

# به نام آنکه جان را فکرت آموخت

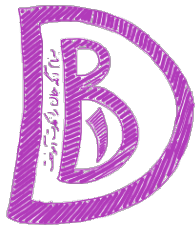


## بخش اول : مقدمه

مرتضی امینی

نیمسال دوم ۹۴-۹۵

(محتویات اسلایدها برگرفته از یادداشت‌های کلاسی استاد محمدتقی روحانی رانکوهی است.)

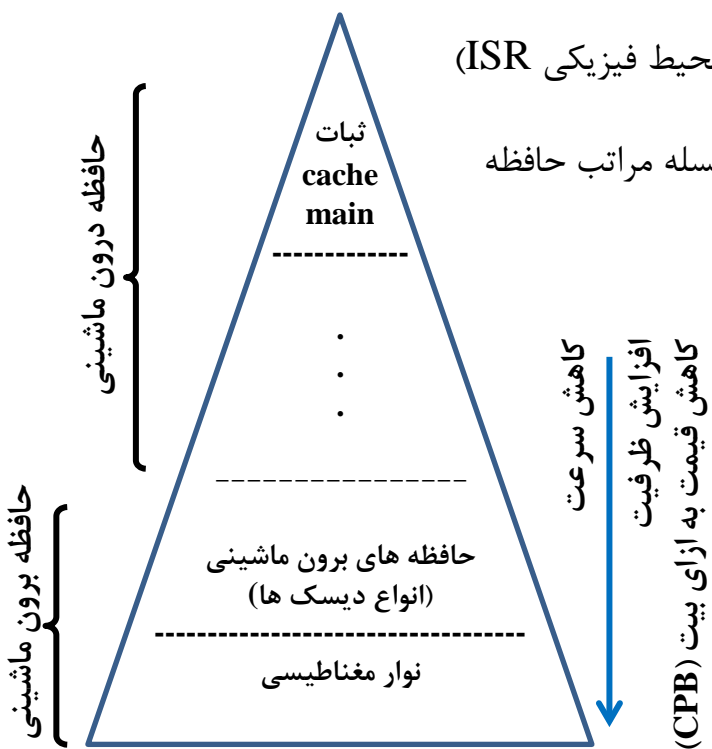


هر سیستم نرم‌افزاری از مجموعه‌ای از داده‌های ذخیره شده ممکن است استفاده کند.

در قالب تعدادی فایل (محیط فیزیکی ISR)  
در کجا؟ ← در یک سلسله مراتب حافظه

فرمت ثابت و از پیش تعیین شده دارد.  
well-formatted است،  
ساختمند (structured)  
نیم ساختمند (semi-structured)  
ناساختمند (un-structured)

به لحاظ ساختاری



آیا نیاز به تحمیل یک ساختار در اینها داریم؟ آیا واقعا داده ناساختمند داریم؟



انواع سیستم نرم‌افزاری:



- بنیادی یا پایه (سیستم‌های عامل)
- نیمه بنیادی (DBMS، DMS، کامپایلرها، اسمبلرها، و ...)
- کاربردی (برنامه‌های کاربردی)
- ابزاری: انواع toolها

Data Management System

Database Management System

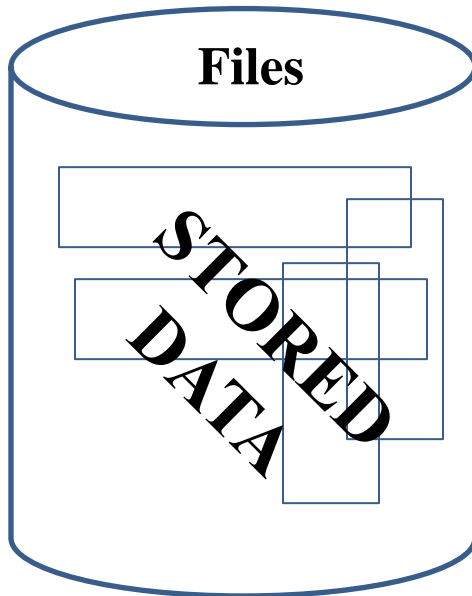


کنجکاوی: دلایل استفاده از این سلسله مراتب حافظه چیست؟

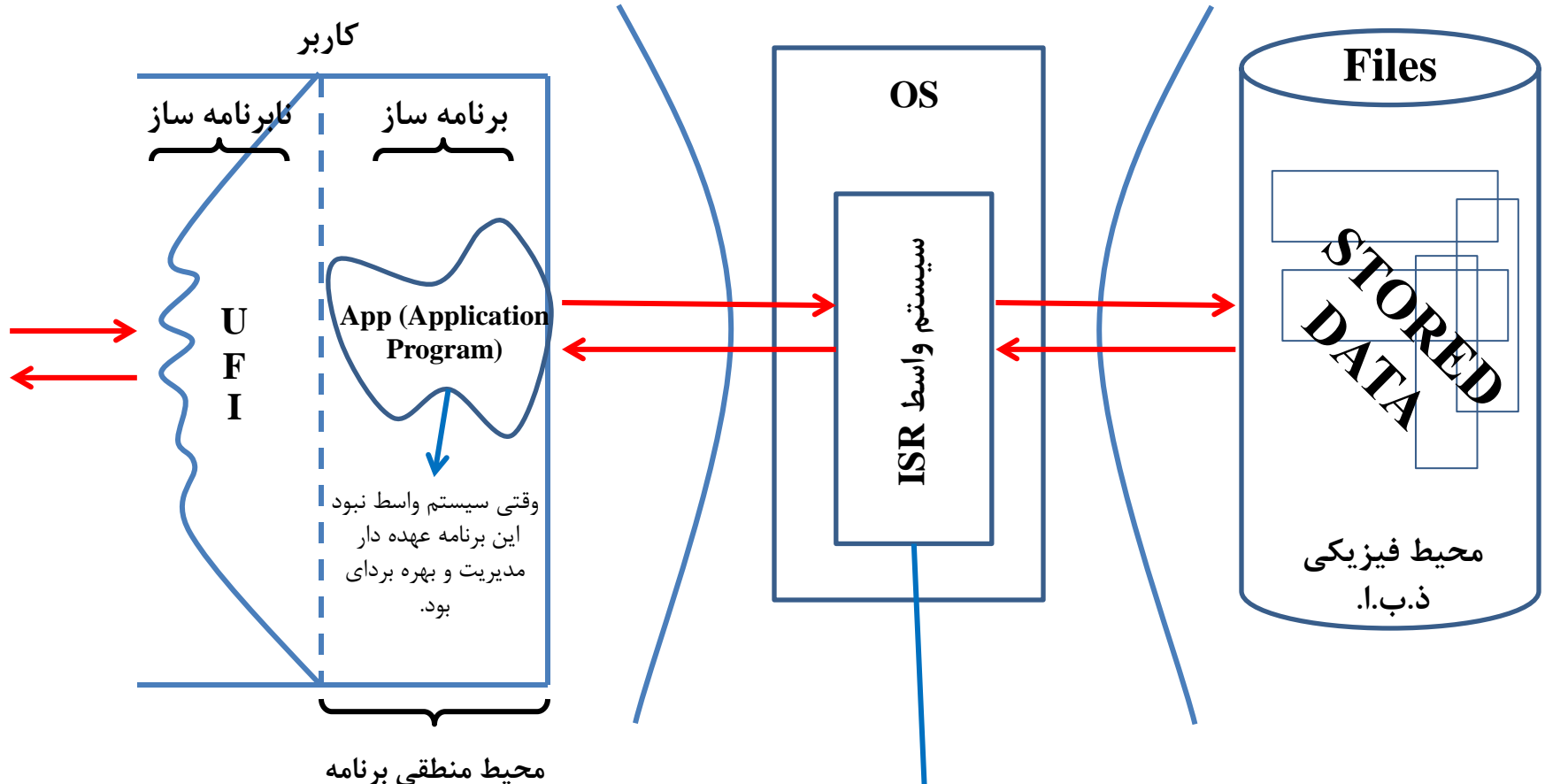
کنجکاوی: چه داده ای، برای چه مدتی، در کدامیک از مراتب سلسله مذکور قرار می گیرد؟

کنجکاوی: خصوصیات عمومی فایل ها چیست؟

□ محیط فیزیکی «ذ.ب.ا.» (ذخیره و بازیابی اطلاعات) یا (Information Storage and Retrieval) ISR



ISR: باید { ایجاد  
مدیریت  
بهره برداری } شود. ← نیاز به یک سیستم واسط ذ.ب.ا داریم.



این سیستم این امکان را می دهد تا کاربر داده های خود را { ذخیره  
بازیابی  
پردازش } کند.

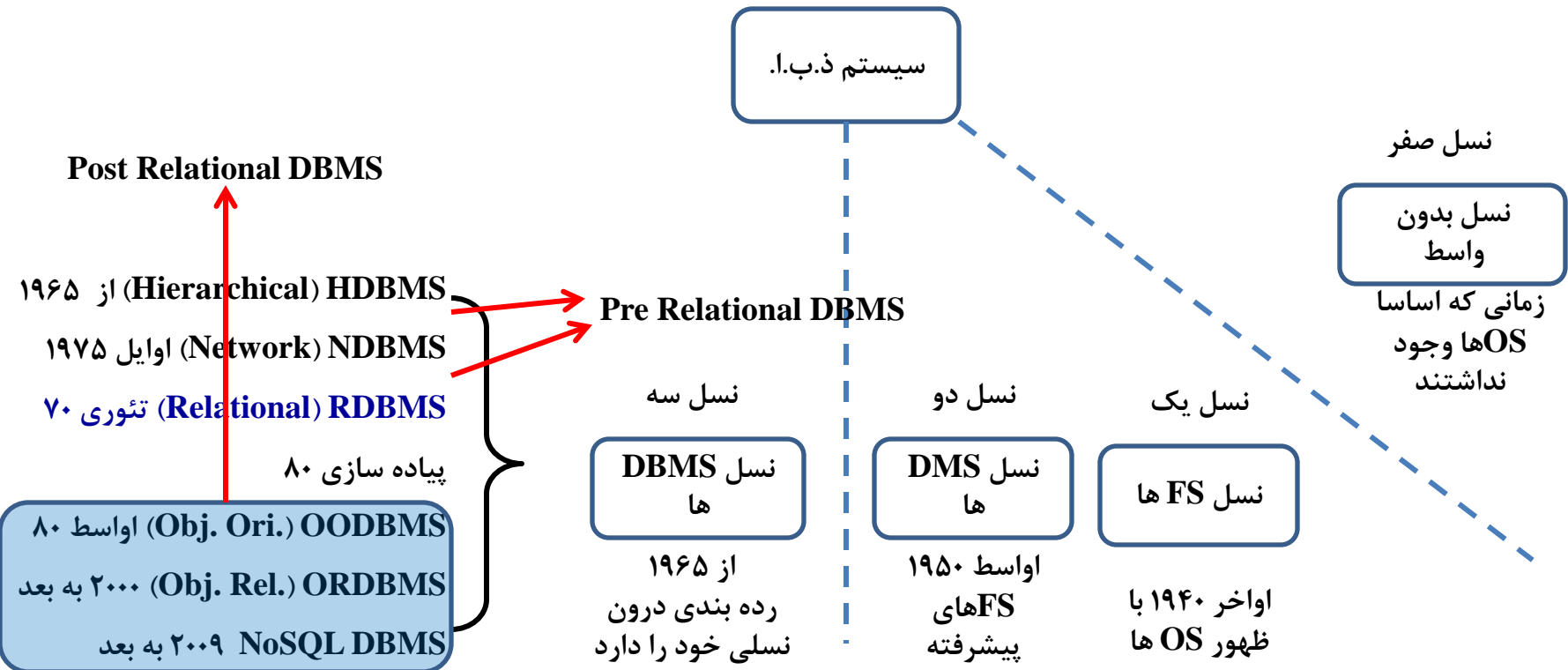


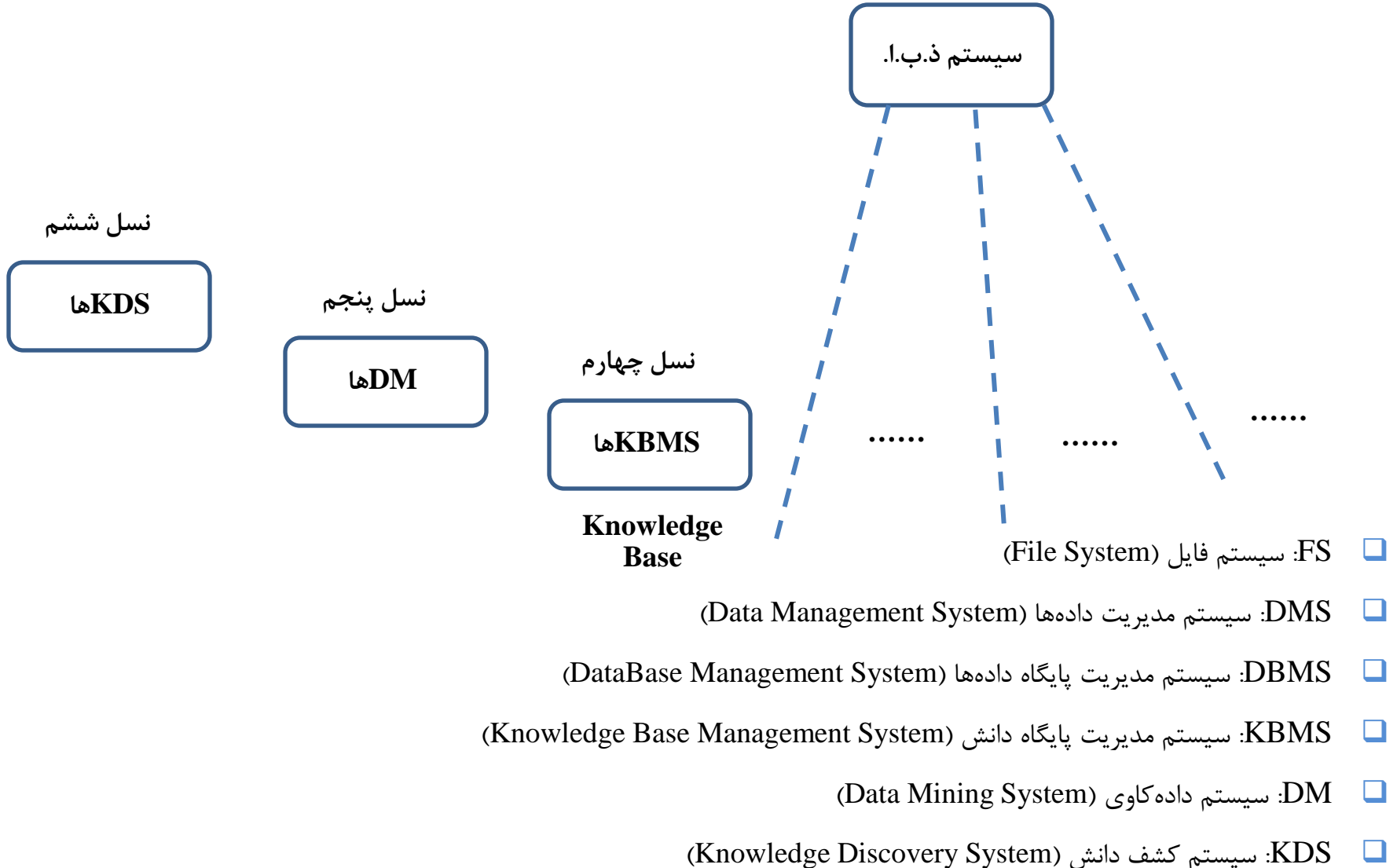
# سیر تحول سیستم ذ.ب.ا. (۱)

کنجکاوی: رده بندی از مفهوم کاربر ارایه کنید؟ به بیان دیگر گونه های دیگر کاربر کدامند؟

□ سیستم واسط "ISR" سیر تحول خاص خود را دارد :

□ ۶ نسل تکنولوژیک قابل بازیابی است (به طور کلی) [دیدگاه نرم‌افزاری]







- در این نسل بندی، نسل بعدی نسل قبلی را منسوخ نمی کند. نسل بعدی نسل قبلی را تکمیل می کند و از آن استفاده می کند.
- انواع نیازهای پردازشی، کنترلی، و عملیاتی سبب ایجاد نسل های سیستم «ذ.ب.ا.» شد.



### داده (Data) □

- تعریف اول ANSI: نمایش بوده‌ها، پدیده‌ها، مفاهیم یا شناخته‌ها به طرز صوری و مناسب برای برقراری ارتباط، تفسیر یا پردازش توسط انسان یا هر امکان خودکار
- تعریف دوم ANSI: هر نمایشی اعم از کاراکتری (نویسه‌ای) یا کمیت‌های قیاسی که معنایی به آن قابل انتساب باشد (توسط انسان یا یک مکانیسم خودکار)

### اطلاع (Information) □

- تعریف دقیق و جامعی از مفهوم اطلاع وجود ندارد.
- تعریف اول [LIPS92]: اطلاع، داده پردازش شده است.
- تعریف دوم [روحا ۷۸-الف]: معنایی که انسان به داده منتسب می‌کند، از طریق قراردادهای شناخته شده‌ای که در نمایش داده به کار می‌روند.
- برخی داده را همان مقدار واقعا ذخیره شده و اطلاع را معنای آن می‌دانند. بنابراین اطلاع دارای خاصیت اطلاع‌دهندگی و ارتباط‌دهندگی است، در حالیکه داده مجرد این خاصیت را ندارد.





### دانش (Knowledge) □

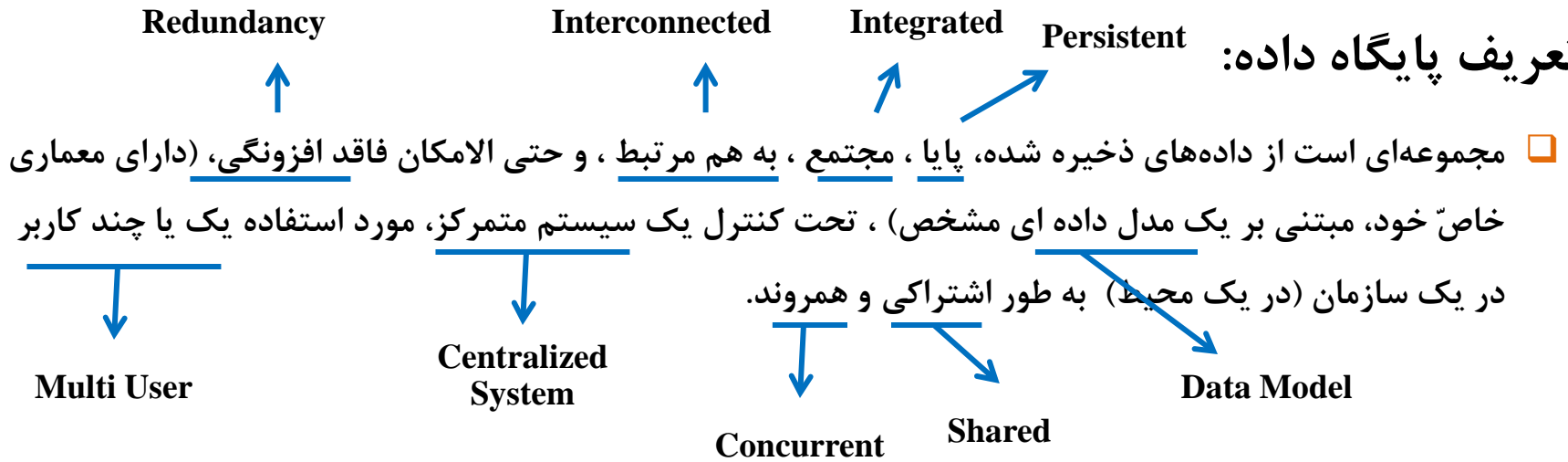
□ **تعریف [FROS87]:** دانش عبارت است از نمایش نمادین جنبه‌هایی از بخشی از جهان واقع (جهان موردنظر یا محیط مطرح)

▪ مثال: شنبه هوا بارانی است. حسن فرزند علی است.

□ **تعریف دوم [روحا ۹۱]:** دانش منطقی نوعی شناخت است که از یک مجموعه از اطلاعات بر اساس یک مجموعه از قواعد مشخص، معمولاً با روش استقراء حاصل می‌شود. حصول این شناخت می‌تواند توسط انسان یا یک سیستم خودکار انجام شود.



### تعریف پایگاه داده:



## □ مثال کاربردی

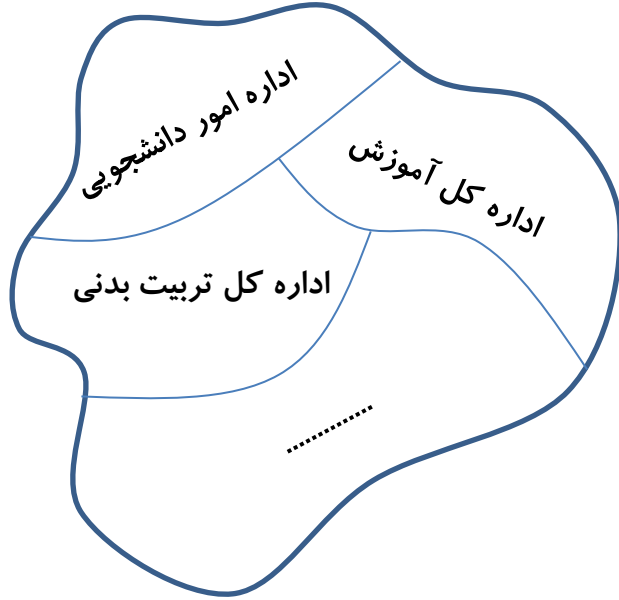
### □ محیط عملیاتی: دانشگاه



بخشی از جهان واقعی که قصد ایجاد سیستم برای آن را داریم.



Micro Real World (خرد جهان واقع)  
 Mini World  
 Universe of Discourse (جهان مطرح)



□ نکته: هر محیط از تعدادی زیر محیط تشکیل شده است.

□ در هر محیط مجموعه‌ای از **نوع موجودیت‌ها** وجود دارند که نیازهای

به داده‌هایی در مورد آنها نیاز دارند).

داده‌ای }  
 کاربران ناظر به آنهاست (یعنی }  
 پردازی



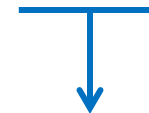
□ **نکته:** زیرمحویط های یک محیط معمولا با هم اشتراک دارند در نوع موجودیتها (Entity Type یا Object Type)

□ مثال: در محیط دانشگاه دانشجو، استاد، درس، کلاس، و ...

□ مثال: نوع موجودیت دانشجو در هر سه زیر محیط مطرح است.

□ **مسئله (خواسته):** ایجاد سیستم(های) کاربردی برای این زیر محیطها

□ برای این منظور در اساس دو مَشی-روش (approach) وجود دارد. }  
 مَشی فایلینگ [سنتی یا کلاسیک] یا ناپایگاهی  
 مَشی پایگاهی Database Approach



یعنی ممکن است مَشی های بینابینی نیز وجود داشته باشد.



کارهای لازم در مشی فایلینگ به طور خلاصه:

توجه: این کارها معمولاً برای هر زیرمحیط به طور جداگانه انجام می شود. ← تعدادی سیستم کاربردی جدا (نامجتمع) و بی ارتباط در یک محیط ...

- |   |                                  |   |
|---|----------------------------------|---|
| <ul style="list-style-type: none"> <li>۱- تشخیص نیاز های داده‌ای</li> <li>۲- تشخیص نیاز های پردازشی</li> <li>۳- مستندسازی نیازها</li> <li>۴- دریافت تایید سازمان</li> </ul> | <p>← Requirement Engineering</p> | <ul style="list-style-type: none"> <li>۱- مطالعه و شناخت محیط</li> <li>۲- انجام مهندسی نیازها</li> <li>-----</li> <li>۳- تعیین مشخصات سیستم کاربردی System Specification</li> <li>-----</li> <li>۴- [انتخاب پیکربندی سخت افزار و نرم افزار H/S]</li> <li>۵- [انتخاب یک FS و/یا DMS] سیستم واسط ISR</li> <li>-----</li> <li>۶- طراحی تعدادی فایل (طبق مشخصات سیستم)</li> </ul> |
|---|----------------------------------|---|



۱-۶- تعیین فرمت رکورد

۲-۶- تعیین ساختار فایل

ساختار فایل: ساختاری که براساس آن فقره داده ها (رکوردها) در سطح منطقی [و/یا فیزیکی] با یکدیگر مرتبطند. ساختار فایل یک امکان برای نمایش ارتباط بین فقره داده‌هاست (Data Items) خواه در سطح نمایش منطقی باشد یا فیزیکی.

کنجکاو: چند نوع ساختار فایل وجود دارد؟

۳-۶- نحوه دسترسی به رکوردها - استراتژی دسترسی

۴-۶- اندازه فایل ها

۵-۶- میزان گسترش چه میزان باشد

۶-۶- ارتباط با فایل های دیگر

۷-۶- عملیات مجاز در فایل ها + کاربران

□ کارهای لازم در مشی فایلینگ به طور خلاصه : (ادامه)

۷- طراحی واسط‌های کاربری (UFI)

۸- طراحی تعدادی برنامه کاربردی (Application Program) [ضمن تعیین تراکنش(ها)]

۹- تولید برنامه‌های { ایجاد  
کنترل  
پردازش } فایل‌ها

۱۰- ایجاد محیط فیزیکی «ذ.ب.ا.» به طور آزمایشی (برای داده‌های تست)

۱۱- ایجاد محیط فیزیکی «ذ.ب.ا.» با داده‌های واقعی اما حجم محدود و انجام تست مرحله دوم

۱۲- ایجاد محیط فیزیکی «ذ.ب.ا.» با داده‌های واقعی و حجم واقعی و انجام تست مرحله سوم]

۱۳- رفع اشکال‌ها در هر مرحله

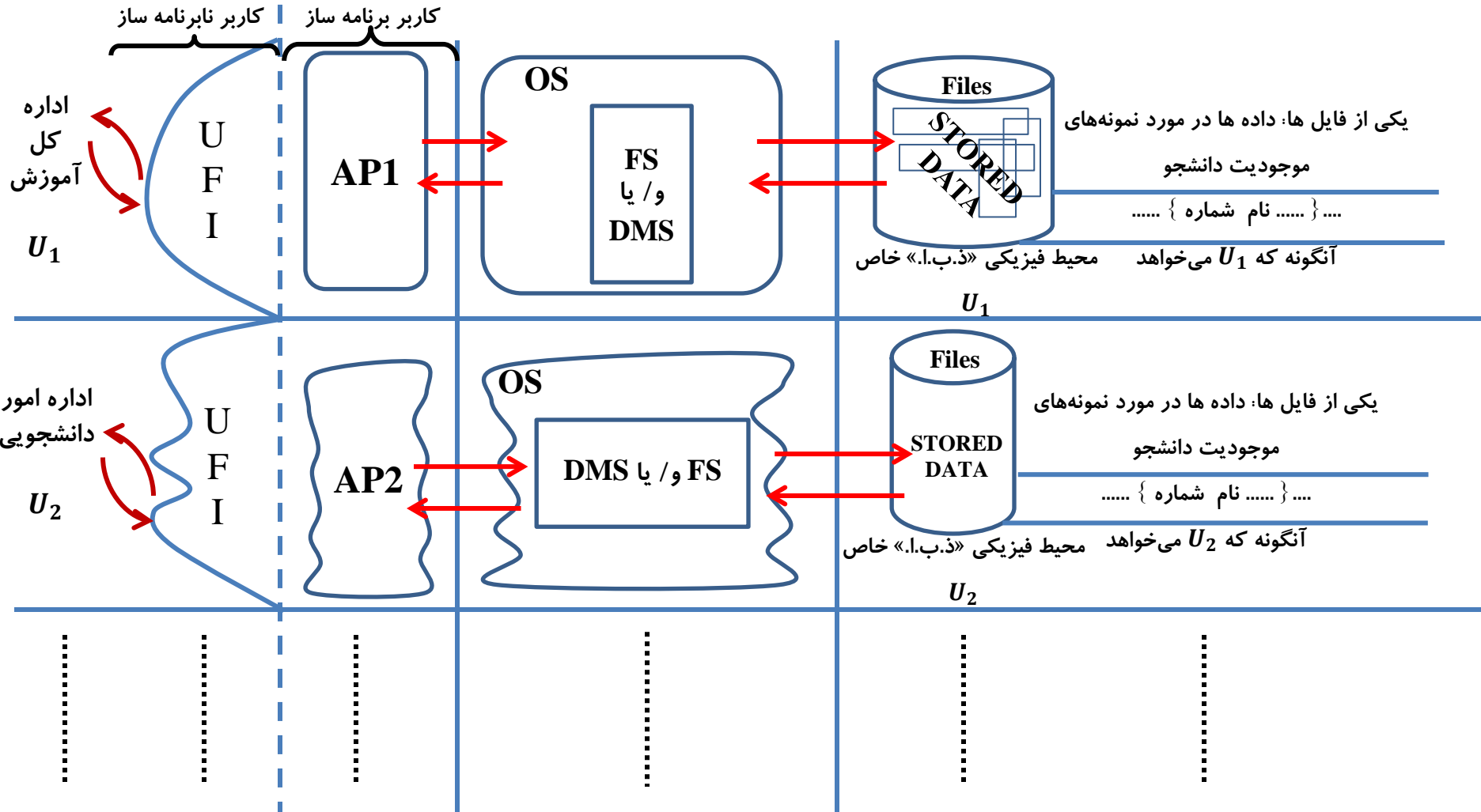
۱۴- ایجاد محیط فیزیکی واقعی با نصب، پیکربندی و ورود داده‌های اولیه (Data Entry)

۱۵- آغاز بهره‌برداری و نگهداری سیستم

۱۶- رفع معایب و بهینه‌سازی سیستم



## فصل اول - مقدمه







## برخی از معایب مشی فایلینگ:

وجود سیستم های نامجتمع در یک سازمان [محیط] و نامرتبط به هم

عدم وجود یک سیستم کنترل متمرکز روی کل داده های سازمان

وجود افزونگی زیاد

خطر بروز ناسازگاری داده ها (Data Inconsistency) ← **کنجکاوی**: جنبه های بروز ناسازگاری کدامند؟

عدم امکان اعمال ضوابط حفظ امنیت داده ها (Data Security)

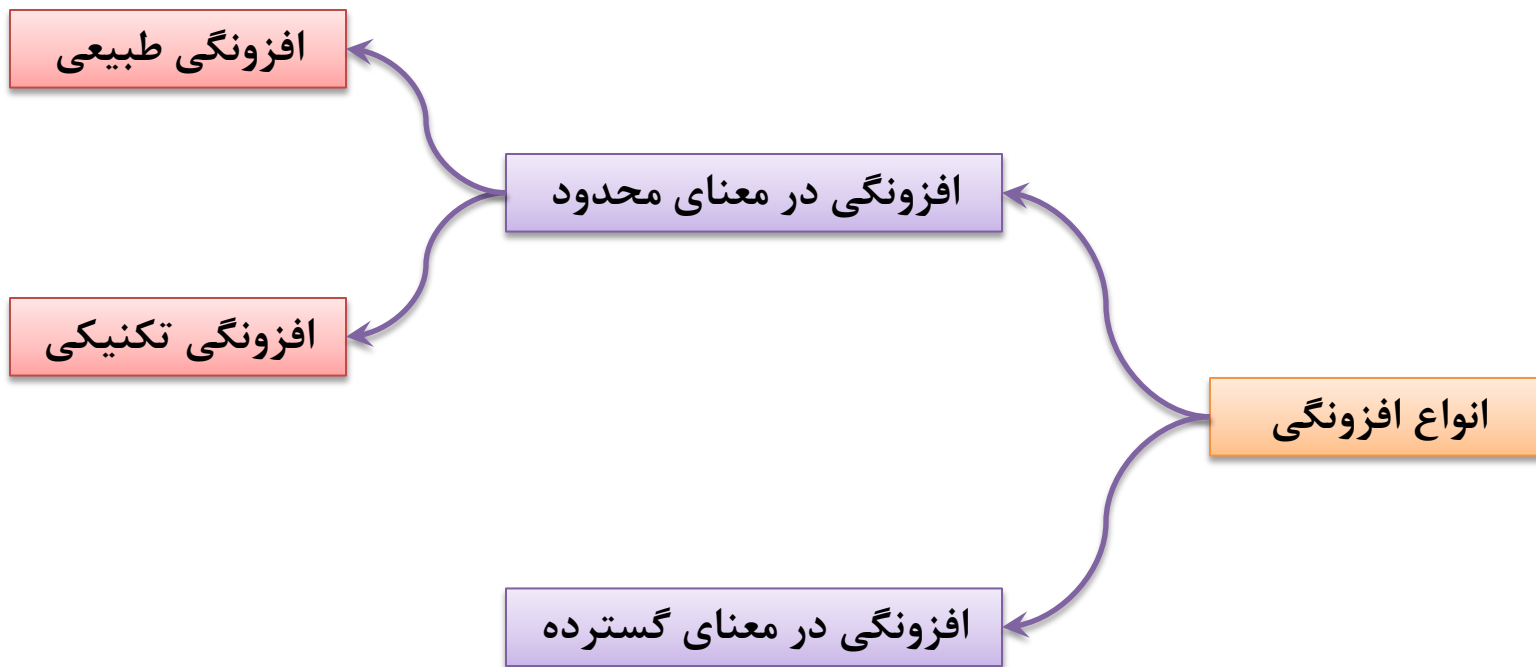
عدم امکان اشتراکی شدن داده ها (Data Sharing) [یا در حداقل و یا با دشواری]

مصرف نابهبینه سخت افزار (به ویژه سخت افزار ذخیره ساز)

وابسته بودن برنامه ها به جنبه های فایلینگ محیط ذخیره سازی، به گونه ای که اگر قرار باشد در فایلینگ

تغییراتی ایجاد شود، برنامه ها هم متناسبا باید تغییر یابد. (به طور مثال فرمت ساختار یا نحوه دسترسی

(Access Strategy) را تغییر دهیم)



□ افزونگی در معنای محدود (یعنی درون فایلی - intrafile redundancy - در مباحث فایلینگ)

□ عبارت است از تکرار ذخیره سازی مقادیر (value) یک صفت یا بیش از یک صفت در فایل داده‌ای یا فایل کمکی آن.

□ این نوع افزونگی گونه‌هایی دارد:

۱- **طبیعی**: ناشی از ماهیت داده‌های محیط (مثل صفت رشته دانشجو که برای دانشجویان مختلف می‌تواند یکسان و در نتیجه تکراری باشد)

▪ **کنجکاوی**: برای کاهش مصرف حافظه در حالت افزونگی طبیعی چه باید کرد؟

۲- **تکنیکی**: ناشی از استفاده از یک تکنیک معمولا برای افزایش سرعت (مثل نمایه سازی [شاخص بندی (Indexing)



افزونگی در معنای گسترده (یعنی برون‌فایلی - در مباحث پایگاه داده)

عبارت است از تکرار ذخیره‌سازی داده‌ها در مورد نمونه‌های یک یا بیش از یک نوع موجودیت از یک محیط.

این نوع افزونگی نه از نوع طبیعی و نه از نوع تکنیکی است بلکه ناشی از رهیافت انتخاب شده برای طراحی و تولید سیستم‌های کاربردی است.

به طور مثال تکرار اطلاعات دانشجویان در دو زیرسیستم اداره کل آموزش و زیرسیستم اداره امور دانشجویی.

**نکته:** افزونگی از نوع طبیعی و تکنیکی در پایگاه داده هم می‌تواند وجود داشته باشد.

دلایل بروز افزونگی در سیستم‌های ISR به ویژه سیستم‌های پایگاهی کدامند؟





تشکیل شده از تعدادی درایه (مدخل-entry)

نمایه متراکم dense

یک

هر مدخل اشاره دارد به { یا گروهی (به صورت) از رکورد ها

چند سطحی

نمایه نامتراکم Non-dense

مقدار | آدرس

تکیه گاه (Anchor point)

مقدار یک صفت (معمولا کلید)



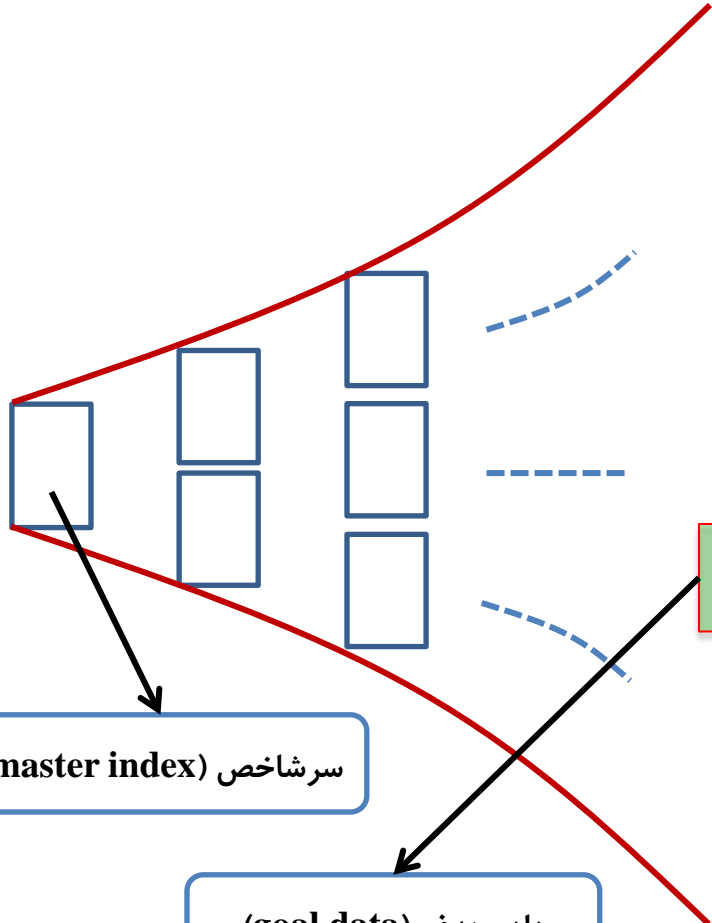
نمایه نامتراکم

نمایه متراکم

فایل نمایه سازی شده

(چندسطحی معمولاً با ساختار B-Tree یا B+-Tree)

(زمان جستجو بالاست)



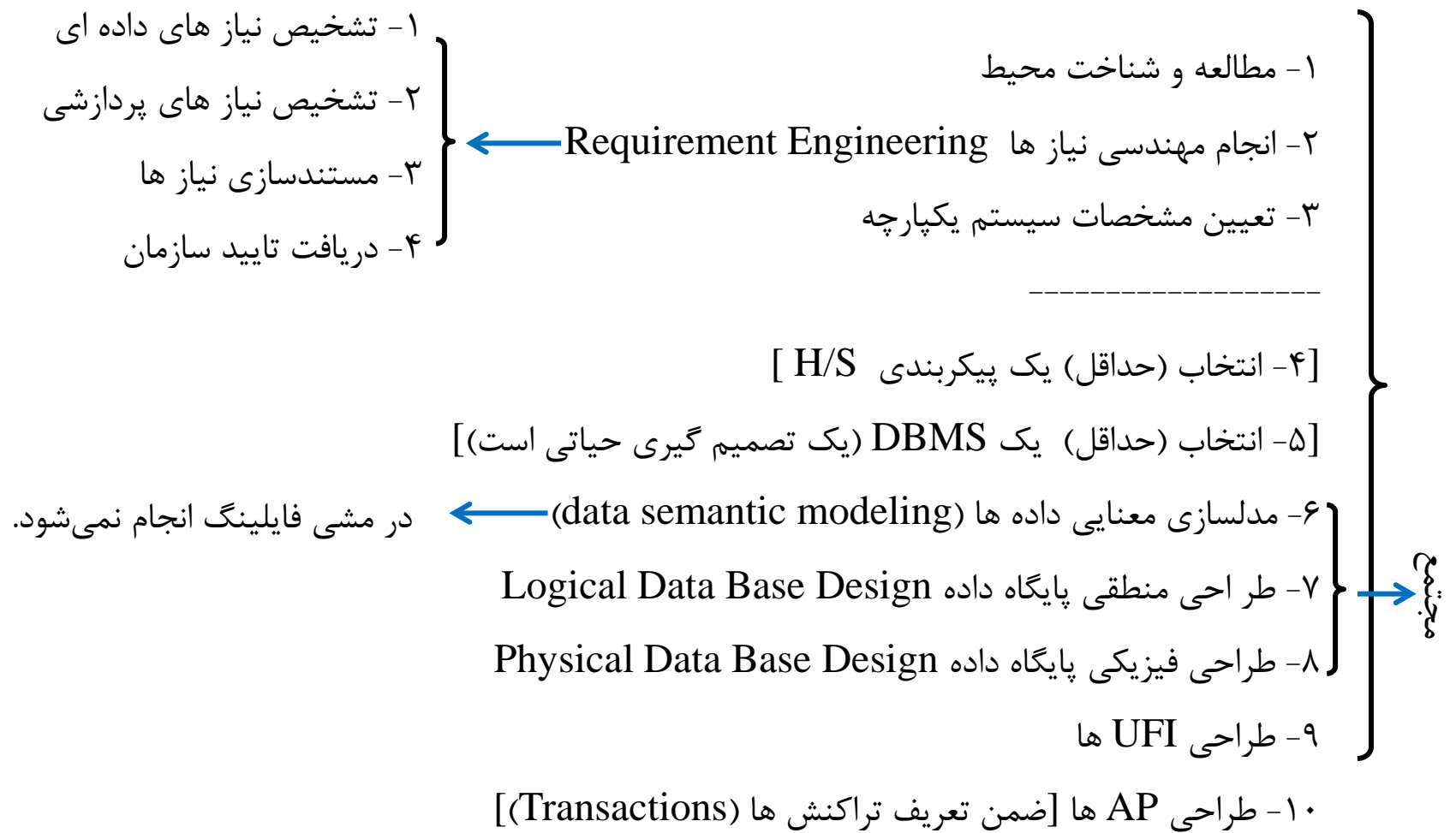
شماره	نام	رشته	...
۱۰۰	۱	نرم افزار	
۱۰۱	۲	نرم افزار	
۱۰۲	۳	سخت افزار	
۱۰۳	۴	نرم افزار	
۱۰۴	۵	سخت افزار	
۱۰۵	۶	نرم افزار	
⋮			
<i>k</i>		نرم افزار	
⋮			
۹۹۷	۸۹۸	سخت افزار	
۹۹۸	۸۹۹	سخت افزار	
۹۹۹	۹۰۰	نرم افزار	

سر شاخص (master index)

داده هدف (goal data)



□ کارهای لازم در انجام یک «پروژه پایگاهی»: (فعلا نه در جزئیات)



به منظور ایجاد یک سیستم یکپارچه



ادامه:...

مزایا و معایب جداسازی این دو دسته برنامه



تعریف و کنترل و عملیات در داده‌ها چیست؟

۱- از دیدگاه عملیات در داده‌ها

۲- از دیدگاه زبان‌های برنامه‌سازی

۱۱- تولید برنامه‌های تعریف (ایجاد) و کنترل DB

۱۲- تولید برنامه‌های عملیات در داده‌ها (پردازش داده‌ها)

۱۳- ایجاد محیط فیزیکی «ذ.ب.ا.» با داده‌های تستی و رفع اشکال‌ها (تست مرحله اول)

۱۴- ایجاد محیط فیزیکی «ذ.ب.ا.» با داده‌های واقعی اما حجم محدود و انجام تست مرحله دوم

۱۵- ایجاد محیط فیزیکی «ذ.ب.ا.» با داده‌های واقعی و حجم واقعی و انجام تست مرحله سوم

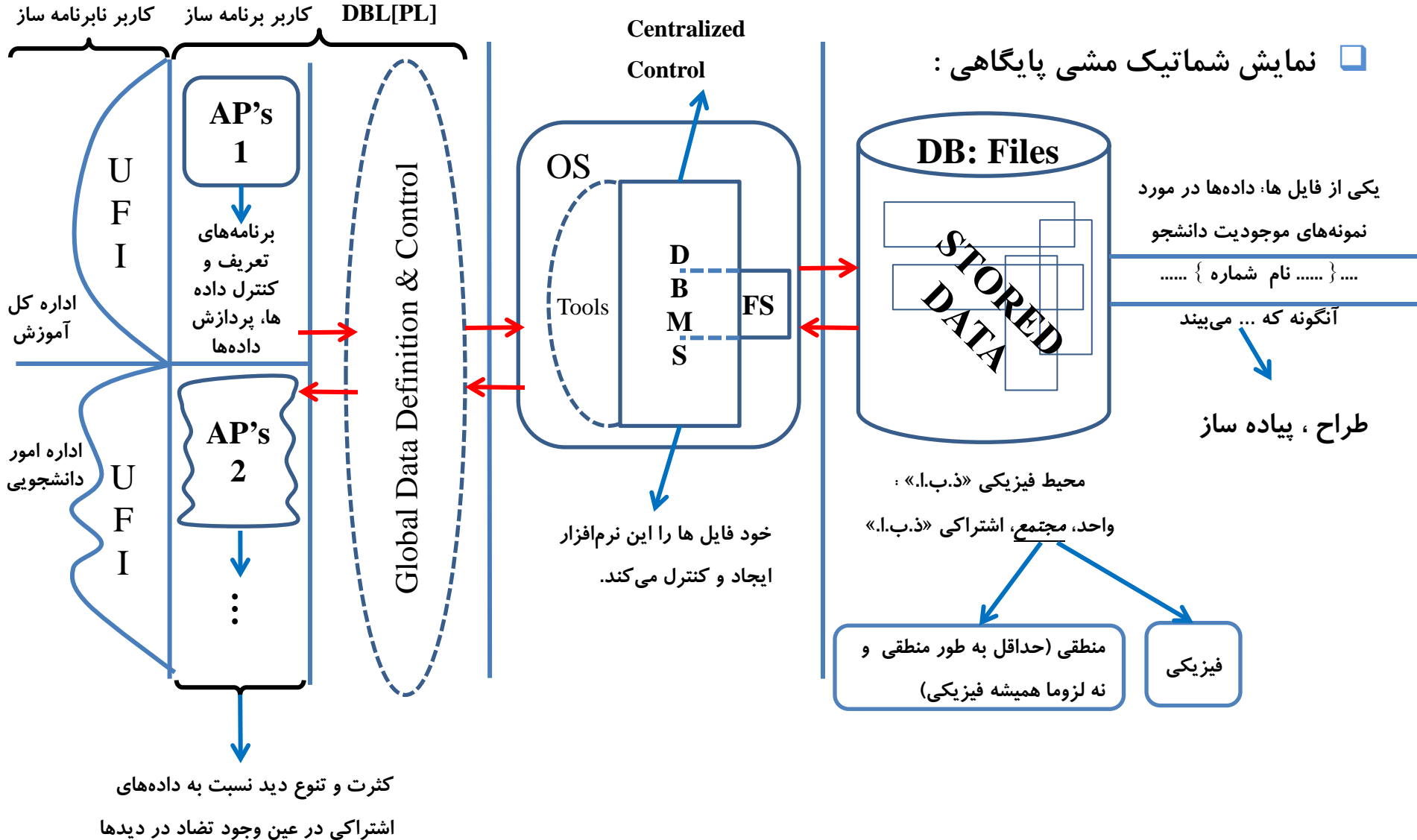
۱۶- تنظیم سیستم پایگاهی (Data Base System Tuning) ← به طور مثال به منظور افزایش کارایی

۱۷- آغاز بهره‌برداری و نگهداری از سیستم

۱۸- گسترش سیستم ← یکی از ویژگی‌های DBMS گسترش‌پذیری سیستم است.

۱۹- رفع معایب و بهینه‌سازی سیستم







چگونه از این کثرت دید می توان به آن «وحدت» رسید؟ ←



۱- خود نرم افزار DBMS

۲- معماری پایگاه داده

تمرین: مزایای مشی پایگاهی چیست؟ ← (طبق معلومات فعلی: عکس معایب مشی فایلینگ)

تمرین: چند سطح تعریف داده داریم؟

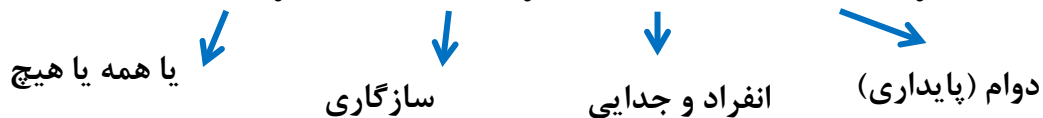


## تراکنش Transaction:



□ دنباله ای از عملیات («قطعه برنامه») که معمولاً حد اقل یک عمل تغییردهنده (درج، حذف، به روزرسانی) در محیط ذخیره‌سازی داده‌ها انجام می‌دهد و باید یا به تمامی اجرا شود و یا اجرا نشده تلقی شود.

□ دارای خواص ACID (Atomicity Consistency Isolation Durability)



شرط سازگاری پایگاه داده در این مثال :  $A+B$  ثابت باشد



**BEGIN TRANS**

**READ (A)**

**A = A - 50**

**UPDATE (A)**

**READ (B)**

**B = B + 50**

**UPDATE (B)**

**END TRANS**



## □ عناصر اصلی محیط پایگاهی:

- ۱- سخت افزار ←
  - ذخیره سازی
  - پردازشگر
  - ارتباطی (همرسانی) Data Communication
- ۲- نرم افزار
- ۳- کاربر
- ۴- داده



- رسانه اصلی: دیسک ترجیحا با تکنولوژی RAID  
(Redundant Array of Inexpensive Disk)

سخت افزار ذخیره سازی:

- رسانه فرعی: نوار مغناطیسی [از جمله برای تولید نسخه های پشتیبان]



اغلب DBMS های امروزی تکنیک های تولید Back up را دارا هستند.

تکنیک های تولید نسخه پشتیبان؟

سطوح مختلف Back up

- کامپیوتر های معمولی از هر رده [ PC, main,... ]

سخت افزار پردازشگر:

- اما ماشین های خاص DB هم داریم : DB Machines

- امکانات محلی: برای ارتباط دستگاه های جانبی با پردازنده

سخت افزار ارتباطی (همرسانی):

- امکانات شبکه ای: برای ایجاد شبکه در سیستم پایگاهی نامتمرکز

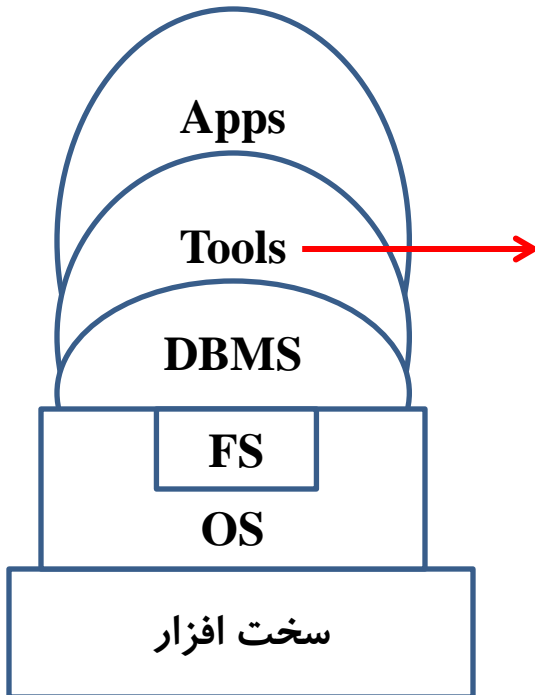
## انواع نرم افزارهای مطرح در محیط پایگاهی:

سیستم عامل و سیستم فایل (FS و OS)

سیستم مدیریت پایگاه داده‌ها (DBMS)

ابزارها (Tools)

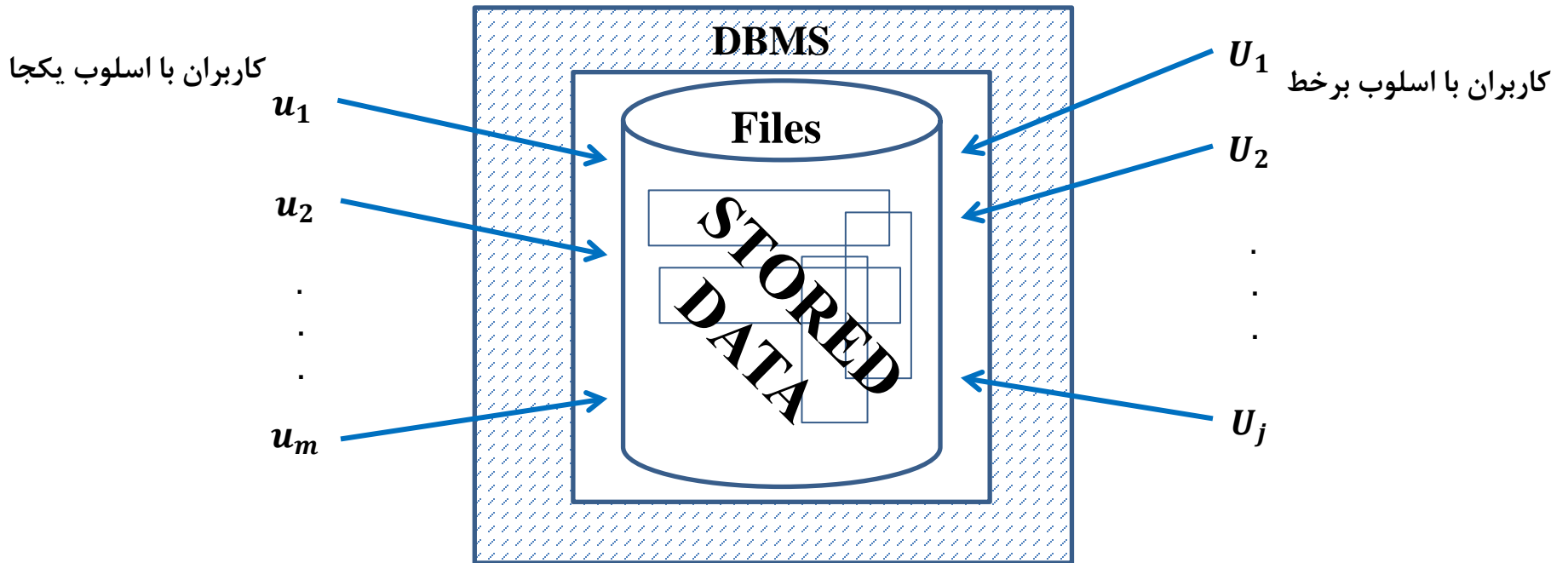
برنامه‌های کاربردی (Apps)



یا با خود DBMS می فروشند،  
یا جداگانه خریداری می شود و به  
امکانات آن اضافه می شود.

تسهیلات نرم افزار

□ در معنای عام هر استفاده‌کننده از سیستم پایگاهی را **کاربر** گوئیم که انواع مختلفی دارد.





□ انواع کاربر از نظر اسلوب عملیاتی:

□ **Batch - یکجا** (تعدادی برنامه یا پرس و جو جمع آوری می شود و به صورت یکجا به سیستم داده می شود و جواب آن بر می گردد).

□ **Online - برخط - پیوسته** (یک برنامه یا پرس و جو به سیستم داده می شود، اجرا می شود، و جوابش برمی گردد).

□ **Interactive - تعاملی** - بسته به اینکه چه جوابی داده شود عمل دیگری از کامپیوتر درخواست می شود.

▪ Online لزوماً Interactive نیست اما Interactive لزوماً Online است.

□ سیستم پایگاهی به صورت پیش فرض چند کاربره (multi-user) است.





□ داده‌های ذخیره شده در یک سیستم پایگاهی عبارتند از:

□ داده‌های کاربران

□ داده‌های سیستمی

□ مباحث مرتبط با داده در محیط پایگاهی در ادامه درس مطرح می‌گردد.



سوال: می‌خواهیم یک سیستم کاربردی پایگاهی ایجاد کنیم. بر اساس کدام معماری ایجاد کنیم؟

در توصیف معماری یک سیستم باید مشخص کنیم که

از چه مولفه‌هایی، از هر مولفه چند عدد و با چه کیفیتی تشکیل شده است،

مولفه‌ها چگونه با هم ترکیب شده‌اند (جنبه ساختاری سیستم)،

مولفه‌ها چگونه با یکدیگر در تعامل هستند (جنبه رفتاری سیستم).

انواع معماری سیستم پایگاهی:

معماری متمرکز

معماری نامتمرکز

▪ معماری مشتری-خدمتگزار

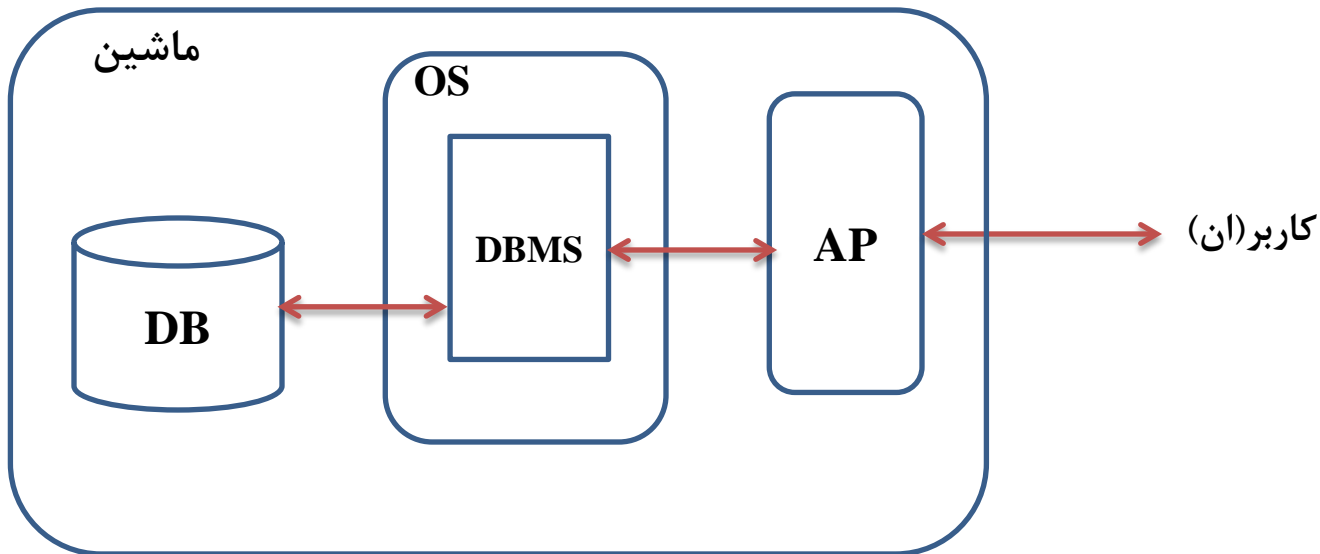
▪ معماری توزیع شده

▪ معماری چندپایگاهی

▪ معماری با پردازش موازی



- در این معماری یک پایگاه داده (متمرکز و مجتمع) روی یک سیستم کامپیوتری و بدون ارتباط با سیستم کامپیوتری دیگر ایجاد می‌شود.
- معمولاً به صورت تک کاربری و برای کاربردهای کوچک و با امکانات محدود از این معماری استفاده می‌شود.





**دلیل:** دلیل اصلی استفاده از معماری مشتری-خدمتگذار (Client-Server): تقسیم وظایف سیستم

**تعریف:** هر ماشینی (فیزیکی یا منطقی) که خدمتی را به ماشین دیگر بدهد، **خدمتگذار** نامیده می‌شود.

نمونه‌هایی از انواع خدمتگذارها: File Server, Print Server, Message Server, DB Server



انواع معماری مشتری – خدمتگذار

معماری تک مشتری – تک خدمتگذار

معماری چند مشتری – تک خدمتگذار

معماری تک مشتری – چند خدمتگذار

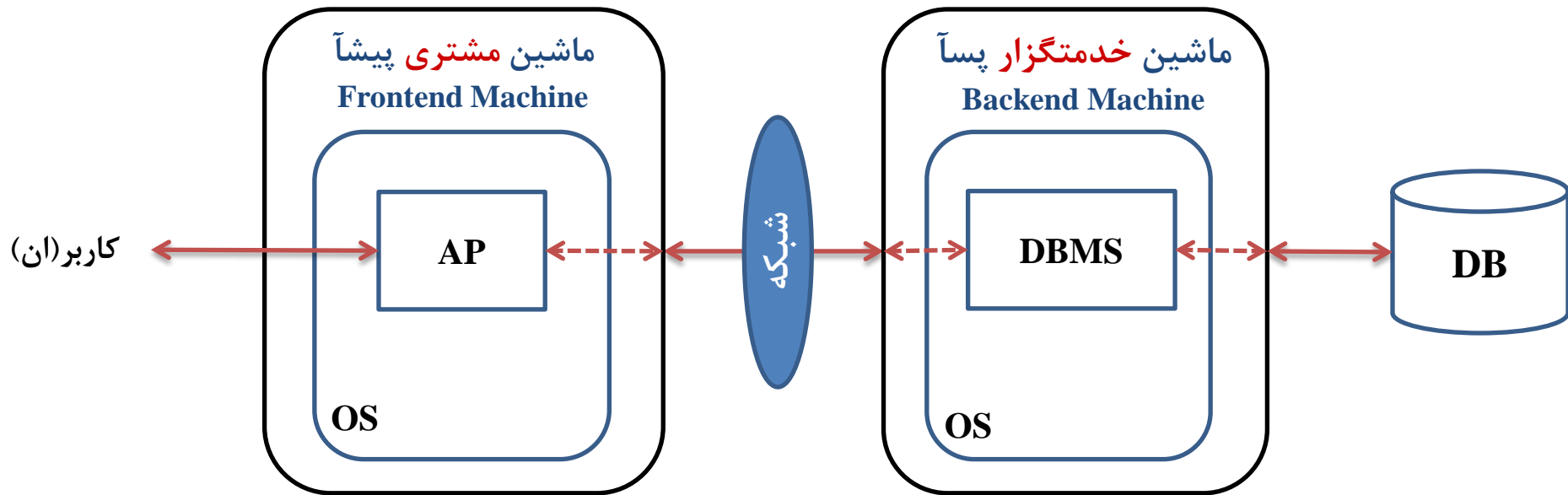
معماری چند مشتری – چند خدمتگذار

معمولا شامل دو سایت:

سایت مشتری: تمام برنامه‌های کاربردی در آن اجرا می‌شوند.

سایت خدمتگذار: تمام داده‌ها در آن ذخیره می‌شوند

به این معماری، **معماری دولایه (2-tier)** نیز گویند.





ماشین‌های ساده، ارزان و حتی بدون دیسک (thin client)

برخی مزایای معماری سه لایه نسبت به دو لایه:

گسترش پذیری بهتر

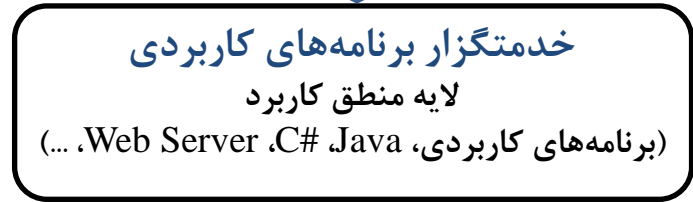
کارایی بالاتر

امنیت داده‌ای بیشتر (عدم ارتباط مستقیم مشتری‌ها با کارگزار داده)

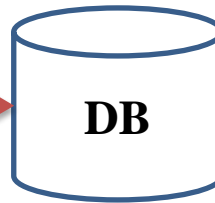
قابلیت کاهش هزینه سخت افزاری (با استفاده از thin client)

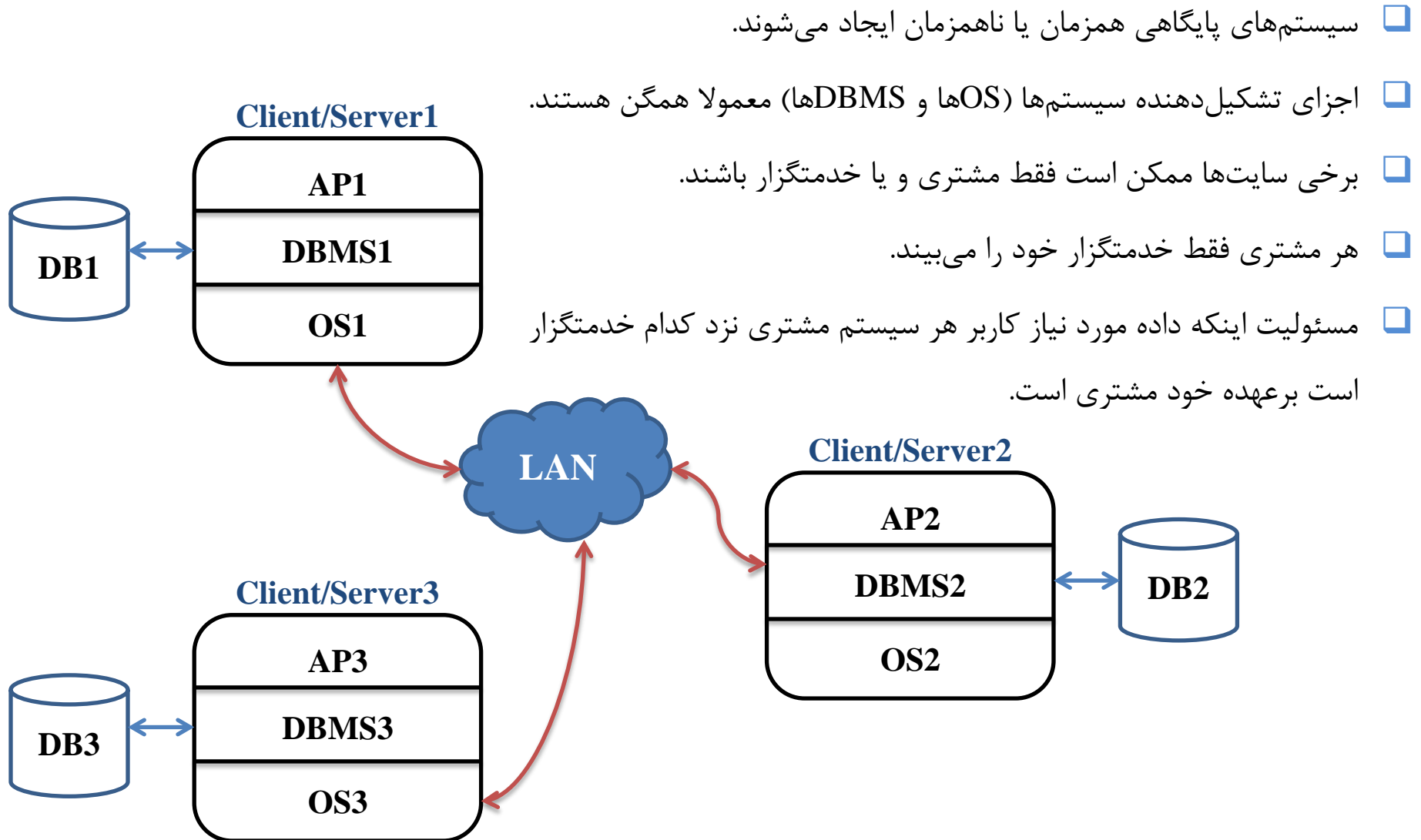


پروتکل HTTP



ODBC, JDBC, SQL, SQL/CLI





سیستم‌های پایگاهی همزمان یا ناهمزمان ایجاد می‌شوند.

اجزای تشکیل دهنده سیستم‌ها (OSها و DBMSها) معمولاً همگن هستند.

برخی سایت‌ها ممکن است فقط مشتری و یا خدمتگزار باشند.

هر مشتری فقط خدمتگزار خود را می‌بیند.

مسئولیت اینکه داده مورد نیاز کاربر هر سیستم مشتری نزد کدام خدمتگزار

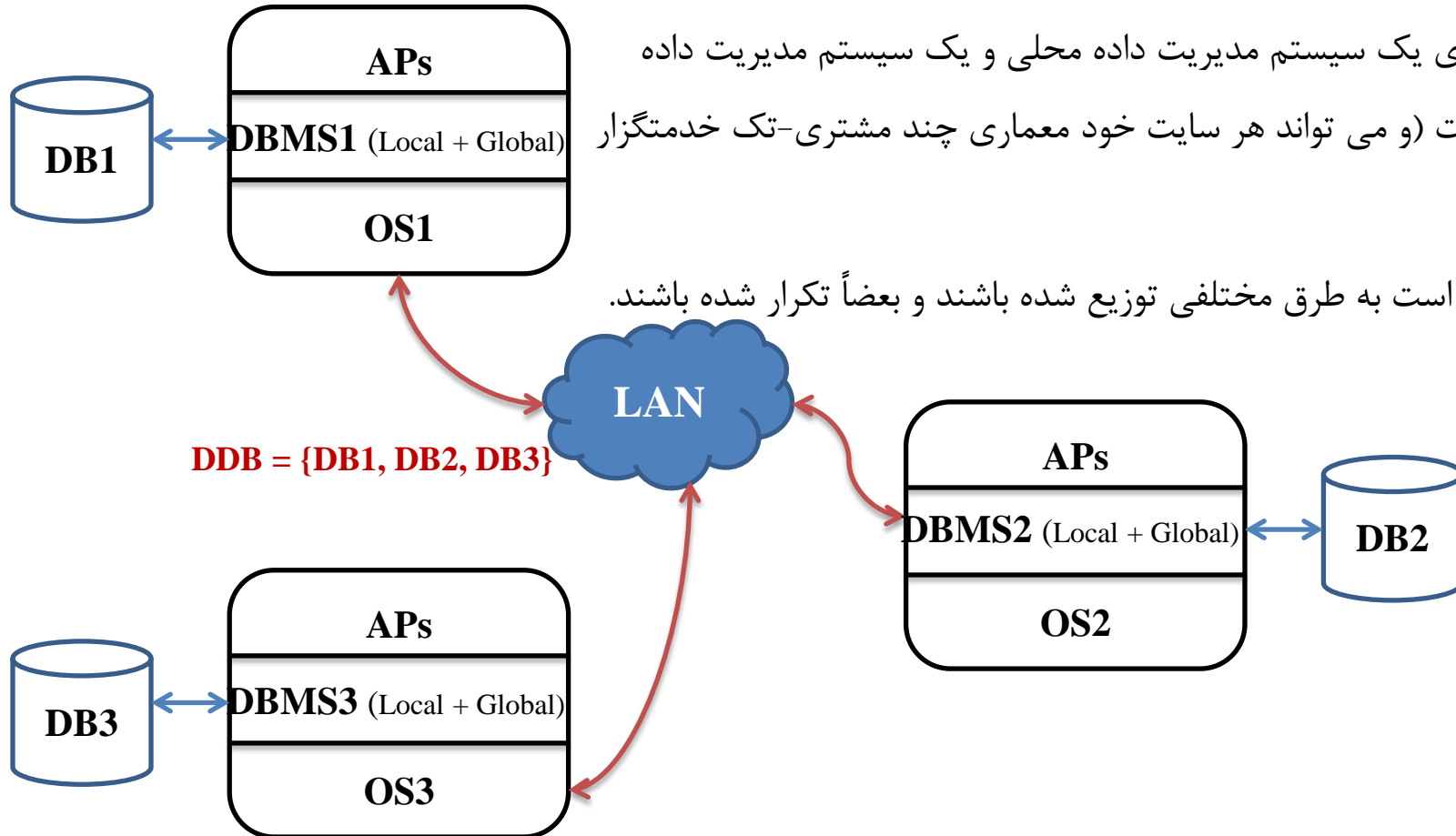
است برعهده خود مشتری است.

مجموعه‌ای است از چند پایگاه داده منطقاً یکپارچه (مجتمع)، ولی به طور فیزیکی توزیع شده روی یک شبکه کامپیوتری.

توزیع شدگی از دید برنامه‌ها و کاربران پایگاه داده پنهان است.

هر سایت دارای یک سیستم مدیریت داده محلی و یک سیستم مدیریت داده توزیع شده است (و می‌تواند هر سایت خود معماری چند مشتری-تک خدمتگذار داشته باشد).

داده‌ها ممکن است به طرق مختلفی توزیع شده باشند و بعضاً تکرار شده باشند.







**پرسش و پاسخ ...**

**amini@sharif.edu**