In [ ]:

In [1]:
```python
import sqlite3
import pandas as pd
```

In [2]:
```python
print("heello")
```
heello

In [3]:
```python
connections=sqlite3.connect('chaitu.sqlite')
```

In [4]:
```python
cursor=connections.cursor()
cursor.execute("""select name from sqlite_master where type='table';""")
```
Out[4]: <sqlite3.Cursor at 0x19c9dde7ac0>

In [5]:
```python
print('list of tables present in the database')
```
list of tables present in the database

In [6]:
```python
table_list=[table[0] for table in cursor.fetchall()]
table_list
```
Out[6]:
```
['aircrafts_data',
 'airports_data',
 'boarding_passes',
 'bookings',
 'flights',
 'seats',
 'ticket_flights',
 'tickets']
```

In [7]:
```python
aircrafts_data=pd.read_sql_query("select * from aircrafts_data",connections)
aircrafts_data
```
Out[7]:

	aircraft_code	model	range
0	773	{"en": "Boeing 777-300", "ru": "Боинг 777-300"}	11100
1	763	{"en": "Boeing 767-300", "ru": "Боинг 767-300"}	7900
2	SU9	{"en": "Sukhoi Superjet-100", "ru": "Сухой Суп...	3000
3	320	{"en": "Airbus A320-200", "ru": "Аэробус A320-...	5700
4	321	{"en": "Airbus A321-200", "ru": "Аэробус A321-...	5600
5	319	{"en": "Airbus A319-100", "ru": "Аэробус A319-...	6700
6	733	{"en": "Boeing 737-300", "ru": "Боинг 737-300"}	4200
7	CN1	{"en": "Cessna 208 Caravan", "ru": "Сессна 208...	1200
8	CR2	{"en": "Bombardier CRJ-200", "ru": "Бомбардье ...	2700

In [8]:
```python
aircrafts_data.head()
```
Out[8]:

	aircraft_code	model	range
0	773	{"en": "Boeing 777-300", "ru": "Боинг 777-300"}	11100
1	763	{"en": "Boeing 767-300", "ru": "Боинг 767-300"}	7900
2	SU9	{"en": "Sukhoi Superjet-100", "ru": "Сухой Суп...	3000
3	320	{"en": "Airbus A320-200", "ru": "Аэробус A320-...	5700
4	321	{"en": "Airbus A321-200", "ru": "Аэробус A321-...	5600

In [9]:
```python
aircrafts_data
```
Out[9]:

	aircraft_code	model	range
0	773	{"en": "Boeing 777-300", "ru": "Боинг 777-300"}	11100
1	763	{"en": "Boeing 767-300", "ru": "Боинг 767-300"}	7900
2	SU9	{"en": "Sukhoi Superjet-100", "ru": "Сухой Суп...	3000
3	320	{"en": "Airbus A320-200", "ru": "Аэробус A320-...	5700
4	321	{"en": "Airbus A321-200", "ru": "Аэробус A321-...	5600
5	319	{"en": "Airbus A319-100", "ru": "Аэробус A319-...	6700
6	733	{"en": "Boeing 737-300", "ru": "Боинг 737-300"}	4200
7	CN1	{"en": "Cessna 208 Caravan", "ru": "Сессна 208...	1200
8	CR2	{"en": "Bombardier CRJ-200", "ru": "Бомбардье ...	2700

In [10]:
```python
airports_data=pd.read_sql_query("select * from airports_data",connections)
airports_data
```
Out[10]:

	airport_code	airport_name	city	coordinates	timezone
0	YKS	{"en": "Yakutsk Airport", "ru": "Якутск"}	{"en": "Yakutsk", "ru": "Якутск"}	(129.77099609375,62.0932998657226562)	Asia/Yakutsk
1	MJZ	{"en": "Mirny Airport", "ru": "Мирный"}	{"en": "Mirnyj", "ru": "Мирный"}	(114.03900146484375,62.534698486328125)	Asia/Yakutsk
2	KHV	{"en": "Khabarovsk-Novy Airport", "ru": "Хабар...	{"en": "Khabarovsk", "ru": "Хабаровск"}	(135.18800354004,48.5279998779300001)	Asia/Vladivostok
3	PKC	{"en": "Yelizovo Airport", "ru": "Елизово"}	{"en": "Petropavlovsk", "ru": "Петропавловск-К...	(158.453994750976562,53.1679000854492188)	Asia/Kamchatka
4	UUS	{"en": "Yuzhno-Sakhalinsk Airport", "ru": "Юж...	{"en": "Yuzhno-Sakhalinsk", "ru": "Южно-Сахали...	(142.718002319335938,46.8886985778808594)	Asia/Sakhalin
...	...	...	...	...	...
99	MMK	{"en": "Murmansk Airport", "ru": "Мурманск"}	{"en": "Murmansk", "ru": "Мурманск"}	(32.7508010864257812,68.7817001342773438)	Europe/Moscow
100	ABA	{"en": "Abakan Airport", "ru": "Абакан"}	{"en": "Abakan", "ru": "Абакан"}	(91.3850021362304688,53.7400016784667969)	Asia/Krasnoyarsk
101	BAX	{"en": "Barnaul Airport", "ru": "Барнаул"}	{"en": "Barnaul", "ru": "Барнаул"}	(83.5384979248046875,53.363800048828125)	Asia/Krasnoyarsk
102	AAQ	{"en": "Anapa Vityazevo Airport", "ru": "Витяз...	{"en": "Anapa", "ru": "Анапа"}	(37.3473014831539984,45.002101898192997)	Europe/Moscow
103	CNN	{"en": "Chulman Airport", "ru": "Чульман"}	{"en": "Neryungri", "ru": "Нерюнгри"}	(124.914001464839998,56.9138984680179973)	Asia/Yakutsk

104 rows × 5 columns

In [11]:
```python
boarding=pd.read_sql_query("select * from boarding_passes",connections)
```

In [12]:
```python
boarding
```
Out[12]:

	ticket_no	flight_id	boarding_no	seat_no
0	0005435212351	30625	1	2D
1	0005435212386	30625	2	3G
2	0005435212381	30625	3	4H
3	0005432211370	30625	4	5D
4	0005435212357	30625	5	11A
...	...	...	...	...
579681	0005434302871	19945	85	20F
579682	0005432892791	19945	86	21C
579683	0005434302869	19945	87	20E
579684	0005432802476	19945	88	21F
579685	0005432802482	19945	89	21E

579686 rows × 4 columns

In [13]:
```python
boarding.head()
```
Out[13]:

	ticket_no	flight_id	boarding_no	seat_no
0	0005435212351	30625	1	2D
1	0005435212386	30625	2	3G
2	0005435212381	30625	3	4H
3	0005432211370	30625	4	5D
4	0005435212357	30625	5	11A

In [14]:
```python
total_bookings=pd.read_sql_query("select * from bookings",connections)
```

In [15]:
```python
total_bookings
```
Out[15]:

	book_ref	book_date	total_amount
0	00000F	2017-07-05 03:12:00+03	265700
1	000012	2017-07-14 09:02:00+03	37900
2	000068	2017-08-15 14:27:00+03	18100
3	000181	2017-08-10 13:28:00+03	131800
4	0002D8	2017-08-07 21:40:00+03	23600
...	...	...	...
262783	FFFEF3	2017-07-17 07:23:00+03	56000
262784	FFFF2C	2017-08-08 05:55:00+03	10800
262785	FFFF43	2017-07-20 20:42:00+03	78500
262786	FFFFA8	2017-08-08 04:45:00+03	28800
262787	FFFFF7	2017-07-01 22:12:00+03	73600

262788 rows × 3 columns

In [53]:
```python
total_flights=pd.read_sql_query("select * from flights",connections)
```

In [54]:
```python
total_flights
```
Out[54]:

	flight_id	flight_no	scheduled_departure	scheduled_arrival	departure_airport	arrival_airport	status	aircraft_code	actual_departure	actual_arrival
0	1185	PG0134	2017-09-10 09:50:00+03	2017-09-10 14:55:00+03	DME	BTK	Scheduled	319	\N	\N
1	3979	PG0052	2017-08-25 14:50:00+03	2017-08-25 17:35:00+03	VKO	HMA	Scheduled	CR2	\N	\N
2	4739	PG0561	2017-09-05 12:30:00+03	2017-09-05 14:15:00+03	VKO	AER	Scheduled	763	\N	\N
3	5502	PG0529	2017-09-12 09:50:00+03	2017-09-12 11:20:00+03	SVO	UFA	Scheduled	763	\N	\N
4	6938	PG0461	2017-09-04 12:25:00+03	2017-09-04 13:20:00+03	SVO	ULV	Scheduled	SU9	\N	\N
...	...	...	...	...	...	...	...	...	...	...
33116	33117	PG0063	2017-08-02 19:25:00+03	2017-08-02 20:10:00+03	SKX	SVO	Arrived	CR2	2017-08-02 19:25:00+03	2017-08-02 20:10:00+03
33117	33118	PG0063	2017-07-28 19:25:00+03	2017-07-28 20:10:00+03	SKX	SVO	Arrived	CR2	2017-07-28 19:30:00+03	2017-07-28 20:15:00+03
33118	33119	PG0063	2017-09-08 19:25:00+03	2017-09-08 20:10:00+03	SKX	SVO	Scheduled	CR2	\N	\N
33119	33120	PG0063	2017-08-01 19:25:00+03	2017-08-01 20:10:00+03	SKX	SVO	Arrived	CR2	2017-08-01 19:26:00+03	2017-08-01 20:12:00+03
33120	33121	PG0063	2017-08-26 19:25:00+03	2017-08-26 20:10:00+03	SKX	SVO	Scheduled	CR2	\N	\N

33121 rows × 10 columns

In [18]:
```python
seats=pd.read_sql_query("select * from seats",connections)
```

In [19]:
```python
seats
```
Out[19]:

	aircraft_code	seat_no	fare_conditions
0	319	2A	Business
1	319	2C	Business
2	319	2D	Business
3	319	2F	Business
4	319	3A	Business
...	...	...	...
1334	773	48H	Economy
1335	773	48K	Economy
1336	773	49A	Economy
1337	773	49C	Economy
1338	773	49D	Economy

1339 rows × 3 columns

In [20]:
```python
tickets_rate=pd.read_sql_query("select * from tickets",connections)
```

In [21]:
```python
tickets_rate
```
Out[21]:

	ticket_no	book_ref	passenger_id
0	0005432000987	06B046	8149 604011
1	0005432000988	06B046	8499 420203
2	0005432000989	E170C3	1011 752484
3	0005432000990	E170C3	4849 400049
4	0005432000991	F313DD	6615 976589
...	...	...	...
366728	0005435999869	D730BA	0474 690760
366729	0005435999870	D730BA	6535 751108
366730	0005435999871	A1AD46	1596 156448
366731	0005435999872	7B6A53	9374 822707
366732	0005435999873	7B6A53	7380 075822

366733 rows × 3 columns

In [33]:
```python
for table in table_list:
    print('\ntable:',table)
    column_info=connections.execute("PRAGMA table_info({})".format(table))
    for column in column_info.fetchall():
        print(column[1:3])
```
```
table: aircrafts_data
('aircraft_code', 'character(3)')
('model', 'jsonb')
('range', 'INTEGER')

table: airports_data
('airport_code', 'character(3)')
('airport_name', 'jsonb')
('city', 'jsonb')
('coordinates', 'point')
('timezone', 'TEXT')

table: boarding_passes
('ticket_no', 'character(13)')
('flight_id', 'INTEGER')
('boarding_no', 'INTEGER')
('seat_no', 'character varying(4)')

table: bookings
('book_ref', 'character(6)')
('book_date', 'timestamp with time zone')
('total_amount', 'numeric(10,2)')

table: flights
('flight_id', 'INTEGER')
('flight_no', 'character(6)')
('scheduled_departure', 'timestamp with time zone')
('scheduled_arrival', 'timestamp with time zone')
('departure_airport', 'character(3)')
('arrival_airport', 'character(3)')
('status', 'character varying(20)')
('aircraft_code', 'character(3)')
('actual_departure', 'timestamp with time zone')
('actual_arrival', 'timestamp with time zone')

table: seats
('aircraft_code', 'character(3)')
('seat_no', 'character varying(4)')
('fare_conditions', 'character varying(10)')

table: ticket_flights
('ticket_no', 'character(13)')
('flight_id', 'INTEGER')
('fare_conditions', 'character varying(10)')
('amount', 'numeric(10,2)')

table: tickets
('ticket_no', 'character(13)')
('book_ref', 'character(6)')
('passenger_id', 'character varying(20)')
```

