

فصل ۱- تاریخچه و مفاهیم پایگاه داده‌ها

در این فصل، تاریخچه و سیر تکاملی پایگاه داده‌ها را بررسی کرده، مدل‌های پایگاه داده‌ها را مقایسه کرده و اجزاء و روند کار سیستم‌های پایگاه داده‌ها را مورد بحث قرار می‌دهیم.

۱-۱) مفاهیم مربوط به پایگاه داده‌ها

قبل از تعریف پایگاه داده‌ها لازم است تفاوت میان داده^۱ و اطلاع^۲ را درک کنیم. داده، واقعیتی خام است که نمی‌تواند هیچ تأثیری در تصمیم‌گیری‌های سیستم داشته باشد. به عنوان مثال، موارد ذیل سه داده در یک سیستم فروشگاه می‌باشند:

شماره فاکتور = ۱۰۰۱

تاریخ فاکتور = ۸۳/۴/۴

مبلغ کل فاکتور = ۶۰۰۰۰

کاملاً مشخص است که هیچ یک از موارد فوق نمی‌توانند مدیران فروشگاه را در اتخاذ یک تصمیم یاری کنند. فرض کنید در این فروشگاه دو بخش فروش وجود داشته باشد. با انجام برخی محاسبات روی مبالغ کل فاکتورهای هر بخش می‌توان میزان فروش کل هر یک از بخش‌ها را بدست آورده، از آن به عنوان معیاری برای مقایسه بازدهی دو بخش استفاده کرد. فرض کنید انجام این مقایسه، مدیران فروشگاه را به این نتیجه برساند که میزان فروش بخش ۱ نسبت به میزان فروش بخش ۲، ۵۰ درصد بیشتر بوده است. درک این واقعیت می‌تواند به مدیران فروشگاه کمک کند تا در قبال دو بخش تصمیم‌های مناسبی اتخاذ کنند. در واقع، این واقعیت که میزان فروش بخش ۱ نسبت به میزان فروش بخش ۲، ۵۰ درصد بیشتر بوده است یک اطلاع است. می‌توان چنین استنباط کرد که:

داده‌ها، واحدهای سازنده اطلاعات هستند.

و یا:

اطلاع، از پردازش داده‌ها بدست می‌آید.

اطلاعات، مبنای تصمیم‌گیری در سازمانها هستند. از طرف دیگر، اطلاعات مفید و جامع از داده‌های خوب بدست می‌آیند. به همین دلیل، برای تولید، ذخیره و بازیابی صحیح داده‌ها مکانیزم موثری مورد نیاز است. چنین مکانیزمی مدیریت داده‌ها^۳ نامیده می‌شود. آنچه مسلم است در دنیای کنونی برای مدیریت موثر داده‌ها باید از پایگاه داده‌های کامپیوتری استفاده شود. پایگاه داده‌ها، یک ساختار کامپیوتری یکپارچه^۴ شامل داده‌های مورد نیاز کاربران نهایی^۵ و فرا داده‌ها^۱ می‌باشد که

^۱ - Data

^۲ - Information

^۳ - Data Management

^۴ - Integrated

^۵ - End Users

توسط کلیه کاربران يك محیط عملیاتی (مثل دانشگاه، فروشگاه و...) به صورت اشتراکی مورد استفاده قرار می‌گیرد. برای درك تفاوت میان داده‌ها و فرا داده‌ها، محیط يك دانشگاه را در نظر بگیرید. در این سیستم داده‌هایی برای کاربران نهایی اهمیت دارند که یکی از موجودیتهای محیط عملیاتی (مانند دانشجو، استاد، وام و...) و یا یکی از روابط محیط عملیاتی (مانند ثبت نام يك دانشجو در يك درس، اخذ وام توسط دانشجو و...) را تشریح می‌کنند. علاوه بر این نوع داده‌ها، دسته دیگری از داده‌ها وجود دارند که مربوط به هیچ موجودیت یا رابطه عملیاتی نیستند. این دسته از داده‌ها که خود ساختار داده‌های اصلی را توصیف می‌کنند و یا برای سهولت و افزایش کارایی کار با داده‌های اصلی مورد استفاده قرار می‌گیرند، فراداده نامیده می‌شوند. مثلاً يك فراداده ممکن است ساختار جدول دانشجو را تشریح کند یعنی نام، نوع و طول فیلدهای جدول دانشجو را مشخص کند.

