c-48b6-b1de-55be323ba622	

In [18]:

```
df_f = df_final_Product_quantity[['origin_warehouse', 'destination_store', 'pr
df_f.head()
```

Out[18]:

	origin_warehouse	destination_store	product	on_time	product_quantity	driver_identifie
0	026ffd7c-cff2-4daa-ae76-6768d3283861	d33a7b5d-15e5-4ba6-93ef-136bbcbf4946	pants	True	1	6e905a32-46a8-4555-8434-a6b16580e87
1	026ffd7c-cff2-4daa-ae76-6768d3283861	d33a7b5d-15e5-4ba6-93ef-136bbcbf4946	pencils	True	4	6e905a32-46a8-4555-8434-a6b16580e87
2	20efa3c2-c498-4908-8af4-f81c76781912	0167b0c3-60fd-4fbd-b378-fba0ceac39ff	eggs	False	3	45a23a59-09cb-4906-8293-6014b72de24
3	20efa3c2-c498-4908-8af4-f81c76781912	0167b0c3-60fd-4fbd-b378-fba0ceac39ff	toys	False	3	45a23a59-09cb-4906-8293-6014b72de24
4	372fd2b1-b2a7-4553-b6d7-426a1bc88e56	e34973c8-9ca9-4a06-b497-7a8b49625fc2	baby food	False	1	85b8d394-a67c-48b6-b1de-55be323ba62

In [19]:

```
df_f.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44 entries, 0 to 43
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   origin_warehouse      44 non-null    object
1   destination_store     44 non-null    object
2   product               44 non-null    object
3   on_time               44 non-null    bool
4   product_quantity      44 non-null    int64
5   driver_identifier     44 non-null    object
dtypes: bool(1), int64(1), object(4)
memory usage: 1.9+ KB
```

```
In [43]: df_f.to_csv('./data/shipping_data_1&2_formated.csv', index=False)
```

```
In [20]: df = pd.concat([df0, df_f], ignore_index=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 154 entries, 0 to 153
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   origin_warehouse      154 non-null    object
1   destination_store     154 non-null    object
2   product               154 non-null    object
3   on_time               154 non-null    bool
4   product_quantity      154 non-null    int64
5   driver_identifier     154 non-null    object
dtypes: bool(1), int64(1), object(4)
memory usage: 6.3+ KB
```

Data to DB

```
In [21]: import sqlite3
```

```
In [43]: con = sqlite3.connect("shipment_database.db")
cur = con.cursor()
r = cur.execute("SELECT name FROM sqlite_master WHERE type='table'").fetchall()
r
```

```
Out[43]: [('product',), ('shipment',)]
```

```
In [23]: cur.execute("SELECT * FROM shipment")
column_names = [description[0] for description in cur.description]

print(column_names)
```

```
['id', 'product_id', 'quantity', 'origin', 'destination']
```

```
In [24]: cur.execute("SELECT * FROM product")
column_names = [description[0] for description in cur.description]
```

```
print(column_names)
```

```
['id', 'name']
```

```
In [25]: df.head()
```

```
Out[25]:
```

	origin_warehouse	destination_store	product	on_time	product_quantity	driver_id
0	d5566b15-b071-4acf-8e8e-c98433083b2d	50d33715-4c77-4dd9-8b9d-ff1ca372a2a2	lotion	True	59	d8da0460-cf39-4f38-9ff6c9b4e344d8
1	c42f0de8-b4f0-4167-abd1-ae79e5e18eea	172eb8f3-1033-4fb6-b66b-d0df09df3161	windows	True	28	293ccaec-6592-4f04-aae53e238fe6261
2	b145f396-de9b-42f1-9cc9-f5b52c3a941c	65e4544d-42ae-4751-9580-bdcb90e5fcda	skis	True	63	80988f09-91a3-4e1b-8e6913551c53f31
3	f4372224-759f-43b3-bc83-ca6106bba1af	745bee4e-710c-4538-8df1-5c146e1092a6	bikes	True	47	5f79b402-6554d8e-8ff5ef05870e0a
4	49d0edae-9091-41bb-a08d-ab1c66bd08d5	425b7a1a-b744-4c6b-898e-d424dd8cf18e	candy	False	73	58beb5d3-98ff-4077-a9641f04f7cb11e

```
In [26]: df['product'].nunique()
```

```
Out[26]: 45
```

```
In [27]: df_product = pd.DataFrame({'product_name':df['product'].unique()})
df_product['id'] = df_product.index + 1
df_product.head()
```

```
Out[27]:
```

	product_name	id
0	lotion	1
1	windows	2
2	skis	3
3	bikes	4
4	candy	5

```
In [28]: df = pd.merge(df,df_product,how='inner',left_on='product',right_on='product_name')
df.head()
```

Out[28]:

	origin_warehouse	destination_store	product	on_time	product_quantity	driver_identifi
0	d5566b15-b071-4acf-8e8e-c98433083b2d	50d33715-4c77-4dd9-8b9d-ff1ca372a2a2	lotion	True	59	d8da0460-cf39-4f38-9ff6c9b4e344d8
1	c42f0de8-b4f0-4167-abd1-ae79e5e18eea	172eb8f3-1033-4fb6-b66b-d0df09df3161	windows	True	28	293ccaec-6592-4f04-aae53e238fe6261
2	b145f396-de9b-42f1-9cc9-f5b52c3a941c	65e4544d-42ae-4751-9580-bdcb90e5fcda	skis	True	63	80988f09-91a3-4e1b-8e6913551c53f31
3	f4372224-759f-43b3-bc83-ca6106bba1af	745bee4e-710c-4538-8df1-5c146e1092a6	bikes	True	47	5f79b402-6554d8e-8ff35ef05870e0a
4	49d0edae-9091-41bb-a08d-ab1c66bd08d5	425b7a1a-b744-4c6b-898e-d424dd8cf18e	candy	False	73	58beb5d3-98ff-4077-a9641f04f7cb11e