In [32]:
```python
for table in table_list:
    print('\ntable:',table)
    df_table=pd.read_sql_query(f"select * from {table}",connections)
    print(df_table.isnull().sum())
```
```
table: aircrafts_data
aircraft_code    0
model            0
range            0
dtype: int64

table: airports_data
airport_code    0
airport_name    0
city            0
coordinates     0
timezone        0
dtype: int64

table: boarding_passes
ticket_no      0
flight_id      0
boarding_no    0
seat_no        0
dtype: int64

table: bookings
book_ref        0
book_date       0
total_amount    0
dtype: int64

table: flights
flight_id              0
flight_no              0
scheduled_departure    0
scheduled_arrival      0
departure_airport      0
arrival_airport        0
status                 0
aircraft_code          0
actual_departure       0
actual_arrival         0
dtype: int64

table: seats
aircraft_code      0
seat_no            0
fare_conditions    0
dtype: int64

table: ticket_flights
ticket_no          0
flight_id          0
fare_conditions    0
amount             0
dtype: int64

table: tickets
ticket_no       0
book_ref        0
passenger_id    0
dtype: int64
```

In [24]:
```python
# HOW MANY PLANES HAVE MORE THAN 100 SEATS
```

In [25]:
```python
pd.read_sql_query("""select aircraft_code,count(*) as total_seats from seats group by aircraft_code having total_seats> 100""",connections)
```
Out[25]:

	aircraft_code	total_seats
0	319	116
1	320	140
2	321	170
3	733	130
4	763	222
5	773	402

In [26]:
```python
# HOW MANY NUMBER OF TICKETS BOOKED AND TOTAL AMOUNT EARNED CHANGED WITH THE TIME
```

In [34]:
```python
tickets=pd.read_sql_query("""select * from tickets inner join bookings on tickets.book_ref=bookings.book_ref""",connections)
```

In [35]:
```python
tickets.dtypes
```
Out[35]:
```
ticket_no       object
book_ref        object
passenger_id    object
book_date       object
total_amount     int64
dtype: object
```

In [36]:
```python
tickets['book_date']=pd.to_datetime(tickets['book_date'])
```

In [37]:
```python
tickets.dtypes
```
Out[37]:
```
ticket_no                            object
book_ref                             object
passenger_id                         object
book_date       datetime64[ns, UTC+03:00]
total_amount                          int64
dtype: object
```

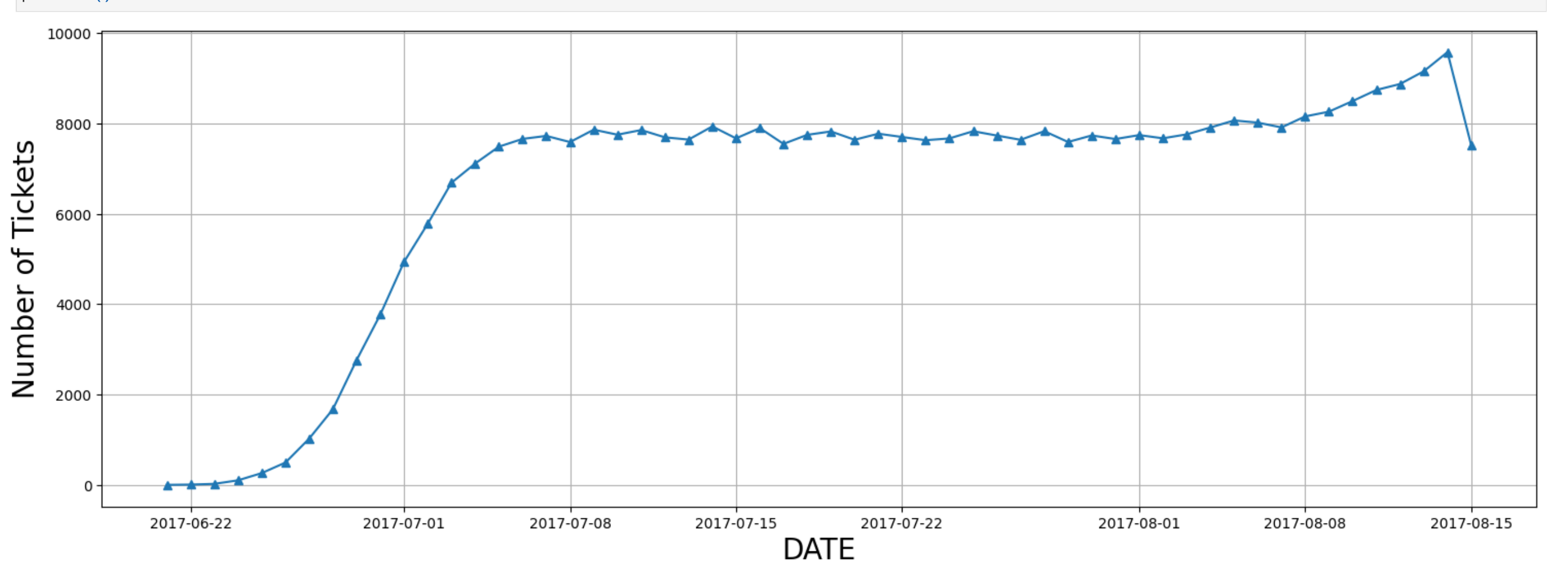
In [38]:
```python
tickets
```
Out[38]:

	ticket_no	book_ref	passenger_id	book_date	total_amount
0	0005432000987	06B046	8149 604011	2017-07-05 20:19:00+03:00	12400
1	0005432000988	06B046	8499 420203	2017-07-05 20:19:00+03:00	12400
2	0005432000989	E170C3	1011 752484	2017-06-29 01:55:00+03:00	24700
3	0005432000990	E170C3	4849 400049	2017-06-29 01:55:00+03:00	24700
4	0005432000991	F313DD	6615 976589	2017-07-03 04:37:00+03:00	30900
...	...	...	...	...	...
366728	0005435999869	D730BA	0474 690760	2017-08-14 11:50:00+03:00	210600
366729	0005435999870	D730BA	6535 751108	2017-08-14 11:50:00+03:00	210600
366730	0005435999871	A1AD46	1596 156448	2017-08-13 03:49:00+03:00	45900
366731	0005435999872	7B6A53	9374 822707	2017-08-15 15:54:00+03:00	219400
366732	0005435999873	7B6A53	7380 075822	2017-08-15 15:54:00+03:00	219400

366733 rows × 5 columns

In [39]:
```python
import matplotlib.pyplot as plt
```

In [40]:
```python
tickets['date']=tickets['book_date'].dt.date
x=tickets.groupby('date')[['date']].count()
plt.figure(figsize=(18,6))
plt.plot(x.index,x['date'],marker='^')
plt.xlabel('DATE',fontsize=20)
plt.ylabel('Number of Tickets',fontsize=20)
plt.grid('b')
plt.show()
```

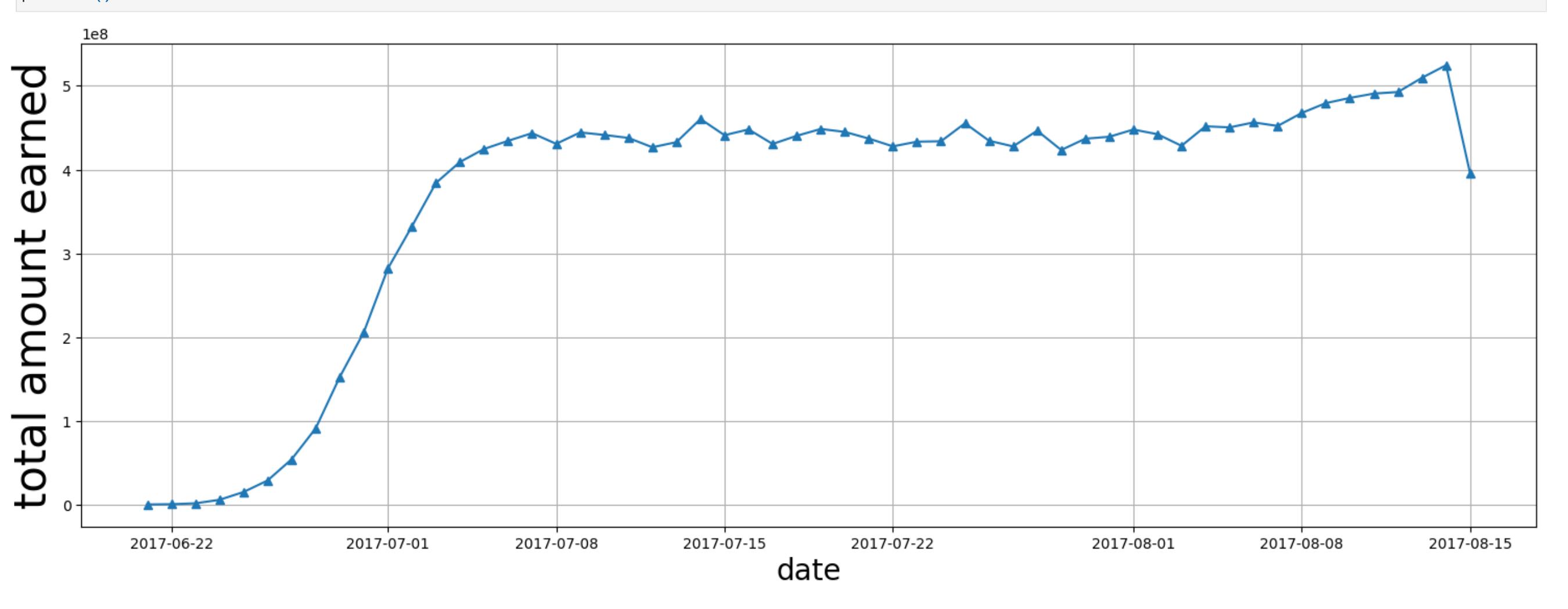


In [41]:
```python
bookings=pd.read_sql_query("select * from bookings",connections)
bookings
```
Out[41]:

	book_ref	book_date	total_amount
0	00000F	2017-07-05 03:12:00+03	265700
1	000012	2017-07-14 09:02:00+03	37900
2	000068	2017-08-15 14:27:00+03	18100
3	000181	2017-08-10 13:28:00+03	131800
4	0002D8	2017-08-07 21:40:00+03	23600
...	...	...	...
262783	FFFEF3	2017-07-17 07:23:00+03	56000
262784	FFFF2C	2017-08-08 05:55:00+03	10800
262785	FFFF43	2017-07-20 20:42:00+03	78500
262786	FFFFA8	2017-08-08 04:45:00+03	28800
262787	FFFFF7	2017-07-01 22:12:00+03	73600

262788 rows × 3 columns

In [42]:
```python
bookings=pd.read_sql_query("select * from bookings",connections)
bookings['book_date']=pd.to_datetime(bookings['book_date'])
bookings['date']=bookings['book_date'].dt.date
x=bookings.groupby('date')[['total_amount']].sum()
plt.figure(figsize=(18,6))
plt.plot(x.index,x['total_amount'],marker='^')
plt.xlabel('date',fontsize=20)
plt.ylabel('total amount earned',fontsize=30)
plt.grid('b')
plt.show()
```



In [43]:
```python
#CALCULATE THE AVERAGE CHARGE FOR EACH AIRCRAFT WITH DIFFERENT FARE CONDITIONS
```

In [57]:
```python
df=pd.read_sql_query("""select * from ticket_flights join flights on ticket_flights.flight_id= flights.flight_id""",connections)
```

In [58]:
```python
df
```
Out[58]:

	ticket_no	flight_id	fare_conditions	amount	flight_no	scheduled_departure	scheduled_arrival	departure_airport	arrival_airport	status	aircraft_code	actual_departure	actual_arr
0	0005432159776	30625	Business	42100	PG0013	2017-07-16 18:15:00+03	2017-07-16 20:00:00+03	AER	SVO	Arrived	773	2017-07-16 18:18:00+03	2017-07 20:04:00
1	0005435212351	30625	Business	42100	PG0013	2017-07-16 18:15:00+03	2017-07-16 20:00:00+03	AER	SVO	Arrived	773	2017-07-16 18:18:00+03	2017-07 20:04:00
2	0005435212386	30625	Business	42100	PG0013	2017-07-16 18:15:00+03	2017-07-16 20:00:00+03	AER	SVO	Arrived	773	2017-07-16 18:18:00+03	2017-07 20:04:00
3	0005435212381	30625	Business	42100	PG0013	2017-07-16 18:15:00+03	2017-07-16 20:00:00+03	AER	SVO	Arrived	773	2017-07-16 18:18:00+03	2017-07 20:04:00
4	0005432211370	30625	Business	42100	PG0013	2017-07-16 18:15:00+03	2017-07-16 20:00:00+03	AER	SVO	Arrived	773	2017-07-16 18:18:00+03	2017-07 20:04:00
...	...	...	...	...	...	...	...	...	...	...	...	...	
1045721	0005435097522	32094	Economy	5200	PG0708	2017-09-14 17:15:00+03	2017-09-14 18:00:00+03	SGC	OVS	Scheduled	733	\N	
1045722	0005435097521	32094	Economy	5200	PG0708	2017-09-14 17:15:00+03	2017-09-14 18:00:00+03	SGC	OVS	Scheduled	733	\N	
1045723	0005435104384	32094	Economy	5200	PG0708	2017-09-14 17:15:00+03	2017-09-14 18:00:00+03	SGC	OVS	Scheduled	733	\N	
1045724	0005435104352	32094	Economy	5200	PG0708	2017-09-14 17:15:00+03	2017-09-14 18:00:00+03	SGC	OVS	Scheduled	733	\N	
1045725	0005435104389	32094	Economy	5200	PG0708	2017-09-14 17:15:00+03	2017-09-14 18:00:00+03	SGC	OVS	Scheduled	733	\N	

1045726 rows × 13 columns

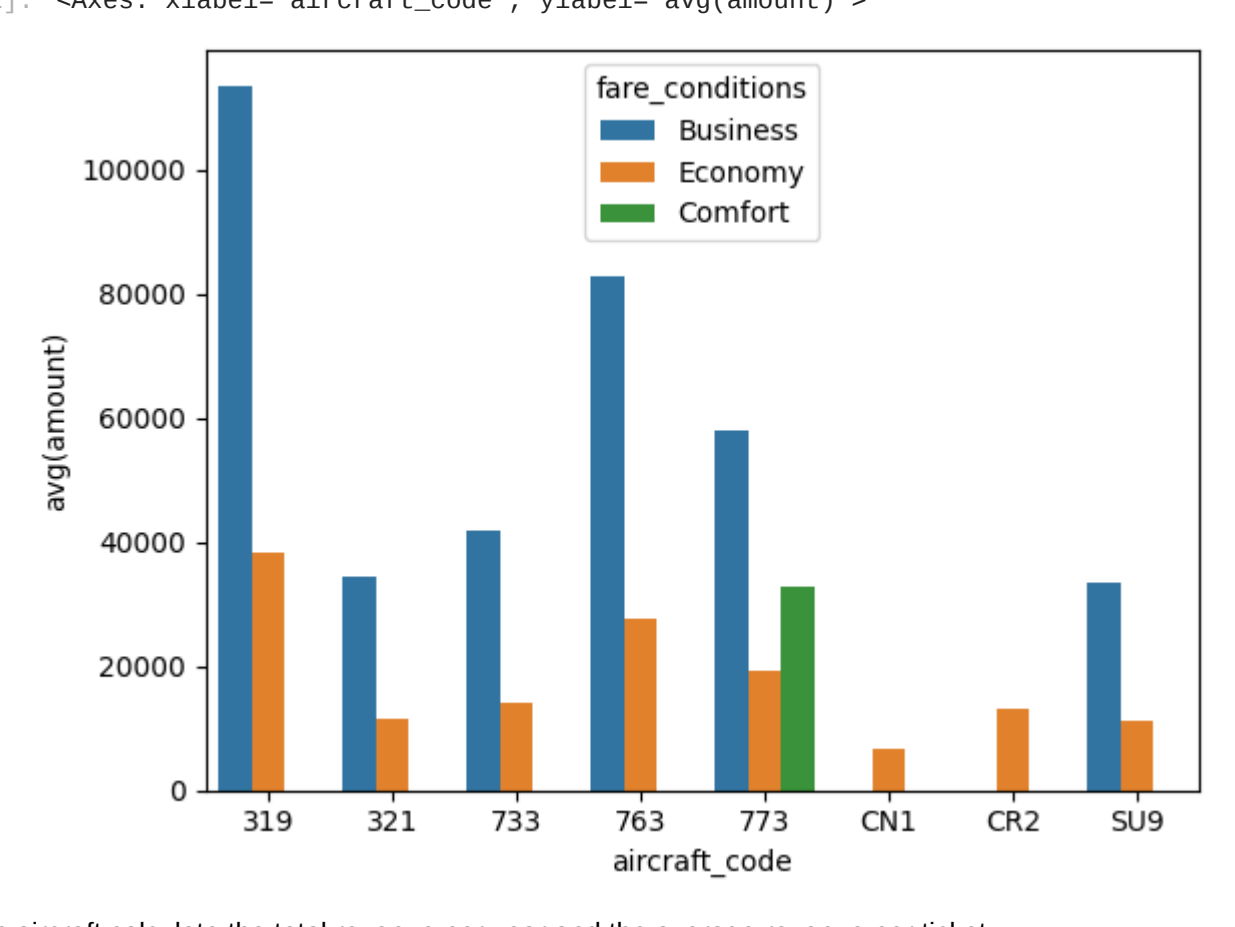
In [64]:
```python
df=pd.read_sql_query("""select fare_conditions,aircraft_code,avg(amount)
from ticket_flights join flights
on ticket_flights.flight_id=flights.flight_id group by aircraft_code,fare_conditions""",connections)
```

