```python
In [1]:   import sqlite3   #for creating and interacting with a SQL database
          import matplotlib.pyplot as plt
          import pandas as pd
          import seaborn as sns
          import random
          from datetime import datetime, timedelta
```

```python
In [7]:   connect_data= sqlite3.connect('food_delivery.db')
          cursor= connect_data.cursor()
```

```python
In [37]:  cursor.executescript('''
          DROP TABLE IF EXISTS users;
          DROP TABLE IF EXISTS restaurants;
          DROP TABLE IF EXISTS orders;

          CREATE TABLE users(
              user_id INTEGER PRIMARY KEY,
              name TEXT,
              location TEXT
          );

          CREATE TABLE restaurants(
              restaurant_id INTEGER PRIMARY KEY,
              name TEXT,
              cuisine TEXT,
              city TEXT
          );

          CREATE TABLE orders(
              order_id INTEGER PRIMARY KEY,
              user_id INTEGER,
              restaurant_id INTEGER,
              timestamp TEXT,
              delivery_time INTEGER,  -- in minutes
              rating REAL,
              FOREIGN KEY (user_id) REFERENCES users (user_id),
              FOREIGN KEY (restaurant_id) REFERENCES restaurants (restaurant_id)
          );
          ''')

          locations= ['Mumbai', 'Delhi', 'Banglore', 'Pune']
          cuisines= ['Indian', 'Chinese', 'Italian', 'Mexican']
          restaurant_names= ['Spicy Bites', 'Food Fusion', 'Tandoori Treats', 'Noodle Hub', 'Cu
          user_names= ['Ayesha', 'Ravi', 'Fatema', 'Sidharth', 'Meera', 'Karan']

          for i in range(1,11):
              cursor.execute("INSERT INTO users VALUES (?, ?, ?)", (i, random.choice(user_names

          for i in range(1,6):
              cursor.execute("INSERT INTO restaurants VALUES (?, ?, ?, ?)", (i, random.choice(r

          for i in range(1,101):
              user_id= random.randint(1,10)
              restaurant_id= random.randint(1,5)
              delivery_time= random.randint(5,60)
              rating= round(random.uniform(2.5, 5.0), 1)
              timestamp=datetime.now() - timedelta(days= random.randint(0, 30))
              cursor.execute("INSERT INTO orders VALUES (?, ?, ?, ?, ?, ?)", (i, user_id, resta

          connect_data.commit()
```

```python
In [46]:  query1='''
          SELECT r.city, AVG(o.delivery_time) as avg_delivery
```

```
FROM orders o
JOIN restaurants r ON o.restaurant_id=r.restaurant_id
GROUP BY r.city
'''
df1=pd.read_sql(query1, connect_data)
df1
```

Out[46]:

	city	avg_delivery
0	Banglore	36.217391
1	Delhi	34.952381
2	Mumbai	35.515152
3	Pune	32.043478

In [54]:
```
query2='''
SELECT r.cuisine, COUNT(*) as total_orders
FROM orders o
JOIN restaurants r ON o.restaurant_id = r.restaurant_id
GROUP BY r.cuisine
ORDER BY total_orders DESC
'''

df2= pd.read_sql(query2, connect_data)
df2
```

Out[54]:

	cuisine	total_orders
0	Mexican	44
1	Chinese	23
2	Indian	19
3	Italian	14

In [56]:
```
query3='''
SELECT r.name, AVG(o.rating) as avg_rating
FROM orders o
JOIN restaurants r ON o.restaurant_id = r.restaurant_id
GROUP BY r.name
ORDER BY avg_rating DESC
LIMIT 5
'''

df3= pd.read_sql(query3, connect_data)
df3
```

Out[56]:

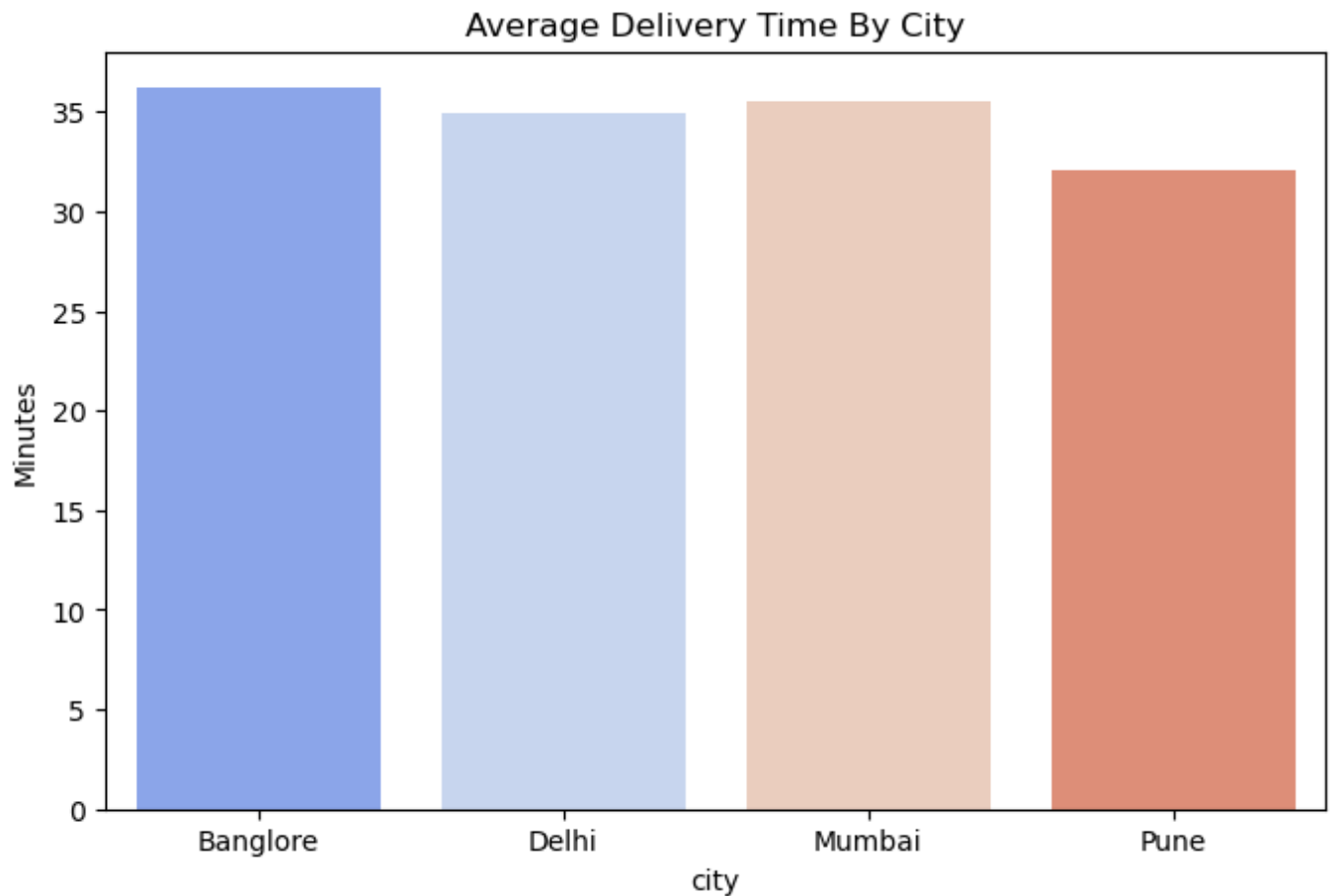|	name	avg_rating
0	Tandoori Treats	4.026087
1	Food Fusion	3.871429
2	Spicy Bites	3.760870
3	Curry Palace	3.726316
4	Noodle Hub	3.485714