In [29]:

```
df = df.rename(columns={'id':'product_id'})
df.head()
```

Out[29]:

	origin_warehouse	destination_store	product	on_time	product_quantity	driver_identifi
0	d5566b15-b071-4acf-8e8e-c98433083b2d	50d33715-4c77-4dd9-8b9d-ff1ca372a2a2	lotion	True	59	d8da0460-cf39-4f38-9ff6c9b4e344d8
1	c42f0de8-b4f0-4167-abd1-ae79e5e18eea	172eb8f3-1033-4fb6-b66b-d0df09df3161	windows	True	28	293ccaec-6592-4f04-aae53e238fe6261
2	b145f396-de9b-42f1-9cc9-f5b52c3a941c	65e4544d-42ae-4751-9580-bdcb90e5fcda	skis	True	63	80988f09-91a3-4e1b-8e6913551c53f31
3	f4372224-759f-43b3-bc83-ca6106bba1af	745bee4e-710c-4538-8df1-5c146e1092a6	bikes	True	47	5f79b402-6554d8e-8ff35ef05870e0a
4	49d0edae-9091-41bb-a08d-ab1c66bd08d5	425b7a1a-b744-4c6b-898e-d424dd8cf18e	candy	False	73	58beb5d3-98ff-4077-a9641f04f7cb11e

In [30]:

```
df = df[['origin_warehouse','destination_store','product_id','product', 'on_time',
df['id'] = df.index + 1
```

In [31]:

```
df = df[['id','origin_warehouse','destination_store','product_id','product', 'on
df.head()
```

Out[31]:

	id	origin_warehouse	destination_store	product_id	product	on_time	product_quantity
0	1	d5566b15-b071-4acf-8e8e-c98433083b2d	50d33715-4c77-4dd9-8b9d-ff1ca372a2a2	1	lotion	True	59
1	2	c42f0de8-b4f0-4167-abd1-ae79e5e18eea	172eb8f3-1033-4fb6-b66b-d0df09df3161	2	windows	True	28
2	3	b145f396-de9b-42f1-9cc9-f5b52c3a941c	65e4544d-42ae-4751-9580-bdcb90e5fcda	3	skis	True	63
3	4	f4372224-759f-43b3-bc83-ca6106bba1af	745bee4e-710c-4538-8df1-5c146e1092a6	4	bikes	True	47
4	5	49d0edae-9091-41bb-a08d-ab1c66bd08d5	425b7a1a-b744-4c6b-898e-d424dd8cf18e	5	candy	False	73



```
In [32]: cur.execute("SELECT * FROM product")
column_names = [description[0] for description in cur.description]

print(column_names)

['id', 'name']
```

```
In [33]: df_product.head()
```

Out[33]:

	product_name	id
0	lotion	1
1	windows	2
2	skis	3
3	bikes	4
4	candy	5

```
In [54]: for _, i in df_product.iterrows():
cur.execute(f"insert into product (id,name) values ({i['id']},{i['product_name']})")
```

```
In [44]: r = cur.execute("select * from product").fetchall()
for i in r[:2]:
    print(i)
```

```
(1, 'lotion')
(2, 'windows')
```

```
In [35]: con.commit()
```



```
In [45]: len(r)
```

```
Out[45]: 45
```

```
In [36]: cur.execute("SELECT * FROM shipment")
column_names = [description[0] for description in cur.description]

print(column_names)

['id', 'product_id', 'quantity', 'origin', 'destination']
```

```
In [37]: df.head()
```

```
Out[37]:
```

	id	origin_warehouse	destination_store	product_id	product	on_time	product_quantity
0	1	d5566b15-b071-4acf-8e8e-c98433083b2d	50d33715-4c77-4dd9-8b9d-ff1ca372a2a2	1	lotion	True	59
1	2	c42f0de8-b4f0-4167-abd1-ae79e5e18eea	172eb8f3-1033-4fb6-b66b-d0df09df3161	2	windows	True	28
2	3	b145f396-de9b-42f1-9cc9-f5b52c3a941c	65e4544d-42ae-4751-9580-bdcb90e5fcda	3	skis	True	63
3	4	f4372224-759f-43b3-bc83-ca6106bba1af	745bee4e-710c-4538-8df1-5c146e1092a6	4	bikes	True	47
4	5	49d0edae-9091-41bb-a08d-ab1c66bd08d5	425b7a1a-b744-4c6b-898e-d424dd8cf18e	5	candy	False	73

```
In [38]: for _, i in df.iterrows():
cur.execute(f"insert into shipment values ({i['id']},{i['product_id']},{i['prod
```

```
In [39]: r = cur.execute("select * from shipment").fetchall()
for i in r[:2]:
print(i)
```

```
(1, 1, 59, 'd5566b15-b071-4acf-8e8e-c98433083b2d', '50d33715-4c77-4dd9-8b9d-ff1ca372a2a2')
(2, 2, 28, 'c42f0de8-b4f0-4167-abd1-ae79e5e18eea', '172eb8f3-1033-4fb6-b66b-d0df09df3161')
```

```
In [40]: con.commit()
```

```
In [41]: cur.close()
con.close()
```

```
In [42]: len(r)
```

Out[42]: 154