انوع داده : ساخت یافته و غیر ساخت یافته ( مانند تکست که داخل آن اطلاعات است ولی قابلیت دسترسی آسان ندارد)

برای داده میبایستی دو امکان ذخیره و بازیابی موجود باشد.

علل نیاز به دیتا بیس : 1. انقلاب اینترنتی 1990 2. افزایش استفاده از اینترنت به واسطه تلفن همراه 3. افزایش گوناگونی سرویس های آنلاین

DBMS مسئولیت های زیر را برعهده میگیرد: جمع آوری اطلاعات مربوط به هم/ مجموعه از برنامه هایی که دسترسی به دیتا دارند/ محیطی برای دسترسی آسان و کارامد

مشکلات استفاده از فایل سیستم برای ذخیره دیتا :

* **دوباره نویسی و ناسازگاری دیتا**

اگر بخواهیم دو جدول دانشجو و استاد داشته باشیم چون نمیتوانیم رفرنس داشته باشیم مجبوریم همه اطلاعات را بنویسیم. و ناسازگاری خود نتیجه ناسازگاری است.

* **سختی دسترسی به دیتا**

برای پیدا کردن هر اطلاعاتی میبایست کل فایل را بگردیم. و برای هر Task میبایستی برنامه ای جدا بنویسیم که کل فایل سیستم را بررسی کند و خروجی را به ما بدهد.

* **جداسازی داده ها**

چون دیتا ها در فایل های مختلفی هستند و هر کدام هم فرمتی متفاوت دارند نوشتن برنامه برای خواندن همه آن ها و بررسی آن ها آن هم برای مقصود‌های مختلف

* **مشکلات یکپارچگی**

مثلا برای گذاشتن محدودیت روی دیتا ها دستمان بسته است و این محدودیت میبایست بر روی برنامه پیاده سازی شود

* **آپدیت ها میبایستی اتمیک باشد**

یک سری از آپدیت ها انجام شود و هنوز همه آن ها انجام نشده است ولی سیستم فیل شود مشکل ایجاد میشود.

* **هندل کردن دسترسی همزمان سخت است**

به جای دسترسی یکی یکی بهتر است همزمان دسترسی پیدا کنند (برای پرفورمنس). اما همین موضوع ممکن است موجب ناسازگاری شود.(ماجرای اضافه شدن پول به یک نفر)

* **مشکلات امنیتی**

سخت است سطح دسترسی را تامین کردن.

دیتابیس سیستمی از دیتاها و برنامه های مدیریت دیتا است که این اجازه را به کاربر میدهد که بتواند دیتا را بخواند و تغییر دهد ضمنا اهداف اصلی آن از این قرارند : 1. آماده سازی یک شهود انتزاعی از دیتا برای کاربران 2.جزئیات دیتا بیس را ازکاربر پنهان کند.

لایه ها abstraction : فیزیکی: که توصیف این که دیتا به چه صورت ذخیره میشود. منطقی: توصیف این که دیتا چه روابطی با هم دارند . و شهودی: برنامه ای که جزئیات را از کاربر پنهان میکند و مسائل امنیتی را تامین میکند ( مثلا پنهان کردن حقوق اساتید)

دیتا مدل ها :

مجموعه از ابزار ها برای توصیف 1. داده 2. ارتباط بین داده ها 3. مفهوم دیتا 4. محدودیت های دیتا

چند نوع دیتا مدل :

* relational model (دیتا ها روی جداول ذخیره میشوند و چیزی که میخونیم همینه 😊)
* Entity relationship data model (برای طراحی)
* Object- based data models
* Semi-structured data model

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_اسلاید دو \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

شما : طرح کلی هر تیبل (فیزیکی یعنی چه فضای فیزیکی ای میگیرد و... و منطقی هم منطق هر جدول را نشان میدهد) و instance هر نمونه از دیتا است (هر سطر)

این قابلیت که بتونیم ظرف فیزیکی دیتا ها را تغییر بدهیم بدون تغییر منطق دیتا ها را physical Data Independence میگویند. یعنی ساختار شما بایستی به گونه باشد که وابسطه به منطق دیتابیس باشد نه فیزیک دیتابیس.

DDL برای توصیف شمای دیتابیس به کار میرود/ کامپایلر با استفاده از DDL قالب هایی از جداول تهیه میکند و درdataDictionary ذخیر میکند/ دیتا دیکشنری متادیتا (اطلاعاتی راجع به دیتا) ذخیره میکند/ که یک نوع خاص جدول است که فقط از طریق خود دیتابیس قابل دسترسی است/ میتوان محدودیت هایی بر روی دیتاها ایجاد کرد / محدودیت های معمول با سربار کم بدین شرحند: محدودیت دامنه، یکپارچگی بدون شک (مثل Foreign key) ، assertion ها شرط هایی هستند که با تغییر هر جای دیتا بیس میبایستی بررسی شوند، و authorization هم یعنی اعمال سطح دسترسی به کاربران

DML زبانی برای دسترسی و تغییر دیتاها. انواع دسترسی: ساختن، خواندن ، تغییر ، حذف.

انواع DMl:

procedural: چه میخواهند و چگونه میبایست آن را به ما بدهد

Declarative : که فقط مشخص میکند که چه چیز هایی را میخواهد.

دو دسته بندی برای زبان ها : pure: در مورد کارایی کوئری هایی که میزنی و اثبات الگوریتم ها و ضمنا بهینه سازی و... جبر رابطه‌ای/ حساب رابطه ای سطرها/ حساب رابطه ای دامنه‌ای

commercial: در سیستم های تجاری استفاده میشوند مثل SQL

انواع استفاده از SQL : با استفاده از api / embedded SQL