

امین آقامحمدلو

سوال اول

حجم حافظه ذخیره سازی مورد نیاز برای یک نوع متغیر پایگاه داده به عوامل مختلفی از جمله سیستم مدیریت پایگاه داده (DBMS) خاص، نوع داده و مقادیر ذخیره شده بستگی دارد.

با این حال، می توان برخی از تعمیمات کلی را انجام داد:

انواع عددی:

INT: معمولا ۴ بایت حافظه را اشغال می کند و برای ذخیره اعداد صحیح از -۲,۱۴۷,۴۸۳,۶۴۷ تا ۲,۱۴۷,۴۸۳,۶۴۸ استفاده می شود.  
BIGINT: 8 بایت حافظه را اشغال می کند و برای ذخیره اعداد صحیح بزرگتر از -۲,۱۴۷,۴۸۳,۶۴۷ استفاده می شود.  
11:16

FLOAT: چهار بایت حافظه را اشغال می کند و برای ذخیره اعداد اعشاری با دقت تک دقت استفاده می شود.

DOUBLE: هشت بایت حافظه را اشغال می کند و برای ذخیره اعداد اعشاری با دقت دو برابر استفاده می شود.  
انواع رشته ای:

VARCHAR: حافظه مورد نیاز برای یک VARCHAR به طول رشته بستگی دارد.  
حداکثر طول یک VARCHAR می تواند ۲۵۵ کاراکتر در MySQL و ۸۰۰۰ کاراکتر در SQL Server باشد.

NVARCHAR: شبیه VARCHAR است، اما برای ذخیره کاراکترهای یونیکد طراحی شده است.

TEXT: برای ذخیره رشته های متنی طولانی تر استفاده می شود. حداکثر طول یک TEXT می تواند ۶۵۵۳۵ بایت در MySQL و ۲۱۴۷۴۸۳۶۴۷ بایت در SQL Server باشد

BLOB: برای ذخیره داده های باینری مانند تصاویر و فایل ها استفاده می شود.  
حداکثر طول یک BLOB می تواند ۶۵۵۳۵ بایت در MySQL و ۲۱۴۷۴۸۳۶۴۷ بایت در SQL Server باشد..

سوال دوم

مقادیر: enumeration معمولا به عنوان value type در نظر گرفته می شوند و در stack تخصیص داده می شوند.

record: نوع record می تواند به عنوان value type یا reference type در نظر گرفته شود.

اگر به عنوان value type در نظر گرفته شود، در stack تخصیص داده می شود.  
اگر به عنوان reference type در نظر گرفته شود، در heap تخصیص داده می شود.

struct: struct ها معمولا به عنوان value type در نظر گرفته می شوند و در stack تخصیص داده می شوند.

tuple: tuple ها معمولا به عنوان value type در نظر گرفته می شوند و در stack تخصیص داده می شوند.

سوال سوم

حافظه Stack:

نحوه تخصیص: حافظه stack به طور خودکار توسط کامپایلر یا مفسر در زمان اجرای برنامه مدیریت می شود.

ذخیره سازی: متغیرهای محلی، آرگومان های تابع و مقادیر بازگشتی تابع در حافظه stack ذخیره می شوند.

دسترسی: دسترسی به داده ها در حافظه stack بسیار سریع است.

مدیریت حافظه: حافظه stack به طور خودکار در زمان خروج از یک تابع یا بلوک کد آزاد می شود.

معایب: حافظه stack فضای محدودی دارد و می تواند در صورت استفاده بیش از حد منجر به خطای stack overflow شود.

### حافظه Heap:

نحوه تخصیص: حافظه heap به طور دستی توسط برنامه نویس با استفاده از اپراتورهای new یا malloc (در C) تخصیص داده می شود. ذخیره سازی: اشیاء، ساختارها، آرایه ها و هر نوع داده پویا دیگری که توسط برنامه نویس ایجاد می شود در حافظه heap ذخیره می شوند. دسترسی: دسترسی به داده ها در حافظه heap کمی کندتر از حافظه stack است مدیریت حافظه: حافظه heap به طور خودکار آزاد نمی شود و باید توسط برنامه نویس با استفاده از اپراتورهای delete یا free (در C) آزاد شود. عدم آزادسازی حافظه heap می تواند منجر به نشت حافظه شود. مزایا: حافظه heap فضای نامحدودی دارد و می توان در صورت نیاز به راحتی آن را افزایش داد

	ویژگی	حافظه Stack	حافظه heap
۱	تخصیص	خودکار	دستی
۲	ذخیره سازی	متغیر های محلی	اشیاء و ساختار ها
۳	دسترسی	سریع	کمی کند تر
۴	مدیریت حافظه	خودکار	دستی
۵	محدودیت	فضای محدود	فضای نامحدود
۶	خطرات احتمالی	Stack overflow	نشت حافظه