۲-۱) تاریخچه پایگاه داده‌ها

۱-۲-۱) ریشه‌های تاریخی پایگاه داده‌ها: سیستمهای فایل

قبل از اختراع کامپیوتر، شرکتها و سازمانها از سیستمهای دستی برای نگهداری اطلاعات خود استفاده می‌کردند. در این سیستمها هر دسته از اطلاعات در پوشه مجزایی نگهداری می‌شد. مثلاً در يك دانشگاه، اطلاعات مربوط به دانشجویان در يك پوشه، اطلاعات مربوط به اساتید در يك پوشه، اطلاعات مربوط به ثبت نام در يك پوشه و... طبقه بندی می‌شد. با رشد روزافزون حجم اطلاعات در سازمانها و شرکتها و همچنین نیاز به تهیه گزارشات دقیق و متنوع که فراتر از حد توان فکری و محاسباتی يك اپراتور و خارج از وقت و حوصله مدیران بود، چنین به نظر میرسید که سیستمهای دستی پاسخگوی نیازهای اطلاعاتی نیستند. خوشبختانه با اختراع کامپیوتر و ترویج استفاده همه گیر از آن، راه حلی برای این مسئله پیدا شد و سیستمهای دستی با سیستمهای فایل جایگزین شدند. به این ترتیب، در هر سازمان، اطلاعات هر پوشه در يك فایل کامپیوتری ذخیره شد و برنامه نویسان، با استفاده از زبانهای نسل سوم مانند کوبول و بیسیک برنامه‌هایی برای دستکاری اطلاعات موجود در فایلها و همچنین تهیه گزارشات براساس این اطلاعات نوشتند. مهمترین مشخصه و در واقع عیب سیستمهای فایل، عدم یکپارچگی آنها بود. در این سیستمها هر قسمت از سازمان سیستم فایل مجزایی داشت. به عنوان مثال، در يك دانشگاه قسمت آموزش از فایلها و برنامه‌های مجزا و قسمت امور مالی از فایلها و برنامه‌های مجزا استفاده می‌کرد. اگرچه ممکن بود بسیاری از فایلها مورد استفاده میان دو قسمت مشترک باشد، این دو قسمت از هیچ فایلی به صورت مشترک

استفاده نمی‌کردند مثلاً هر يك از آنها يك نسخهٔ مجزا از فایل دانشجویان را در اختیار داشتند. (در واقع، دو کپی از فایل دانشجویان مورد استفاده قرار می‌گرفت یکی در قسمت آموزش و دیگری در قسمت امور مالی.)

۲-۲-۱) مفاهیم موجود در سیستمهای فایل

- **داده^۷:** عبارت است از حقیقتی خام که هیچ مفهوم خاصی ندارد مثلاً وقتی از عدد ۵ صحبت می‌کنیم، مشخص نمی‌شود این ۵ تعداد فرزندان يك کارمند است یا وزن يك جسم یا سنوات تدریس يك استاد یا ...

تذکر: مفهوم داده در سیستمهای فایل با مفهوم داده در حالت کلی تفاوت دارد.

- **فیلد^۸:** گروهی از کاراکترهاست که خصوصیتی از يك موجودیت را توصیف می‌کنند مثل **age** و **name** و **weight**. هنگامیکه داده‌ای در داخل يك فیلد ریخته می‌شود، مفهوم و معنا پیدا می‌کند مثلاً اگر عدد ۵ را داخل فیلد **weight** بریزیم، وزن يك موجودیت را مشخص می‌کند.

- **رکورد^۹:** مجموعه‌ای از فیلدهاست که مربوط به يك موجودیت خاص هستند. مثلاً رکورد يك دانشجوی می‌تواند به شکل ذیل باشد:

| Stno | Name | Course |
|------|---------|--------|
| ۷۸۰۱ | آرش راد | ریاضی |

- **فایل^{۱۰}:** مجموعه‌ای از رکوردهاست که مربوط به يك نوع موجودیت هستند. مثلاً فایل دانشجویان يك دانشگاه می‌تواند به شکل ذیل باشد:

| Stno | Name | Course |
|------|------------|--------|
| ۷۸۰۱ | علی راد | هنر |
| ۷۸۰۹ | علی تقوی | ریاضی |
| ۷۹۰۱ | مینا رسولی | ریاضی |
| ۷۹۰۵ | آرش رضوی | هنر |

- **سیستم فایل^{۱۱}:** مجموعه‌ای از فایلها و برنامه‌های لازم برای کار با آنهاست. مثلاً سیستم فایل يك دانشگاه شامل:

- فایل دانشجویان و برنامه‌های مدیریت این فایل (برنامه‌هایی که برای درج، حذف و یا تغییر اطلاعات دانشجویان بکار می‌روند) و برنامه‌های مورد استفاده برای تهیه گزارشات براساس اطلاعات این فایل (تهیه لیستی از