In [78]:
```
plt.figure(figsize=(8,5))
sns.barplot(data=df1, x='city', y='avg_delivery', palette='coolwarm')
plt.title('Average Delivery Time By City')
```

```
plt.ylabel('Minutes')
plt.show()
```
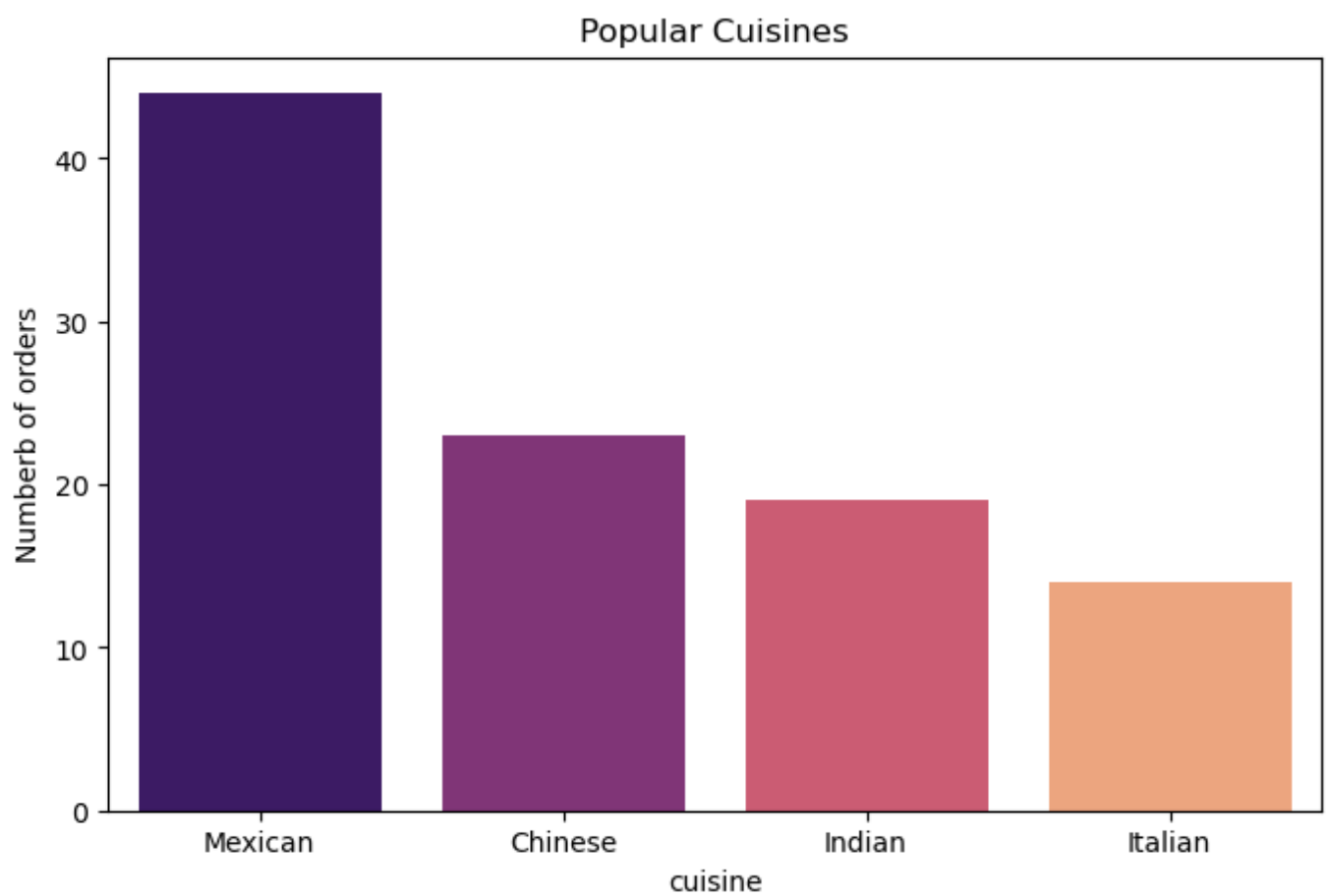
In [96]:
```
plt.figure(figsize=(8,5))
sns.barplot(data=df2, x='cuisine', y='total_orders', palette='magma')
plt.title('Popular Cuisines')
plt.ylabel('Numberb of orders')
plt.show()
```

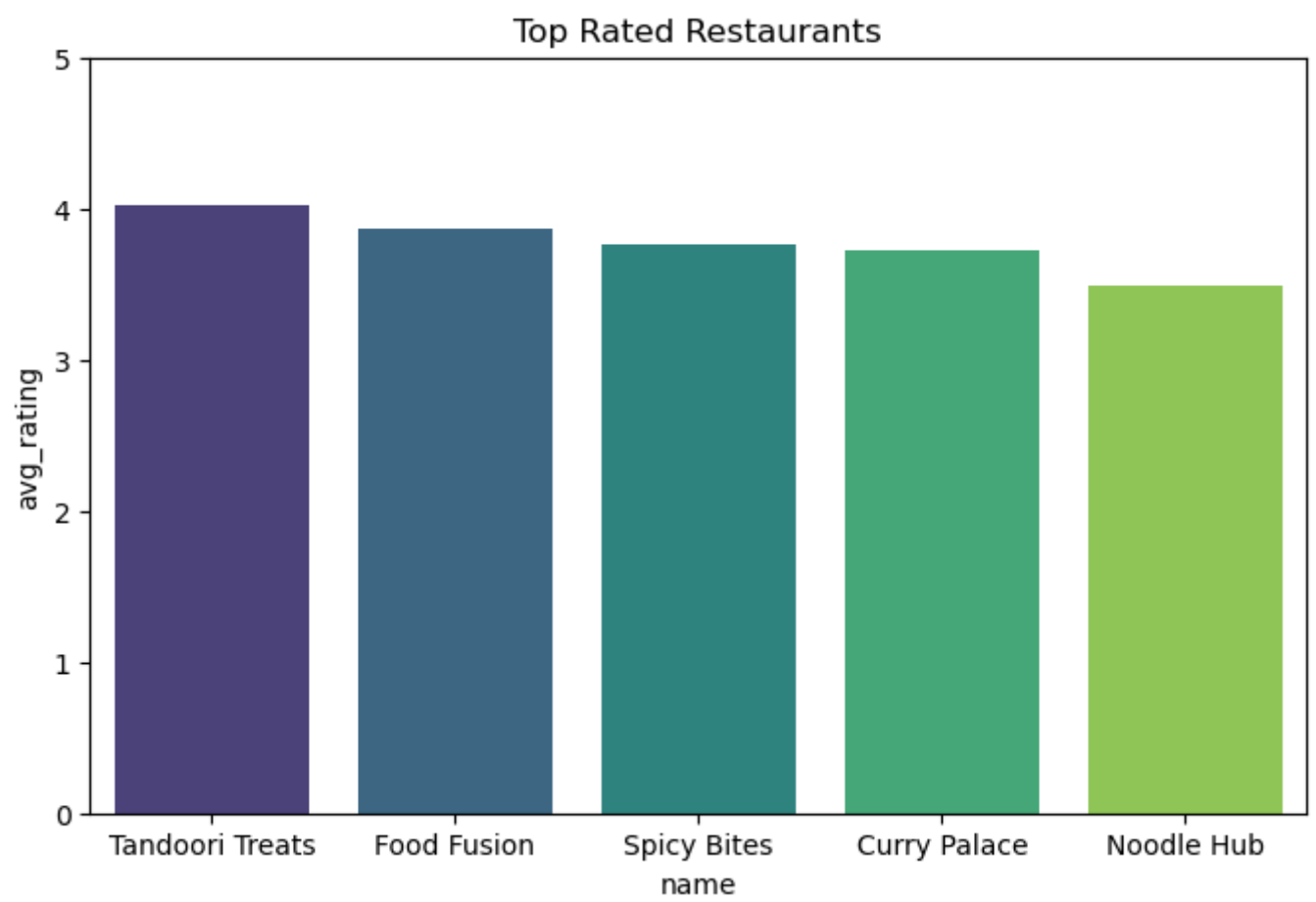Popular Cuisines

```
In [90]: plt.figure(figsize=(8,5))
         sns.barplot(data=df3, x='name', y='avg_rating', palette='viri')
         plt.title('Top Rated Restaurants')
         plt.ylim(0,5)
         plt.show()
```

```
/tmp/ipykernel_6750/1053545568.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(data=df3, x='name', y='avg_rating', palette='viridis')
```

Top Rated Restaurants

In [ ]: