

آیا بازیافت کننده حافظه وجود دارد و در صورت وجود چگونه پیاده سازی شده است. در صورتی که بازیافت کننده حافظه وجود ندارد، زبان با یک زبان دیگر که بازیافت کننده حافظه دارد مقایسه شود.

بازیافت کننده حافظه یا **Garbage Collector** یک مکانیزمی است که به صورت خودکار حافظه‌ای را که توسط برنامه‌ها در حال استفاده نیست، آزاد می‌کند. این کار باعث می‌شود که برنامه‌نویس نیازی نداشته باشد به صورت دستی حافظه را مدیریت کند و از اتلاف و نشت حافظه جلوگیری شود.

زبان **SQL** به طور مستقیم از بازیافت کننده حافظه استفاده نمی‌کند، زیرا این زبان برای پرس و جو و دستکاری داده‌های موجود در پایگاه‌های داده رابطه‌ای طراحی شده است و نه برای مدیریت حافظه. **SQL** یک زبان دکلاراتیو است که به برنامه‌نویس اجازه می‌دهد که بگوید چه چیزی را می‌خواهد و نه چگونه آن را بدست آورد. بنابراین، جزئیات پیاده‌سازی و مدیریت حافظه به عهده سیستم مدیریت پایگاه داده است که **SQL** را اجرا می‌کند.

به عنوان مثال، اگر از **MySQL** برای اجرای **SQL** استفاده کنیم، می‌توانیم از دستوراتی مانند **SHOW ENGINE INNODB STATUS** یا **SHOW ENGINE PERFORMANCE_SCHEMA STATUS** برای مشاهده وضعیت حافظه و عملکرد موتور پایگاه داده استفاده کنیم. این دستورات به ما اطلاعاتی مانند حافظه مصرفی، حافظه آزاد، تعداد تراکنش‌ها و غیره را نشان می‌دهند. اما این دستورات جزئی از زبان **SQL** نیستند و بستگی به نوع موتور پایگاه داده دارند.

از طرف دیگر، برخی از زبان‌های برنامه‌نویسی مانند جاوا، پایتون، روبی و غیره از بازیافت کننده حافظه به صورت خودکار استفاده می‌کنند. این زبان‌ها به عنوان زبان‌های امپراتیو شناخته می‌شوند که به برنامه‌نویس اجازه می‌دهند که چگونگی انجام کار را مشخص کند. این زبان‌ها می‌توانند با استفاده از کتابخانه‌ها یا درایورهای مختلف با پایگاه‌های داده رابطه‌ای ارتباط برقرار کنند و دستورات **SQL** را اجرا کنند. اما در هنگام نوشتن برنامه، برنامه‌نویس نیازی ندارد که به

صورت دستی حافظه را آزاد کند، زیرا بازیافت کننده حافظه این کار را به صورت پشت پرده انجام می‌دهد.

به عنوان مثال، در زبان جاوا، می‌توانیم از کلاس `java.sql.Connection` برای برقراری ارتباط با پایگاه داده و اجرای دستورات SQL استفاده کنیم. این کلاس یک شیء را ایجاد می‌کند که حافظه‌ای را برای ذخیره اطلاعات مربوط به ارتباط اختصاص می‌دهد. اما وقتی این شیء دیگر مورد استفاده قرار نگیرد، بازیافت کننده حافظه جاوا این حافظه را آزاد می‌کند و از اتلاف و نشت حافظه جلوگیری می‌کند.

به این ترتیب، می‌توان گفت که زبان SQL و بازیافت کننده حافظه دو مفهوم متفاوت هستند که برای اهداف مختلف طراحی شده‌اند. SQL برای کار با داده‌های رابطه‌ای و بازیافت کننده حافظه برای مدیریت حافظه در زبان‌های برنامه‌نویسی استفاده می‌شود.

#دکلاراتیو (declarative) بودن یک زبان برنامه نویسی به چه معناست؟

دکلاراتیو بودن یک زبان برنامه‌نویسی به این معناست که زبان برنامه‌نویسی به برنامه‌نویس اجازه می‌دهد که بگوید چه چیزی را می‌خواهد و نه چگونه آن را بدست آورد. به عبارت دیگر، زبان برنامه‌نویسی دکلاراتیو تمرکز خود را بر روی هدف یا نتیجه قرار می‌دهد و جزئیات پیاده‌سازی یا روش را به عهده مفسر یا کامپایلر می‌گذارد. برای مثال، زبان SQL یک زبان برنامه‌نویسی دکلاراتیو است که برای کار با داده‌های رابطه‌ای طراحی شده است. در SQL، می‌توان با استفاده از دستوراتی مانند `SELECT`، `INSERT`، `UPDATE` و `DELETE` پرس و جوها و دستکاری‌های مورد نظر را بر روی داده‌ها انجام داد، بدون اینکه نیاز باشد که بگوییم چگونه این کارها را انجام دهیم. مثلاً برای انتخاب تمام رکوردهایی که در جدول `customers` وجود دارند، می‌توان از دستور زیر استفاده کرد:

```
SELECT * FROM customers;
```

این دستور به ما می گوید که چه چیزی را می خواهیم (تمام رکوردهای جدول customers) ولی نمی گوید که چگونه آن را بدست آوریم. این کار به عهده سیستم مدیریت پایگاه داده است که SQL را اجرا می کند و می تواند از الگوریتم های مختلفی برای انجام این پرس و جو استفاده کند. بنابراین، SQL یک زبان برنامه نویسی دکلاراتیو است که تنها هدف را مشخص می کند و جزئیات را پنهان می کند.

#امپراتیو (Imperative) بودن یک زبان برنامه نویسی به چه معناست؟

برخلاف زبان های برنامه نویسی دکلاراتیو، زبان های برنامه نویسی امپراتیو به برنامه نویس اجازه می دهند که بگویند چگونه چیزی را بدست آورد. به عبارت دیگر، زبان برنامه نویسی امپراتیو تمرکز خود را بر روی روش یا الگوریتم قرار می دهد و جزئیات هدف یا نتیجه را به عهده برنامه نویس می گذارد. برای مثال، زبان C یک زبان برنامه نویسی امپراتیو است که برای کار با داده های سطح پایین و مدیریت حافظه طراحی شده است. در C، می توان با استفاده از دستوراتی مانند `for`، `if`، `switch` و `malloc` عملیات مورد نظر را بر روی داده ها انجام داد، با این شرط که بگوییم چگونه این کارها را انجام دهیم.

منابع:

- <https://dev.mysql.com>
- <https://docs.oracle.com>
- <https://azaronline.com>