⁷ - Data

⁸ - Field

⁹ - Record

¹⁰ - File

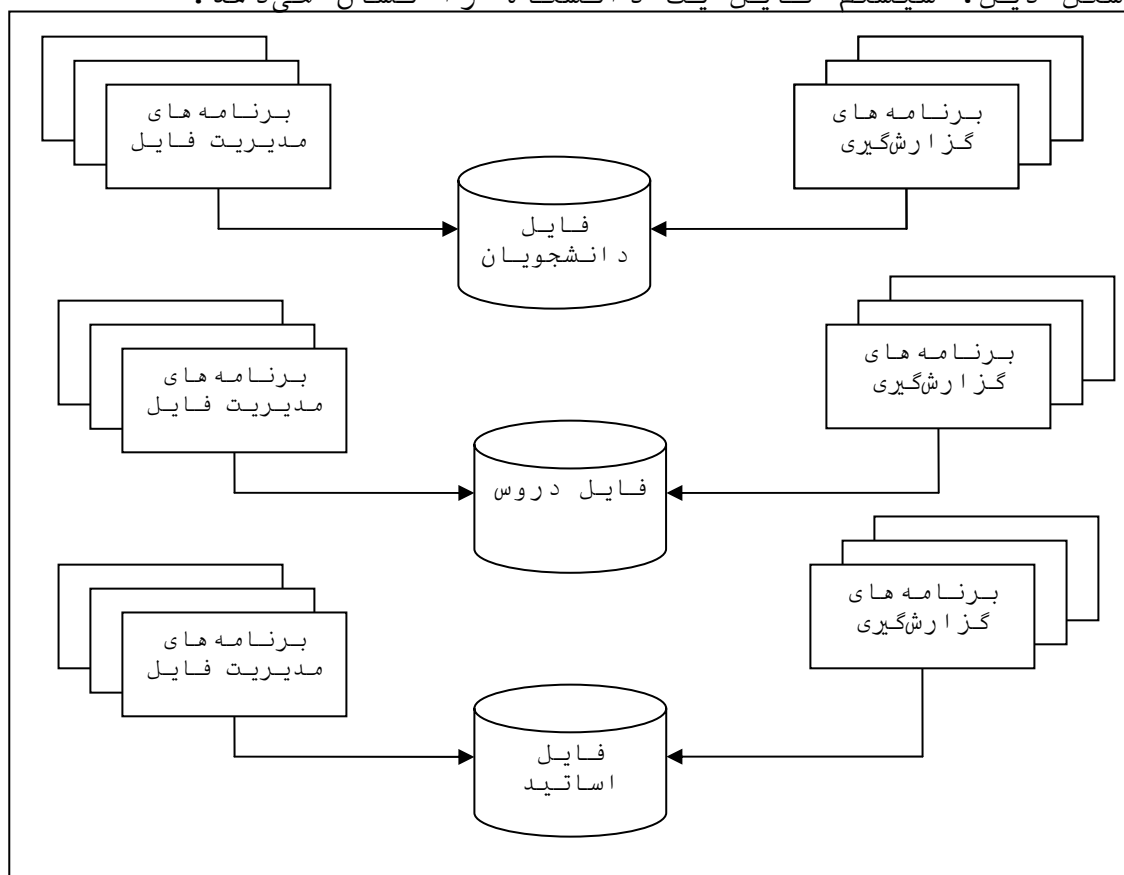
¹¹ - File System

دانشجویان ممتاز، تهیه لیستی از دانشجویان اخراجی و یا مشروط، تهیه لیستی از دانشجویان ورودی ۷۹ و ...)

- فایل دروس و برنامه‌های مربوط به مدیریت این فایل (درج، حذف و یا تغییر اطلاعات دروس) و برنامه‌های مورد استفاده برای تهیه گزارشات لازم براساس اطلاعات دروس (لیستی از دروس عملی، لیستی از دروس تخصصی کامپیوتر و ...)

- فایل اساتید و برنامه‌های مدیریت و گزارشگری مربوط به این فایل و ... می‌باشد.

شکل ذیل، سیستم فایل يك دانشگاه را نشان می‌دهد:



۳-۲-۱) معایب سیستمهای فایل

سیستمهای فایل، در کنار کلیه محاسنی که نسبت به سیستمهای دستی دارا بودند، معایب زیادی داشتند که مهمترین آنها عبارت بودند از:

۱- نیاز به برنامه نویسی زیاد و پیچیده

در سیستمهای فایل برای نوشتن ساده ترین برنامه‌ها که با فایلها در ارتباط بودند، لازم بود برنامه نویس از جزئیات نحوه ذخیره سازی فایلها کاملاً آگاه بوده و برای انجام کار موردنظر قطعه کدی طولانی و پیچیده بنویسد.

۲- وابستگی داده‌ای^{۱۲}

فرض کنید در فایل دانشجویان، برنامه نویس حداکثر ۲۰ کاراکتر را به فیلد `name` اختصاص داده باشد و پس از مدتی بخواهد طول این فیلد را به ۳۰ کاراکتر تغییر دهد. در این صورت، علاوه بر آنکه مجبور است اطلاعات موجود در فایل قدیمی‌را به یک فایل جدید با ساختار جدید منتقل کند (که البته خود این کار نیز ساده نمی‌باشد)، مجبور است کد کلیه قسمت‌هایی از برنامه که با فایل دانشجویان مرتبط می‌باشند را نیز تغییر دهد چرا که برنامه‌ها کاملاً به ساختار فیزیکی فایل وابسته اند. در واقع، وابستگی داده‌ای به "لزوم تغییر کد برنامه‌های مربوط به یک فایل پس از اعمال یک تغییر در مشخصات یکی از فیلدهای فایل (مثل تغییر نوع یک فیلد از `String` به `integer` و یا تغییر طول یک فیلد)" اطلاق می‌شود. در سیستم‌های فایل، وابستگی داده‌ای به شدت زیاد است.

۳- وابستگی ساختاری^{۱۳}

فرض کنید برنامه نویس بخواهد در فایل دانشجویان که مدتها مورد استفاده بوده است، یک فیلد جدید برای ذخیره شماره شناسنامه اضافه کند و یا یکی از فیلدهای موجود در آن را حذف کند. در این صورت، علاوه بر مشکلات مربوط به انتقال اطلاعات از فایل قدیمی‌به فایلی با ساختار جدید، برنامه نویس مجبور خواهد بود کلیه برنامه‌هایی که به نحوی به فایل دانشجویان مربوط می‌شوند را نیز تغییر دهد. در واقع، وابستگی ساختاری به "لزوم تغییر کد برنامه‌های مربوط به یک فایل پس از اعمال هر تغییر در ساختار فایل (اضافه کردن یا حذف فیلد)" اطلاق می‌شود.

۴- افزونگی داده‌ها^{۱۴}

همان گونه که قبلاً نیز به این نکته اشاره شد، هر یک از بخش‌های سازمان‌هایی که از سیستم‌های فایل استفاده می‌کردند، سیستم فایل مجزایی داشتند و حتی در صورت مشترک بودن داده‌های مورد استفاده، بخش‌های مختلف سازمان از فایلها به صورت اشتراکی استفاده نمی‌کردند مثلاً قسمت آموزش یک دانشگاه از یک نسخه از فایل دانشجویان و قسمت امور مالی از نسخه دیگری از فایل دانشجویان که عیناً مشابه هم بودند استفاده می‌کردند. به عبارت دیگر، یک قلم داده در چندین جای مختلف ذخیره می‌شد. این افزونگی یا تکرار داده‌ها، سیستم فایل را با مشکلاتی مواجه می‌کرد:

¹² - Data Dependence

¹³ - Structural Dependence

¹⁴ - Data Redundancy

-هدر رفتن فضا و نیروی کار-

ذخیره يك قلم داده در چند جای مختلف علاوه بر آنکه فضای حافظه زیادی را به هدر می‌داد، باعث به هدر رفتن نیروی کاری نیز می‌شد. مثلاً برای وارد کردن و اصلاح مداوم اطلاعات مربوط به دانشجویان يك دانشگاه دو اپراتور مورد نیاز بود: يك اپراتور برای درج و اصلاح اطلاعات دانشجویان در قسمت آموزش و دیگری برای درج و اصلاح اطلاعات دانشجویان در قسمت امور مالی.

-ناسازگاری داده‌ها^{۱۵}

فرض کنید یکی از دانشجویان به نام علی راد، قسمت امور مالی را از تغییر شماره تلفن خود آگاه کند. در این صورت، اگر اپراتور امور مالی شماره تلفن وی را اصلاح کند و اپراتور قسمت آموزش این تغییر را اعمال نکند، برای فایل شماره تلفن این دانشجو در دو فایل مختلف دو مقدار متفاوت وجود خواهد داشت. (می‌توانید تصور کنید اگر اپراتور قسمت امور مالی و اپراتور قسمت آموزش به مدت یکماه با یکدیگر قهر باشند چه اتفاقی می‌افتد!!!) در واقع، ناسازگاری داده‌ها به "وجود دو یا چند مقدار متفاوت برای يك قلم داده‌ای" اطلاق می‌شود. وجود افزونگی ذاتاً عامل مهمی برای بروز ناسازگاری داده‌ها می‌باشد.

-بروز انواع ناهنجاری

افزونگی داده‌ها، باعث بروز سه نوع ناهنجاری می‌شود. برای درک انواع ناهنجاری، سیستم يك دانشگاه را در نظر بگیرید. قسمت امور مالی این دانشگاه، فایلی به نام Tutor دارد که حاوی اطلاعات اساتید دانشگاه است:

| Tutor | Tname | Ttel |
|-------|-----------|--------|
| ۱۰۰ | آرش راد | ۶۲۴۴۴۴ |
| ۱۰۱ | مینا رضوی | ۶۱۵۲۵۲ |
| ۱۰۲ | عسل رسولی | ۲۱۹۳۹۴ |

از طرف دیگر، قسمت آموزش از فایلی بنام Project برای درج اطلاعات پروژه‌های فارغ التحصیلی دانشجویان استفاده می‌کند و چون به فایل‌های موجود در قسمت امور مالی دسترسی ندارد، مجبور است اطلاعات استاد راهنمای پروژه را نیز در این فایل نگهداری کند.

| St # | Proje-title | Tutor | Tname | Tel |
|------|---|-------|-----------|--------|
| ۷۸۰۱ | طراحی و پیاده‌سازی يك سایت خرید و فروش اینترنتی | ۱۰۰ | آرش راد | ۶۲۴۴۴۴ |
| ۷۹۰۲ | طراحی و پیاده‌سازی يك سیستم ثبت نام دانشگاه | ۱۰۲ | عسل رسولی | ۲۱۹۳۹۴ |
| ۸۰۰۱ | بهینه سازی پرس و جوها در پایگاه داده‌ها | ۱۰۰ | آرش راد | ۶۲۴۴۴۴ |

سازماندهی فایلها به صورت فوق می‌تواند باعث بروز ناهنجاریهای ذیل شود:

***ناهنجاری اصلاح^{۱۶}:** فرض کنید آرش راد شماره تلفن خود را به ۶۲۷۷۷۷ تغییر دهد. در این صورت، برای جلوگیری از بروز ناسازگاری لازم است هر دو قسمت امور مالی و آموزش این تغییر را اعمال کنند. بدیهی است قسمت آموزش برای اعمال این تغییر مجبور است کلیه رکورد های مربوط به پروژه های اخذ شده با آقای آرش راد را تغییر دهد. به عبارت دیگر، اگر آقای آرش راد تاکنون راهنمایی ۱۰۰ پروژه را برعهده گرفته باشد، شماره تلفن وی در هر یک از این ۱۰۰ رکورد باید تغییر کند چرا که در غیر این صورت شماره تلفن وی در رکوردهای مختلف دارای مقادیر متفاوت خواهند بود. این نوع اصلاح به اصلاح منتشر شونده^{۱۷} معروف است.

***ناهنجاری درج^{۱۸}:** برای درج هر پروژه جدید در فایل Project لازم است اطلاعات استاد راهنمای مربوطه نیز درج شود حتی اگر قبلاً اطلاعات وی در پروژه های دیگر درج شده باشد. این مسئله علاوه بر هدر دادن فضا و نیروی کاری، خطر بروز ناسازگاری را نیز افزایش می‌دهد چرا که در ورود دوباره اطلاعات استاد توسط اپراتور امکان خطا وجود دارد.

***ناهنجاری حذف^{۱۹}:** فرض کنید قسمت امور مالی اطلاعات مربوط به آقای آرش راد را از فایل Tutor حذف کند. در این صورت، در فایل Project برخی از دانشجویان استاد راهنمایی خواهند داشت که دیگر وجود ندارد. پس برای جلوگیری از ناسازگاری داده ها قسمت آموزش باید کلیه رکوردهای پروژه هایی را که با این استاد اخذ شده اند، حذف کرده و یا اطلاعات فیلدهای مربوط به استاد را در این رکوردها خالی کرده و یا با اطلاعات استاد دیگری جایگزین کند. از طرف دیگر، فرض کنید دانشجوی ۷۹۰۲ تنها دانشجویی باشد که با غسل رسولي پروژه گرفته است. در این صورت با حذف این دانشجو، قسمت آموزش، اطلاعات مربوط به غسل رسولي را نیز از دست می‌دهد. در واقع، با حذف يك قلم اطلاعاتی، به طور ناخواسته اطلاعات دیگری نیز حذف می‌شود.

¹⁶ - Modification Anomaly

¹⁷ - Propagating Modification

¹⁸ - Insertion Anomaly

¹⁹ - Deletion Anomaly

۵- عدم وجود امکانات لازم برای تأمین امنیت داده‌ها^{۲۰}
سیستم‌های فایل فاقد امکانات لازم برای تعیین حدود اختیارات هر کاربر بودند. به عبارت دیگر، در این سیستمها هر کاربر با هر سمت و اختیاراتی می‌توانست به کلیه داده‌های ذخیره شده در فایلها دسترسی داشته باشد. به راحتی می‌توانید خطرات ناامنی در دسترسی به داده‌ها را درک کنید!