« به نام راستگوی بی همتا »

مبانی کامپیوتر و برنامه‌سازی

پروژه نهایی

دکتر هاشمی و دکتر مرادی

پاییز ۹۳

پایگاه‌داده

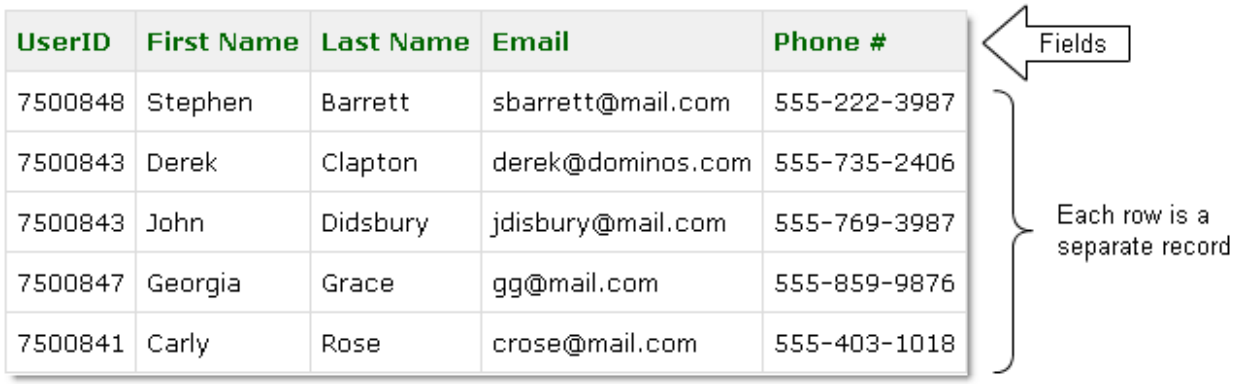
مقدمه

هر برنامه‌ای پس از اجرا با دریافت و دستکاری مداوم اطلاعات سروکار دارد و یک سری خروجی تولید می کند که بعضی از این اطلاعات باید نگهداری شود؛ چرا که نتیجه‌ی محاسبات هستند یا خودشان ورودی بخش‌های دیگری از این برنامه یا برنامه‌های دیگرند. این ذخیره سازی اطلاعات باید ویژگی هایی داشته باشد تا بتوان به داده های ذخیره شده اعتماد و از آنها استفاده کرد. به طور مثال این ذخیره‌سازی باید به نحوی باشد که بازیابی آن به راحتی و با سرعت بالا امکان پذیر شود چرا که برنامه های دیگری در فازهای مختلفی از اجرا به این اطلاعات نیاز دارند که باید به محض دریافت درخواست، در زمان مناسب این اطلاعات در اختیارشان قرار گیرد. همچنین پس از مدت طولانی که حجم داده‌ها زیاد می شود، باید همچنان سرعت و کارایی قابل قبولی داشته باشند. تصور کنید سرورهای شرکت های بزرگ مثل گوگل در کسری از ثانیه مقدار قابل ملاحظه ای داده را بازیابی و ذخیره می کنند. به عبارت دیگر اضافه، حذف و جست‌وجو کردن داده ها باید با سرعت بالاو اطمینان امکان پذیر باشد. چالش دیگری که در نگهداری داده ها با آن مواجه هستیم، حفاظت از داده ها در مقابل خراب شدن به دلیل اتفاقات غیر قابل پیش بینی (مثل قطعی برق، حملات خرابکارانه و...) و ایجاد امکان بازیابی در صورت وقوع است. این خراب شدن ممکن است در هنگام ذخیره سازی، تغییر یک داده و یا پس از اینکه داده در حافظه قرار گرفت، اتفاق بیافتد. فرض کنید شما در حال ذخیره یک سری داده هستید که برق دستگاه شما قطع می شود. در این حالت اگر به داده ها آسیبی وارد شود(به عنوان مثال تنها نصفی از آنها ذخیره شوند)،

دیگر قابل استفاده نیستند. هزینه‌هایی را متصور شوید که شرکت‌ها، بانک‌ها و ... در صورت بروز چنین مشکلی خواهند پرداخت. روش‌های مختلفی برای مدیریت پایداری داده‌ها هست و معمولا در این مواقع داده ها به آخرین وضعیت پایدار بازگردانده می شوند. که این امر معمولا با نگه‌داشتن پشتیبان و ذخیره کردن سلسله تغییراتی که بر داده‌ها اعمال شده امکان پذیر می‌شود. مشکل دیگر مدیریت دسترسی است. فرض کنید یک سری داده قرار است بین چند نرم افزار که توسط چند برنامه نویس مختلف نوشته شده است به صورت مشترک استفاده شود. برای اینکه همه آن ها بتوانند از این داده‌ها استفاده کنند، باید نحوه خواندن و نوشتن اطلاعات با یک استاندارد از قبل تعریف شده باشد تا بی نظمی در ساختار داده ها به وجود نیاید. تصور کنید اگر هر برنامه ای روشی مخصوص به خود را برای ذخیره سازی داده ها داشته باشد، استفاده مشترک از داده ها بسیار مشکل خواهد بود. همچنین در این صورت، تمامی مواردی که در بالا ذکر شد (پایداری داده ها، امکان پشتیبان‌گیری و ...) باید توسط تک تک نرم افزارهای نوشته شده به صورت مجزا پیاده سازی و مدیریت شود!

پس درواقع برای نگهداری داده ها نیازمند یک ساختار منظم و سامان مند هستیم. پایگاه‌های‌داده، ساخته شده‌اند تا این نیاز را برآورده کنند، یعنی با حجم بسیار زیادی از داده ها کار کرده، قابلیت ذخیره سازی، بازیابی و مدیریت کردن آن ها را به همراه تمامی نکاتی که در بالا ذکر شده فراهم نمایند. در اینجا به تعریف پایگاه داده می پردازیم.

پایگاه‌داده مجموعه‌ای از رکوردهای ذخیره شده در رایانه با یک روش سیستماتیک است. پایگاه‌های داده از فیلدها، رکوردها و فایل‌ها ساخته می شوند. به عنوان مثال یک دفترچه تلفن یک پایگاه داده است که لیستی از رکوردها را نگه‌داری می‌کند که هر رکورد سه فیلد نام، آدرس و شماره تلفن را شامل می‌شوند.



برنامه رایانه‌ای که برای مدیریت و پرسش‌وپاسخ بین پایگاه‌های‌داده‌ای و برنامه های دیگر استفاده می‌شود را سیستم مدیریت پایگاه داده‌ای یا به اختصار (DBMS[1]) می‌نامیم.

درواقع DBMS درخواست نرم افزار ها را مدیریت می کند و داده هایی که نیاز دارند را از پایگاه داده در اختیار آنها قرار می دهد بدون اینکه نرم افزار ها بدانند که داده ها به صورت فیزیکی در کجا و چگونه نگهداری می شوند. همچنین DBMS ها پایداری، امنیت و دیگر ویژگی های یک پایگاه داده را تضمین می کنند. IBM DB2 ,MongoDB و OraclePostgreSQL از نمونه DBMS های موفق هستند.

نرم افزار ها باید بتوانند با استاندارد زبانی خاصی با DBMS صحبت کنند. این صحبت کردن برای ایجاد نوعی مجموعه داده یا ایجاد، تغییر و حذف یک رکورد و یا گرفتن داده ها از پایگاه داده نیاز است. به این زبان، زبان جست وجو(Query Language) گفته می شود. در سال ۱۹۸۶ زبان SQL که نقش هر ۳ نوع زبان بالا را بازی می کرد به عنوان یک استاندارد در ANSI قرار گرفت.

پس به طور کلی می توان گفت یک پایگاه داده، مجموعه ای شامل رکورد های مختلف است درصورتی که یک DBMS، سیستمی است که مجموعه ای از این پایگاه های داده را نگه داری و مدیریت می کند و امکان برقراری ارتباط با نرم افزار های دیگر را فراهم می کند. همانطور که یک فایل سیستم، فایل های مختلف را در یک سیستم عامل مدیریت می کند، DBMS هم پایگاه های داده‌ای مختلف را مدیریت می کند.

[1] Database Manegement System

گاهی اوقات برای راحتی واژه دیتابیس بجای DBMS استفاده می شود.

نگهداری داده ها

این که داده ها بر روی دیسک سخت یا حافظه اصلی کامپیوتر نگه داری شوند، DBMS ها را به ۲ دسته مهم Memory-Resident Databases و On-Disk Databases تقسیم می کند.

در پایگاه های داده ای که به صورت On-Disk کار می کنند، داده ها در دیسک سخت نگهداری می شوند. سرعت این دیسک ها بسیار پایین تر از حافظه اصلی کامپیوتر است. برای بالا بردن سرعت، معمولا از تکنیک نگهداری موقت داده ها در حافظه اصلی (Caching)، بافر کردن و شاخص‌گذاری[۲] استفاده می شود. دسترسی به دیسک ۱۰۰۰۰۰بار کندتر از دسترسی به حافظه اصلی است و CPUهای معمولی امروزه می‌توانند در زمان یک دسترسی به دیسک، تعدادی بسیار بیش‌تر از ۱میلیون پردازش را انجام دهند. با توجه با این معلوم می‌شود که گلوگاه (bottleneck)[۳] در دیتابیس‌ها دسترسی به دیسک است و به دلیل این زیاد بودن هزینه‌ی دسترسی به دیسک سخت در مقابل دسترسی به حافظه‌ی اصلی،(و انجام محاسبات توسط CPU)؛ طراحان سعی می‌کنند با روش‌های مختلف، از CPU cycle و حافظه‌ی اصلی هزینه بدهند تا میزان دسترسی به دیسک را کاهش پیدا کند و کارایی مطلوب حاصل شود.

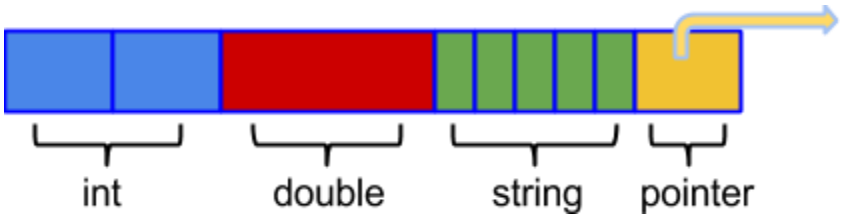
پایگاه های داده In-Memory داده ها را در حافظه اصلی کامپیوتر (RAM) نگه داری می کنند. این موضوع باعث می شود خواندن و نوشتن با سرعت بسیار بالاتری انجام شود. این نوع از پایگاه های داده در چند سال اخیر با توجه به ارزان شدن حافظه های اصلی و اهمیت سرعت بالا ، بسیار مورد توجه قرار گرفته و رقابت میان شرکت‌های بزرگ صنعت پایگاه داده مانند oracle، IBM، SAP و ... در این زمینه مدتی است که آغاز شده

[2] indexing
[3] بخشی از یک برنامه که بیشترین تاثیر را روی سرعت اجرا و کارایی دارد

است. و مسائل مربوط نیزاز جمله عناوین داغ تحقیقاتی در مباحث پایگاه‌داده هستند. در این نوع از پایگاه هـای داده، ۲ نکته قابل توجه است:

۱. چون همه داده ها در حافظه اصلی نگهداری می شوند، نیاز به حافظه زیاد است و همانطور کـه مـی دانید، تهیه حافظه اصلی با حجم بالا گران قیمت تر از تهیه همان حجم با دیسک سخت است.

۲. یکی از ویژگی هایی که برای نگه داری داده ها گفته شد، پایداری آن ها در مقابل اتفاقات اسـت. در این نوع از DBMS ها اگر برق قطع شود، تمام داده ها از دست می روند. مدیریت این موضـوع یکـی از نکاتی است که باید در طراحی این نوع از DBMS ها مدنظر باشد. (لازم به ذکر است که اخیرا چنـین حافظه‌هایی با قابلیت حفظ اطلاعات حتی درصورت نبود برق نیز معرفی شده‌اند[۴])



داده‌هـا در ایـن نـوع از DBMS هـا بـه صورت بخشی از حافظه‌ی اصلی بـوده و می‌توان آن‌ها را به صـورت لیسـت‌هـای پیوندی نگه‌داری کرد. شکل فوق، به عنوان مثال، یک رکورد را نشان می‌دهد که دارای فیلد های مختلفی است. در آخرین قسمت مربوط به این رکورد، اشاره‌گری به رکورد بعدی قرار می گیرد. حال فرض کنید مـی خـواهیم رکوردی که در قسمت string آن "ICSP" نوشته شده است را پیدا کنیم، با توجه به خاصیتی که لیست هـای پیوندی دارن، باید از اولین گره شروع کنیم و به جلو حرکت کنیم. فرض کنید در بین ۱ میلیارد رکورد دنبال ۱ رکورد خاص هستیم. با توجه به نکاتی که در بالا گفته شد، سرعت بالا در زمان ذخیره سازی و بازیـابی یکـی از نکات بسیار مهم است. استفاده از ایندکس و Hash Table که در فاز قبلی پیـاده سـازی کردیـد اینجـا بسـیار اهمیت پیدا می کند. فرض کنید نرم افزاری در بانک اطلاعاتی و در میان مجموعه داده‌هایی که تمـامی رکـورد

[4] non-volatile random access memory

های آن ساختار بالا را دارند، دنبال همان کلمه "ICSP" می گردند. به میسر کردن این دسترسی سریع به کمک ساختارداده‌های کمکی را ایندکسینگ [5](فهرست‌بندی یا شاخص‌گذاری) گفته می شود. روش های مختلفی برای ایندکسینگ وجود دارد. یکی از انواع ایندکس گذاری، Hash indices است. در پایگاه های داده با توجه به استفاده ای که هر فایلی دارد، برروی بعضی از فیلدهای آن ایندکسینگ را فعال می کنند. فرض کنید در مثال بالا، می دانیم جست‌وجو و بازیابی اطلاعات با شرط اسم خاص بسیار پر کاربرد است. اگر یک Hash Table برای این فیلد و برای این مجموعه رکورد خاص ساخته شود، در زمان بازیابی فقط کافیست مقدار Hash متناظر کلمه ای که دنبال آن هستیم را پیدای کنیم و مستقیما به آن آدرس مراجعه کنیم.

در DBMS هایی که بر پایه حافظه اصلی هستند، با توجه به اینکه تمامی داده ها در حافظه نگهداری می شوند، لازم است که به نحوی از آنها در مقابل خرابی و قطعی برق حفاظت کنیم. برای اینکار در زمان‌هایی مشخص یک تصویر [6] (کپی) از پایگاه داده را در دیسک ذخیره می‌کنند (این فرآیندها بسیار زمان‌بر هستند و معمولا به صورت مستقل و به صورت موازی با DBMS انجام می‌شوند؛ پیچیدگی‌های مربوط به این مساله از جمله مسائلی است که همچنان مورد تحقیق است) و تا زمان ذخیره‌سازی بعدی queryهایی که داده‌ها را تغییر داده‌اند ذخیره می‌کنند[7]. به این طریق در صورت از دست رفتن داده‌ها می‌توان با بارگذاری مجدد آخرین snapshot و اجرای دوباره‌ی این تغییرات بخش اعظم داده‌ها را بازیابی کرد.

آنچه که شما در نهایت در پروژه‌ی نهایی خود پیاده‌سازی میکنید درواقع یک کتابخانه برای کمک به مدیریت داده‌ها در برنامه‌های به زبان c است که از جهات بسیاری یادآور یک پیاده‌سازی بسیار ساده از سیستم پایگاه داده ساکن در حافظه‌ی اصلی است. البته با تفاوت‌های اساسی از جمله اینکه این سیستم، یک فرآیند مستقل

[5] indexing
[6] snapshot
[7] logging

نبوده و به عنوان بخشی از برنامه‌ی کاربر اجرا می‌شود (یک کتابخانه است) و کاربر با اجـرای توابـع کتابخانـه‌ی شما بر روی ساختارهای مربوط انبوه داده‌های خود را مدیریت می‌کند.