In [68]:
```python
df
```
Out[68]:

	fare_conditions	aircraft_code	avg(amount)
0	Business	319	113550.557703
1	Economy	319	38311.402347
2	Business	321	34435.662664
3	Economy	321	11534.974764
4	Business	733	41865.626175
5	Economy	733	13985.152000
6	Business	763	82839.842866
7	Economy	763	27594.721829
8	Business	773	57779.909435
9	Comfort	773	32740.552889
10	Economy	773	19265.225693
11	Economy	CN1	6568.552345
12	Economy	CR2	13207.661102
13	Business	SU9	33487.849829
14	Economy	SU9	11220.183400

In [70]:
```python
import seaborn as sns
```

In [71]:
```python
sns.barplot(data=df,x='aircraft_code',y='avg(amount)',hue='fare_conditions')
```
Out[71]: <Axes: xlabel='aircraft_code', ylabel='avg(amount)'>



for each aircraft,calculate the total revenue per year and the average revenue per ticket

In [75]:
```python
pd.read_sql_query("""select aircraft_code,ticket_count,total_revenue,total_revenue/ticket_count as avg_revenue_per_ticket from
(select aircraft_code,count(*) as ticket_count,sum(amount) as total_revenue from ticket_flights

join flights on ticket_flights.flight_id=flights.flight_id group by aircraft_code)""",connections)
```
Out[75]:

	aircraft_code	ticket_count	total_revenue	avg_revenue_per_ticket
0	319	52853	2706163100	51201
1	321	107129	1638164100	15291
2	733	86102	1426552100	16568
3	763	124774	4371277100	35033
4	773	144376	3431205500	23765
5	CN1	14672	96373800	6568
6	CR2	150122	1982760500	13207
7	SU9	365698	5114484700	13985

calculate the average occupancy per aircraft

In [79]:
```python
pd.read_sql_query("""select aircraft_code,flights.flight_id,count(*) as seats_count from boarding_passes
 inner join flights on boarding_passes.flight_id=flights.flight_id
  group by aircraft_code,flights.flight_id""") as a inner join("""select aircraft_code,count(*) as num_seats from seats
  group by aircraft_code""",connections)
```
Out[79]:

	aircraft_code	num_seats
0	319	116
1	320	140
2	321	170
3	733	130
4	763	222
5	773	402
6	CN1	12
7	CR2	50
8	SU9	97

In [ ]: