

« بسمه تعالی »

آشنایی با مدل

برنامه نویسی سه لایه

(3-Tier Programming)

تهیه کننده :

مجید محمدی سامانی

ارتباط با من :

majid.mohamadisamani@gmail.com

برنامه نویسی چند لایه

۱. تعریف لایه و مفاهیم آن

۲. برنامه نویسی دو لایه

• بررسی و مشکلات برنامه های دو لایه

۳. برنامه نویسی سه لایه

۴. برنامه نویسی چند لایه

تعریف لایه و مفاهیم آن :

یک **لایه برنامه (Application Tier)** به یک قسمت از برنامه که به صورت موازی با دیگر قسمت ها در حال تعامل است گفته می شود. و هر لایه وظیفه خاص خود را انجام می دهد.

در واقع برنامه نویسی چند لایه تقسیم فیزیکی برنامه به چند قسمت و توزیع هر قسمت روی یک server می باشد.

مدل برنامه نویسی چند لایه مشکلات مدل های قدیمی مثل client/server که شامل مدیریت متمرکز ، محاسبات توزیع شده ، کارایی و مقیاس پذیری است را رفع نموده است.

مدیریت متمرکز:

در یک محیط که بصورت متمرکز توزیع می شود ، تغییرات مورد نظر در یک قسمت مرکزی انجام می شود و این تغییرات به سرتاسر سیستم توزیع می شوند.

محاسبات توزیع شده :

پردازش ها بین چند سیستم انجام می شود.

کارایی :

به تعداد سیکلهایی که یک پردازنده نیاز دارد تا اجرای یک فرمان یا متدی (مانند ذخیره داده ها یا بازیابی آنها) را به پایان برساند .

کارایی به تنهایی نمی تواند ملاک خوبی برای مدل چند لایه و راحتی کاربر باشد و مفهوم مقیاس پذیری در کنار آن لازم است.

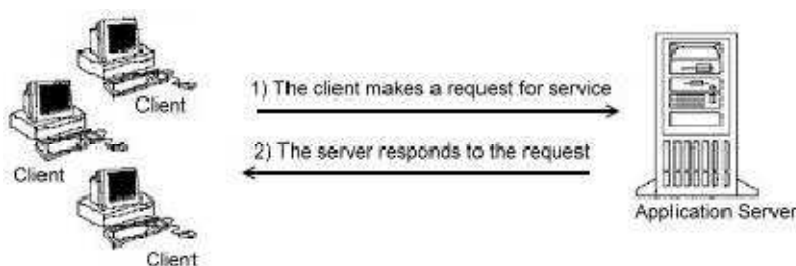
مقیاس پذیری :

منظور از مقیاس پذیر بودن یعنی تعداد کاربرانی که به طور همزمان بتوانند یک وظیفه خاص را در server اجرا کرده و نتیجه مطلوب را بازیابی و مشاهده کنند.

برنامه نویسی دو لایه :

به این مدل ، **مدل سرویس دهنده / سرویس گیرنده client/ server** هم گفته می شود .

در این مدل پردازش اطلاعات بین سرویس دهنده و سرویس گیرنده تقسیم می شود. در واقع برنامه سرویس گیرنده رابط کاربری را نشان می دهد و اطلاعات لازم را از کاربر دریافت می کند. برنامه server یا سرویس دهنده نیز بر مبنای داده ها و اطلاعات ارسالی از client خدمات مناسبی ارائه می کند.



بررسی و مشکلات برنامه های دو لایه :

۱- مدیریت کد

از دو منظر قابل تامل است :

- از طرف سرویس دهنده
- از طرف سرویس گیرنده

از طرف سرویس دهنده :

تا زمانی که سرور عضو گروهی از سرور ها نیست توزیع تغییرات اعمال شده بر روی کد فقط در یک قسمت انجام می شود اما اگر سرور عضو یک گروه از سرورها باشد بایستی تغییرات اعمال شده بصورت دستی یا اتوماتیک به تمام سرورها اعمال گردد. این کار با نرم افزارهای مدیریتی کد مثل **MS Application Center 2000** انجام میگردد.

از طرف سرویس گیرنده :

مدیریت کد در سمت **client** مشکلتر از مدیریت کد در سمت **server** می باشد چون باید تمامی تغییرات اعمال شده روی تمام کامپیوترها یا سرویس گیرنده ها اعمال شود آن هم بطور همزمان .

از طرف دیگر هم باید تغییرات سمت **server** روی همه برنامه های سرویس گیرنده بروزرسانی شوند و اگر یکی یا تعدادی از یرویس گیرنده ها بروز رسانی نگردند برنامه سرور بدون استفاده میگردد.

۲- کارایی

به خاطر اینکه تقاضای اطلاعات بین سرویس دهنده و گیرنده جا به جا میشود و باعث محدودیت در شبکه میگردد واز طرفی هم تعداد کاربران ممکن است زمانی زیاد شود و منجر به قرار گرفتن تقاضاهای دریافتی کاربران برای ارائه خدمات در **صف** شده و هنگامی که تقاضاهای در حال انتظار افزایش یابد شبکه از کار می افتاد .

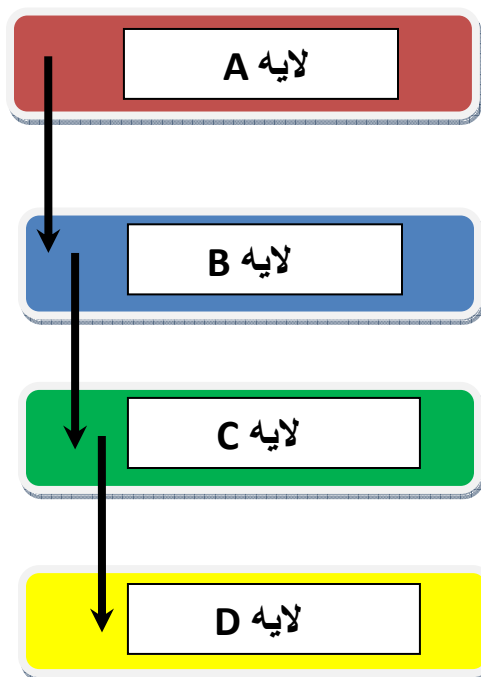
تنها راه حل این وضعیت قطع کردن تمام کاربران از شبکه و بازگرداندن تمامی تراکنشها به حالت قبلی ودر بعضی مواقع راه اندازی مجدد سرور می باشد که اینها باعث آزدگی و کاهش اعتماد کاربر میگردد.

برنامه نویسی سه لایه :

در واقع اجزای برنامه به چند لایه تقسیم می شوند . برنامه نویسی سه لایه نوعی برنامه نویسی چند لایه است.

برای برنامه های چند لایه شرط های زیر لازم است :

- لایه ها بایستی از هم مستقل باشند.
- لایه ها به ترتیب از بالا به پایین باهم در ارتباط باشند یعنی لایه بالایی فقط متدهای لایه پایین تر از خودش را فراخوانی کند و لایه پایین **نتواند** متدهای لایه بالای خودش را call نماید.



مثلا :

لایه B فقط متدهای لایه C را call کند

لایه C فقط متدهای لایه D را call کند

توجه : لایه های پایین تر **نباید** متدهای لایه بالاتر را فراخوانی کنند. (جهت فلش ها)

به برنامه هایی به فرم فوق **معماری چند لایه** می گویند . هر لایه از تعدادی کلاس و تعدادی متد تشکیل شده است.

نکته :

اگر در هر لایه (مثل B) که با لایه های بالا و پایین خود در ارتباط است **پارامترهای ورودی و خروجی تغییری نکنند** می توانیم کدهای متدهای داخل لایه مورد نظر (B) را تغییر دهیم و این همان مسئله **مستقل بودن لایه ها** از هم را نشان میدهد.

مدل برنامه نویسی سه لایه همان مدل دو لایه است که به سه قسمت تقسیم شده است :

- لایه رابط کاربر (سرویس کاربران)

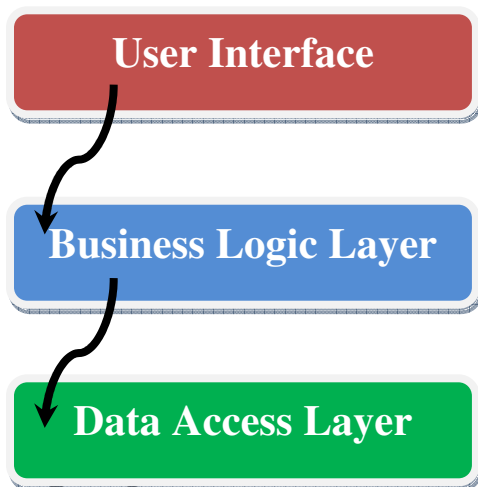
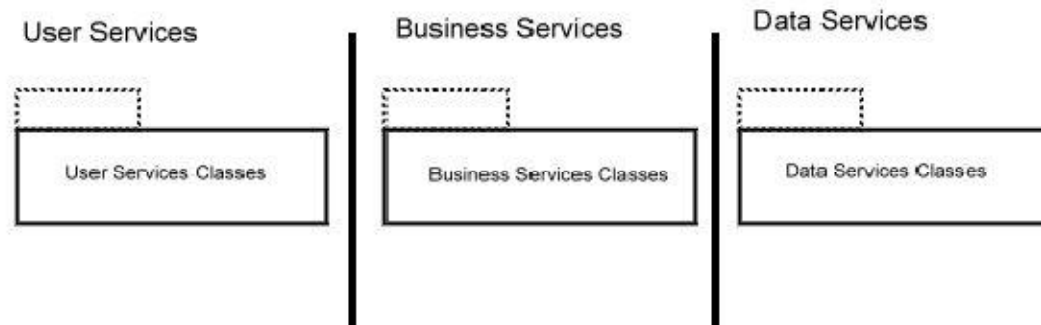
UI : User Interface

- لایه منطق تجاری (سرویس های تجاری)

BLL : Business Logic Layer

- لایه دسترسی به اطلاعات (سرویس های اطلاعاتی)

DAL : Data Access Layer



لایه رابط کاربر (سرویس کاربران) :

این لایه از فایل های اجرایی ویندوز یا صفحات وب مثل HTML یا صفحات ASP تشکیل شده است. در واقع لایه ای است که اطلاعات سرور را به کاربر نشان می دهد (به همین دلیل به آن **لایه ارائه دهنده** هم میگویند) و اطلاعات مورد نیاز سرور را هم از کاربر دریافت می کند.

در مدل سه لایه ، لایه رابط کاربر احتیاجی به دانستن ساختار بانک اطلاعاتی یا هر سرویس دیگری که توسط لایه سرویس های اطلاعاتی ارائه میشود **ندارد**.

لایه UI فقط از متدهای لایه BLL استفاده میکند .

لایه منطق تجاری (سرویس های تجاری) :

این لایه مسئول چگونگی دسترسی به اطلاعات می باشد. این مسئولیت شامل دریافت تقاضای اطلاعات مورد نیاز از طرف لایه رابط کاربر و فرستادن آن اطلاعات به لایه DAL و البته بازگرداندن پاسخ درخواست ها به کاربر است.

این لایه بایستی قواعد تجاری اعمال شده در برنامه را نگهداری و اجرا نماید .

لایه BLL می تواند تمام مواردی را که لایه رابط کاربر درخواست میکند انجام دهد و افزون بر آن کارهایی مانند انجام توابع محاسباتی و یا وظایف ویژه برنامه مثل محاسبات سود و زیان را نیز انجام دهد.

لایه BLL بطور فیزیکی در یکی از سرورهای موجود در شبکه قرار گرفته و اجرای تمام قواعد تجاری جدید یا تغییرات اعمال شده فقط بایستی در این لایه توزیع شوند که این امر موجب حذف نیاز به انتشار این قواعد تجاری بین سرویس گیرنده ها میشود .

در واقع لایه منطق تجاری لایه ای است که عملیات اصلی برنامه در آن انجام می گیرد مثلا در برنامه حسابداری انجام فرمول ها و محاسبات سود و زیان و بازده در این لایه صورت می گیرد. در اصل **مغز متفکر** برنامه همین لایه BLL است.

هدف از لایه BLL هم کمک به لایه UI برای ارائه خدمات بهتر به کاربر می باشد.

لایه دسترسی به اطلاعات (سرویس های اطلاعاتی) :

لایه DAL امکان دسترسی به داده ها و اطلاعات را برای لایه BLL فراهم می کند تا BLL هم آنها را برای لایه UI جهت ارائه خدمات به کاربران فراهم نماید.

سیستم مدیریت بانک اطلاعاتی (DBMS) و رابط ADO.NET که روشی جهت دریافت اطلاعات ارائه می دهد در این لایه قرار دارند.

در واقع لایه DAL ابزاری است که به ما کمک می کند تا به اطلاعات جانبی دسترسی داشته باشیم . اطلاعاتی که در فایل یا در پایگاه داده ذخیره شده اند. و هدف این لایه کمک به لایه BLL برای ذخیره و بازیابی اطلاعاتی که آن لایه لازم دارد می باشد.

مزایای برنامه نویسی سه لایه :

- ۱- به دلیل اینکه لایه ها بطور منطقی از یکدیگر مجزا و مستقل هستند هر کدام از آنها می توانند به طور مجزا کامپایل شده و یا مجدداً پیکربندی شوند بدون اینکه بر لایه های دیگر اثری بگذارند. و این قدرت **مدیریت کد** را می رساند .
- ۲- چون در مدل سه لایه اتصالات بانک اطلاعاتی می تواند بدون آنکه قطع شود از کاربری که به آن نیازی ندارد گرفته شده و برای سایر کاربران در نظر گرفته شود. با این کار **تعداد کاربرانی** که توسط لایه میانی (BLL) می توانند با بانک اطلاعاتی در تماس باشند را افزایش می دهد.
- ۳- چون در مدل سه لایه پردازش اطلاعات از سمت سرویس گیرنده به لایه BLL انتقال می یابد باعث افزایش **کارایی شبکه** به خاطر ارتباط BLL با لایه DAL بجای ارتباط سرویس گیرنده با لایه دسترسی به اطلاعات افزایش می یابد .
- ۴- با وجود مزایای ۲ و ۳ در اصل **مقیاس پذیری** افزایش می یابد زیرا کاربران با تاخیر و مشکل قطع شدن اتصال روبرو نمی شوند.

تذکر :

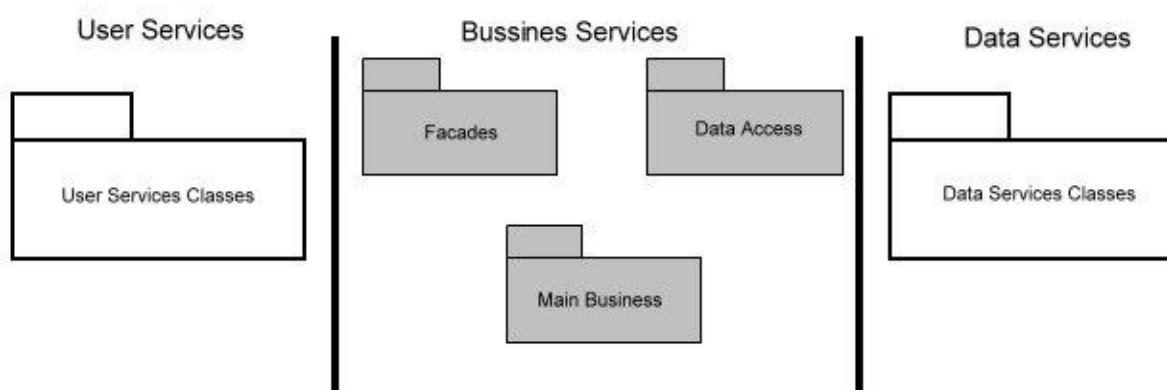
در برنامه های سه لایه **نباید** قواعد تجاری را در لایه کاربر اعمال کرد چون در این صورت کاربر براحتی می تواند از این قوانین براحتی عبور کند . در ضمن با این کار یعنی قرار دادن قواعد تجاری در سمت Client بایستی برای تمام کامپیوترهای این لایه آن قاعده را اعمال نمود .

پس بهتر است قواعد تجاری را در لایه DAL اجرا کنیم زیرا کاربران توانایی دسترسی به این لایه را ندارند .

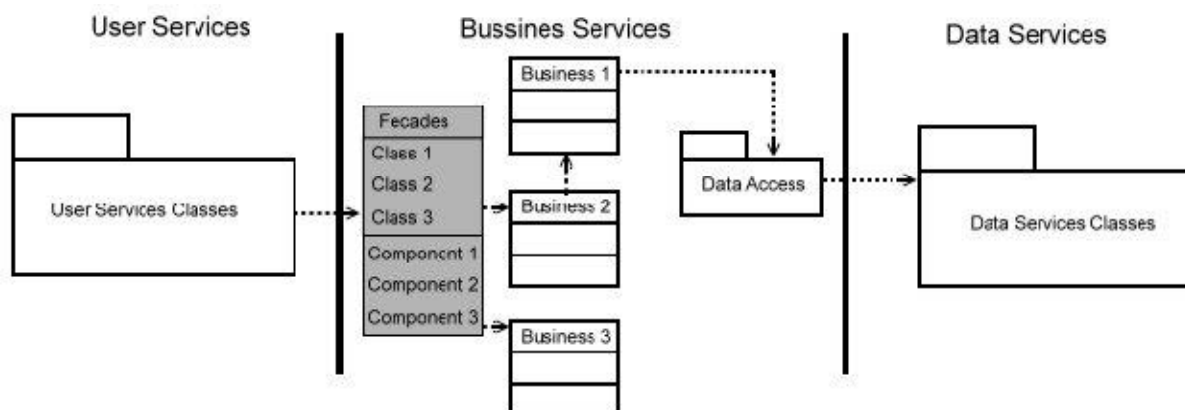
می توان قواعد تجاری را در لایه منطق تجاری هم قرار داد اما باید مطمئن شد **کاربران فقط از طریق متدهای** این لایه به لایه دسترسی به اطلاعات دسترسی داشته و نمی توانند از این لایه عبور کرده و به لایه بانک اطلاعاتی دست یابند.

برنامه نویسی چند لایه :

هر برنامه که بیش از سه لایه باشد جزء برنامه های چند لایه محسوب می شود . معمولترین آنها پنج لایه است که لایه BLL خود به سه لایه یا سه کلاس که عبارتند از : **کلاس خارجی ، کلاس اصلی تجاری و کلاس دسترسی اطلاعات** تقسیم میشود .



کلاس خارجی بعنوان یک بافر بین لایه رویس های کاربر و امکانات ارائه شده توسط لایه سرویسهای تجاری برنامه عمل می کند .



کلاس اصلی تجاری اجرای قواعد تجاری و تضمین کارایی منطق تجاری را انجام میدهد و اجزائی را فراهم می کند که کلاس خارجی بسیار از آنها استفاده میکنند .

کلاس دسترسی به اطلاعات که امکان استفاده از اطلاعات را برای کلاس اصلی تجاری فراهم کرده و تنها کلاسی (لایه ای) است که به لایه بانک اطلاعاتی دسترسی دارد .