

برای استفاده از دستورات `GROUP BY` و `HAVING` در پایتون و پایگاه داده‌ها مانند MySQL یا PostgreSQL، می‌توانیم داده‌ها را بر اساس یک یا چند ستون گروه‌بندی کرده و سپس با استفاده از دستور `HAVING`، فیلترهایی را پس از گروه‌بندی اعمال کنیم. در ادامه توضیح می‌دهم که چطور می‌توان این دستورات را در پایتون استفاده کرد.

## 1. گروه‌بندی داده‌ها با `GROUP BY`

دستور `GROUP BY` برای گروه‌بندی داده‌ها بر اساس یک یا چند ستون استفاده می‌شود. این دستور اغلب همراه با توابع تجمعی مانند `MIN()`، `MAX()`، `AVG()`، `SUM()`، `COUNT()` استفاده می‌شود تا اطلاعاتی مانند مجموع، میانگین یا تعداد هر گروه محاسبه شود.

### 1.1 مثال استفاده از `GROUP BY`

فرض کنید یک جدول به نام `orders` داریم که شامل اطلاعات سفارش‌ها و مبلغ آن‌ها است. می‌خواهیم تعداد سفارش‌ها و مجموع مبلغ هر مشتری را بدست آوریم:

```
import mysql.connector

# اتصال به پایگاه داده
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="password",
    database="your_database"
)

cursor = conn.cursor()

# محاسبه تعداد سفارش‌ها و مجموع مبلغ customer_id گروه‌بندی داده‌ها بر اساس
cursor.execute("""
    SELECT customer_id, COUNT(*) AS order_count, SUM(amount) AS total_amount
    FROM orders
    GROUP BY customer_id
""")
result = cursor.fetchall()

for row in result:
    print(row)
```

در این مثال:

- داده‌ها بر اساس ستون `customer_id` گروه‌بندی شده‌اند.
- برای هر گروه (یعنی هر مشتری)، تعداد سفارش‌ها و مجموع مبلغ سفارش‌ها محاسبه شده است.

## 2. فیلتر داده‌ها پس از گروه‌بندی با `HAVING`

دستور `HAVING` به شما این امکان را می‌دهد که داده‌های گروه‌بندی شده را فیلتر کنید. این دستور معمولاً پس از دستور `GROUP BY` استفاده می‌شود و به شما اجازه می‌دهد تا نتایج گروه‌بندی شده را براساس شرایط خاصی فیلتر کنید.

## 2.1. مثال استفاده از HAVING

فرض کنید می‌خواهیم فقط مشتریانی را نمایش دهیم که مجموع مبلغ سفارش‌هایشان بیشتر از 1000 واحد باشد:

```
cursor.execute("""
SELECT customer_id, COUNT(*) AS order_count, SUM(amount) AS total_amount
FROM orders
GROUP BY customer_id
HAVING SUM(amount) > 1000
""")
result = cursor.fetchall()

for row in result:
    print(row)
```

در این مثال:

- ابتدا داده‌ها بر اساس `customer_id` گروه‌بندی شده‌اند.
- سپس با استفاده از دستور `HAVING`، فقط مشتریانی که مجموع مبلغ سفارش‌هایشان بیشتر از 1000 واحد است، نمایش داده می‌شوند.

## 3. تفاوت WHERE و HAVING

- `WHERE`: برای فیلتر کردن داده‌ها قبل از گروه‌بندی استفاده می‌شود.
  - `HAVING`: برای فیلتر کردن داده‌ها پس از گروه‌بندی استفاده می‌شود.
- اگر بخواهید داده‌ها را قبل از گروه‌بندی فیلتر کنید، از `WHERE` استفاده می‌کنید. اگر بخواهید فیلترهایی پس از گروه‌بندی اعمال کنید، از `HAVING` استفاده می‌کنید.

### 3.1. مثال تفاوت WHERE و HAVING

فرض کنید می‌خواهیم فقط سفارش‌هایی که مبلغشان بیشتر از 100 را دارند را گروه‌بندی کنیم و سپس مجموع مبلغ‌های هر گروه را فیلتر کنیم که مجموع آن بیشتر از 1000 واحد باشد.

```
cursor.execute("""
SELECT customer_id, COUNT(*) AS order_count, SUM(amount) AS total_amount
FROM orders
WHERE amount > 100
GROUP BY customer_id
HAVING SUM(amount) > 1000
""")
result = cursor.fetchall()

for row in result:
    print(row)
```

در اینجا:

- ابتدا از دستور `WHERE` برای فیلتر کردن سفارش‌ها با مبلغ بیشتر از 100 استفاده شده است.
- سپس داده‌های فیلتر شده گروه‌بندی شده‌اند.
- بعد از گروه‌بندی، دستور `HAVING` برای فیلتر کردن گروه‌ها با مجموع مبلغ بیشتر از 1000 استفاده شده است.

## 4. استفاده از `GROUP BY` و `HAVING` در PostgreSQL

در PostgreSQL هم نحوه استفاده از `GROUP BY` و `HAVING` مشابه MySQL است.

### 4.1. اتصال به PostgreSQL

```
import psycopg2

# اتصال به پایگاه داده
conn = psycopg2.connect(
    dbname="your_database",
    user="postgres",
    password="password",
    host="localhost"
)

cursor = conn.cursor()
```

### 4.2. گروه‌بندی و فیلتر کردن داده‌ها

```
cursor.execute("""
    SELECT customer_id, COUNT(*) AS order_count, SUM(amount) AS total_amount
    FROM orders
    GROUP BY customer_id
    HAVING SUM(amount) > 1000
""")
result = cursor.fetchall()

for row in result:
    print(row)
```

## 5. نتیجه‌گیری

در پایتون و هنگام کار با پایگاه داده‌ها (چه MySQL و چه PostgreSQL)، می‌توانید از دستورات `GROUP BY` و `HAVING` برای گروه‌بندی داده‌ها و سپس فیلتر کردن نتایج گروه‌بندی شده استفاده کنید. این دو دستور به شما کمک می‌کنند تا داده‌ها را به شکلی مؤثر و بهینه تجزیه و تحلیل کنید.