

بسم الله الرحمن الرحيم

نام درس: اصول طراحی پایگاه داده‌ها

نام منبع:

مفاهیم بنیادی پایگاه داده‌ها

مؤلف: سید محمد تقی روحانی رانکوهی

تهیه کننده:

دکتر احمد فراهی

فهرست جلسات

جلسه اول: مفاهیم پایگاه داده‌ها

جلسه دوم: مدل‌سازی معنایی داده‌ها

جلسه سوم: محدودیتهای روش **ER**

جلسه چهارم: پایگاه داده در محیط انتزاعی

جلسه پنجم: معماری پایگاه داده‌ها

جلسه ششم: سیستم مدیریت پایگاه داده‌ها

جلسه هفتم: **DBMS** در یک سیستم کامپیوتری

جلسه هشتم: معماری سیستم پایگاه داده‌ها

فهرست جلسات

جلسه نهم: مدل رابطه‌ای

جلسه دهم: قواعد جامعیت پایگاه داده‌ها

جلسه یازدهم: زبان **SQL**

جلسه دوازدهم: ادامه زبان **SQL**

جلسه سیزدهم: دید در مدل رابطه‌ای

جلسه چهاردهم: طراحی پایگاه داده‌های رابطه‌ای

جلسه پانزدهم: نرمال‌سازی رابطه‌ها

جلسه شانزدهم: طراحی فیزیکی پایگاه داده‌ها

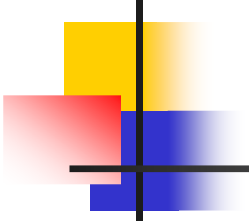
اهداف این درس

در این درس به بخشی از مفاهیم مبنایی دانش و تکنولوژی پایگاه داده‌ها و اصول مدل‌سازی و طراحی آن پرداخته می‌شود. پایگاه داده و عناصر اصلی محیط آن معرفی می‌شود. با مدل‌سازی معنایی داده‌ها و محیط انتزاعی آشنا می‌شویم. سطوح معماریهای پایگاه داده‌ها بیان می‌شود. سیستم مدیریت پایگاه داده‌ها (DBMS) و اجزاء آن شرح داده می‌شود. به انواع معماریهای سیستم پایگاه‌ها و مفاهیم اساسی مدل رابطه‌ای پرداخته می‌شود. با زبان SQL به عنوان یک زبان رابطه‌ای و همچنین با دیدهای رابطه‌ای آشنا می‌شویم و در نهایت طراحی پایگاه داده‌ها به روش بالا به پایین، سنتز و طراحی فیزیکی را خواهیم دید.



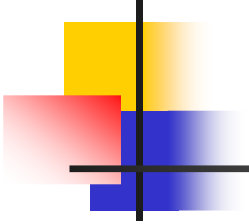
جلسه اول

مفاهیم پایگاه داده‌ها



آنچه در این جلسه می خوانید:

- 1- سیستم ذخیره و بازیابی اطلاعات در معنای عام
- 2- رده های تکنولوژیکی سیستم مدیریت پایگاه داده ها
- 3- داده
- 4- تعریف اطلاع
- 5- تعریف دانش
- 6- تعریف پایگاه داده ها



آنچه در این جلسه می خوانید:

7- مراحل کلی کار در مشی فایلینگ

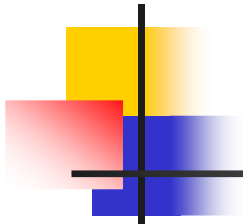
8- معایب مشی فایلینگ

9- مراحل کلی کار در مشی پایگاهی

10- عناصر محیط پایگاه داده‌ها

11- انواع سخت افزارهای محیط پایگاه داده

12- انواع نرم افزارهای موجود در محیط پایگاه داده‌ها



هدفهاي كلي: مقدمه و آشنائي با مفاهيم پايگاه داده‌ها

هدفهاي رفتاري: دانشجو در پايان اين جلسه مي‌تواند:

- داده ، اطلاع ، شناخت ، سيستم ذخيره و بازيايي و پايگاه داده‌ها را تعريف کند.
- رده‌هاي تكنولوجيكي پايگاه داده را بيان کند.
- رهيافتهاي ايجاد يك سيستم کاربردي را ارائه کند.



اصطلاح پایگاه داده‌ها یکی از رایج‌ترین اصطلاحات در
دانش و فن کامپیوتر است

در این درس دانشجویان تنها با بخشی از مفاهیم بنیادی
دانش و فن پایگاه داده‌ها آشنا شده، آگاهی پایه‌ای لازم را برای
مطالعه بیشتر و یا کار در این زمینه کسب می‌کنند.



سیستم مدیریت پایگاه داده‌ها یکی از سیستم‌های ذخیره و بازیابی اطلاعات است.



سیستم ذخیره و بازیابی اطلاعات در معنای عام:

هر سیستمی که به کاربر برنامه‌ساز یا نابرمه‌ساز امکان دهد
تا اطلاعات خود را ذخیره ، بازیابی و پردازش کند.



رده‌های تکنولوژیکی سیستم مدیریت پایگاه داده‌ها

- 1- سیستم فایلینگ
- 2- سیستم مدیریت داده‌ها
- 3- سیستم مدیریت پایگاه داده‌ها
- 4- سیستم مدیریت پایگاه شناخت
- 5- سیستم مدیریت پایگاه داده‌های شیء‌گرا
- 6- سیستم هوشمند مدیریت پایگاه داده‌ها
- 7- سیستم معنایی مدیریت پایگاه داده‌ها
- 8- سیستم مدیریت پایگاه داده‌های زمانبند
- 9- سیستم مدیریت پایگاه داده‌های نیم‌ساختمند و ناساختمند



ادامه



رده‌های تکنولوژیکی سیستم مدیریت پایگاه داده‌ها

- 10- سیستم مدیریت پایگاه داده‌های بی درنگ
- 11- سیستم داده‌کاوی و کشف شناخت
- 12- سیستم مدیریت چند پایگاه‌های
- 13- سیستم اطلاعات اجرایی
- 14- سیستم فعال مدیریت پایگاه داده‌ها
- 15- سیستم مدیریت پایگاه داده‌های شیء-رابطه‌ای



داده

تعریف اول- نمایش ذخیره شده اشیاء فیزیکی ، چیزهای مجرد ،
بوده ها ، رویدادها یا چیزهای قابل مشاهده که در تصمیم سازی
بکار می آیند.



داده

تعریف دوم- هر مجموعه‌ای از بوده‌ها



داده

تعریف سوم- بوده‌های خام که معنای اندکی دارند مگر اینکه
به صورت منطقی سازمان‌دهی شده باشند



تعریف داده از دیدگاه ANSI

نمایش بوده‌ها ، پدیده‌ها ، مفاهیم یا شناخته‌ها به طرزی صوری و مناسب برای برقراری ارتباط ، تفسیر یا پردازش توسط انسان یا هر امکان خودکار

هر نمایشی اعم از کاراکتری یا کمیتهای قیاسی که معنایی به آن قابل انتساب باشد.



تعريف اطلاع

اطلاع به داده‌اي اطلاق مي‌شود كه توسط يك فرد يا سازمان
براي تصميم‌گيري بكار مي‌رود

اطلاع ، داده پردازش شده است.

اطلاع عبارت است از داده سازمان يافته‌اي كه شناختي را منتقل
مي‌كند



تعریف دانش

دانش عبارتست از نمایش نمادین جنبه‌هایی از
بخشی از خرد جهان واقع



تعريف پایگاه داده‌ها

مجموعه‌ای است از داده‌های ذخیره شده و پایا ، به صورت مجتمع (یکپارچه) (نه لزوماً فیزیکی ، بلکه حداقل به طور منطقی) ، بهم مرتبط ، با کمترین افزونگی ، تحت مدیریت یک سیستم کنترل متمرکز ، مورد استفاده یک یا چند کاربر از یک یا بیش از یک “سیستم کاربردی” ، به طور همزمان و اشتراکی



براي ايجاد يك سيستم کاربردي دو رهيافت وجود دارد:

1- رهيافت سنتي يا مشي فايلينگ

2- رهيافت (مشي) پاىگاهي



مراحل کلی کار در مشی فایلینگ

- تحلیل و بررسی نیازهای اطلاعاتی و پردازشی هر قسمت به طور جداگانه
- اجرای مراحل کلاسیک اولیه لازم برای طراحی و تولید یک سیستم کاربردی
- تعیین مشخصات هر سیستم و وظایف آن
- طراحی تعدادی فایل
- نوشتن مجموعه‌ای از برنامه‌های ایجاد، کنترل و پردازش فایل



ادامه



مراحل کلی کار در مشی فایلینگ

- استفاده از يك پیکربندی سخت افزاری و نرم افزاری مشخص
- انجام تستهای لازم و تنظیم سیستم کاربردی
- ایجاد يك سیستم کاربردی برای هر قسمت و برپایی محیط فیزیکی ذخیره و بازیابی اطلاعات و سیستم بهره برداری از آن خاص همان قسمت.



معایب مشی فایلینگ

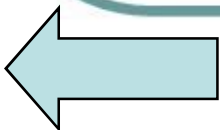
1- عدم وجود محیط مجتمع ذخیره سازی اطلاعات و عدم وجود سیستم یکپارچه

2- عدم وجود سیستم کنترل متمرکز روی کل داده ها

3- افزونگی

4- عدم وجود ضوابط ایمنی کارا و مطمئن

5- خطر بروز پدیده ناسازگاری داده ها



ادامه



معایب مشی فایلینگ

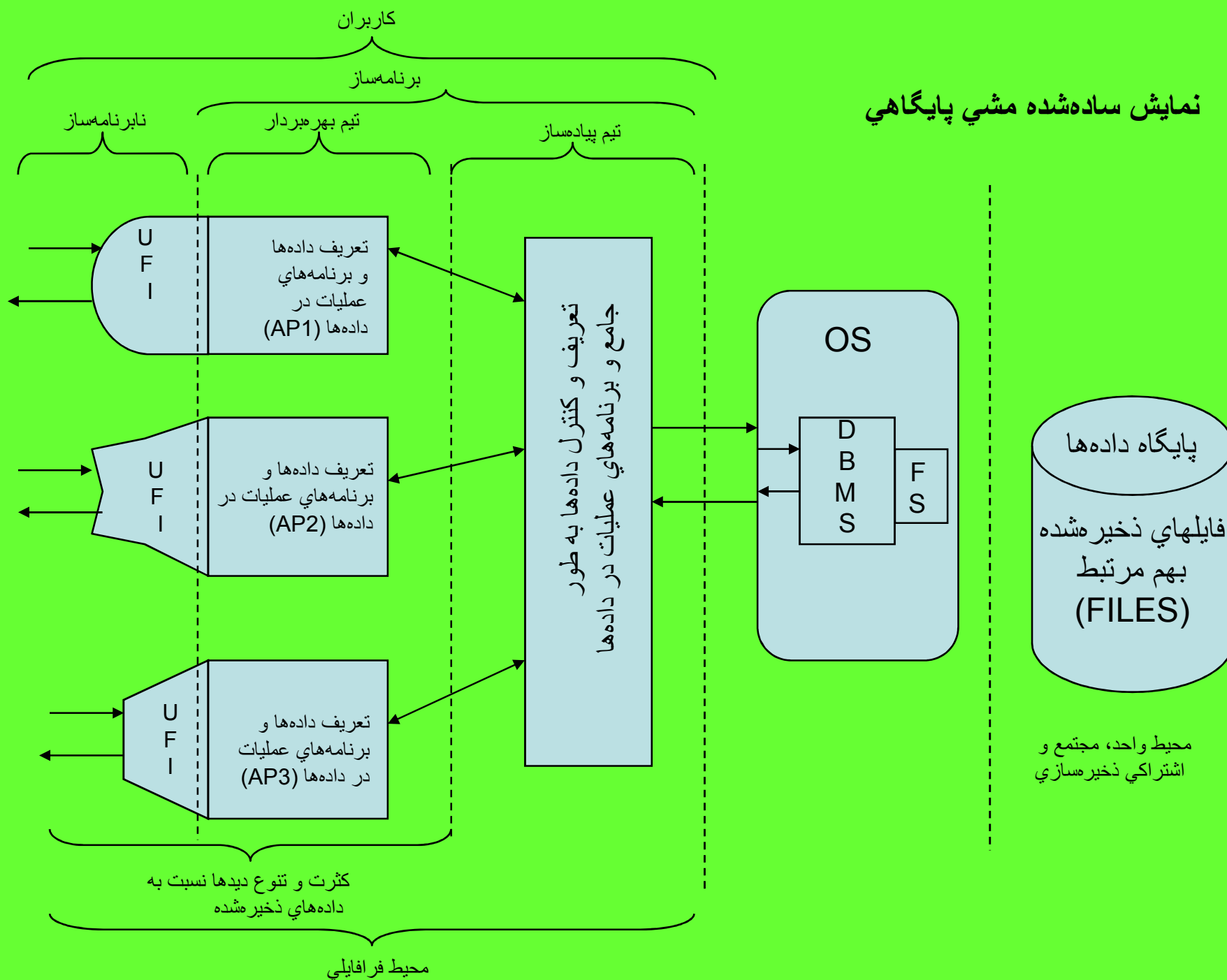
6- عدم امکان اشتراکي شدن داده‌ها

7- مصرف نابهينه امکانات سخت افزاري و نرم افزاري

8- حجم زياد برنامه سازي

9- وابستگي برنامه هاي کاربردي به محيط ذخيره سازي داده ها

نمایش ساده شده مشی پایگاهی





مراحل كلي كار در مشي پايگاهي

- بررسي و تحليل نيازهاي پردازشي و اطلاعاتي همه قسمت‌ها توسط يك گروه
- مدلسازي معنايي داده‌ها
- تعيين مشخصات جامع (يكپارچه) کاربردي و وظائف آن
- انتخاب يك يا چند پيكربندي سخت‌افزاري-نرم‌افزاري
- استفاده از يك يا چند DBMS
- طراحي پايگاه داده‌ها در سطوح لازم



ادامه

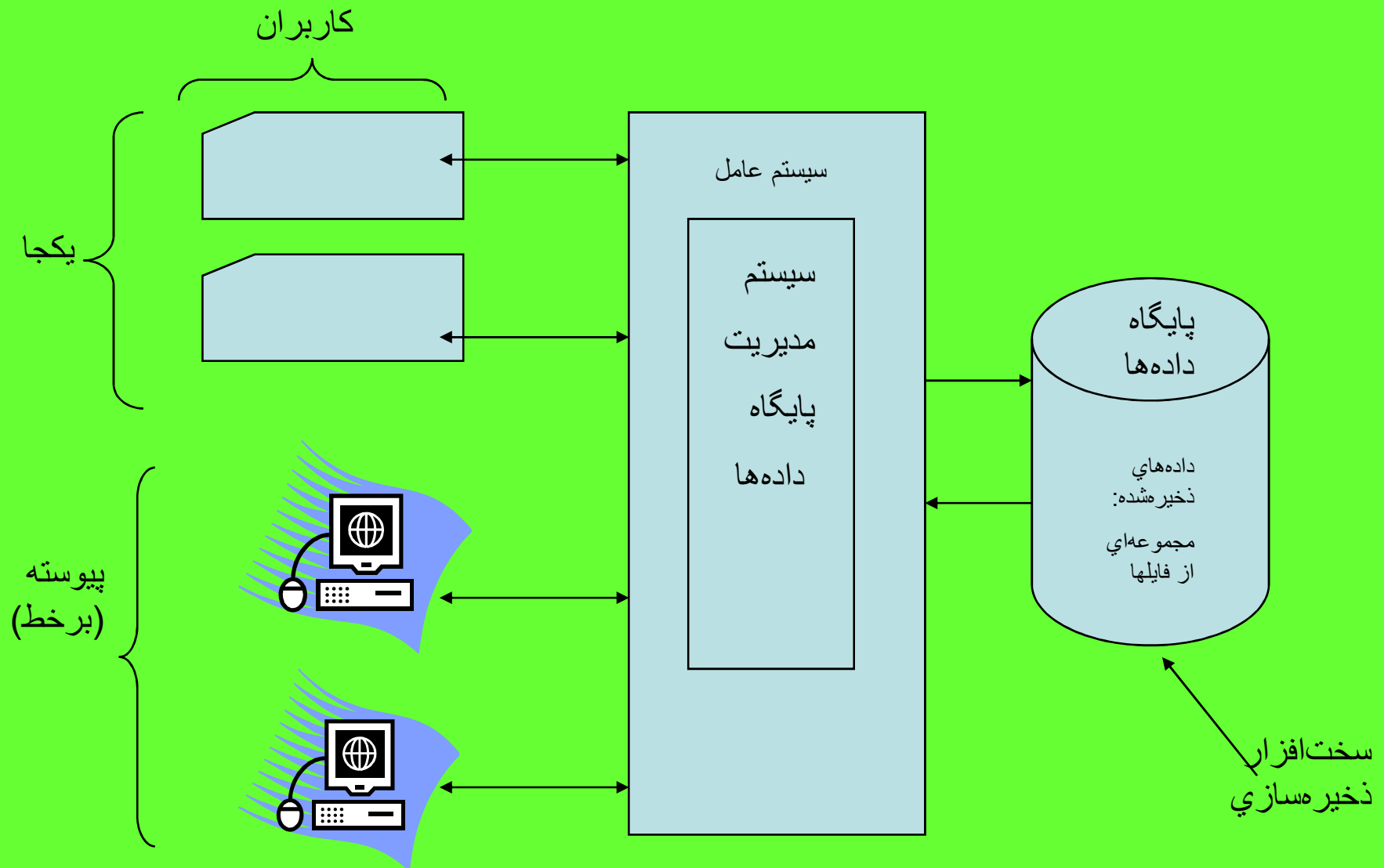


مراحل کلی کار در مشی پایگاهی

- تولید مجموعه‌ای از برنامه‌های ایجاد و کنترل پایگاه داده
- ایجاد محیط واحد و مجتمع ذخیره‌سازی و مشترک بین کاربران
- طراحی و تولید واسطه‌های کاربرپسند مورد نیاز
- تعریف پایگاه داده هر قسمت توسط کاربر مربوطه
- طراحی برنامه‌های عملیات در پایگاه داده
- بهره‌برداری واقعی از سیستم پس از تستهای لازم

عناصر محیط پایگاه داده‌ها

1- سخت افزار 2- نرم افزار 3- کاربر 4- داده





انواع سخت افزارهاي محيط پايگاه داده

1- سخت افزار ذخيره سازي داده ها

2- سخت افزار پردازشگر

3- سخت افزار هم رسانش (ارتباط)



انواع نرم افزارهاي موجود در محيط پايگاه داده ها

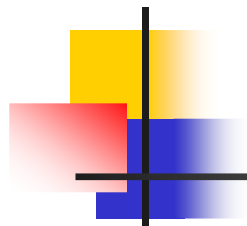
- 1- سيستم مديريت پايگاه داده ها (DBMS)
- 2- برنامه هاي کاربردي قابل اجرا در محيط DBMS
- 3- رويه هاي ذخيره شده
- 4- نرم افزار شبکه



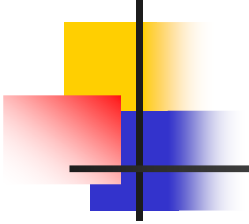
جلسه دوم

مدلسازی معنایی داده‌ها

آنچه در این جلسه می خوانید:



- 1- مدل سازی معنایی داده ها
- 2- انواع روش های مدل سازی معنایی داده ها
- 3- سه مفهوم معنایی موجود در روش **ER**
- 4- تعریف موجودیت
- 5- سه ضابطه در رابطه با تشخیص یک نوع موجودیت
- 6- موجودیت مستقل و وابسته
- 7- تعریف صفت



آنچه در این جلسه می خوانید:

8- انواع صفت و تعاریف هریک

9- ارتباط

10- خصوصیات نوع ارتباط

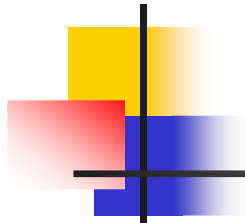
11- نمودار **ER**

12- نمادهای رسم نمودار **ER**

13- وضع مشارکت در ارتباط

14- درجه آن ارتباط

15- چندی یا ماهیت نوع ارتباط



هدفهاي كلي: آشنائي با مدل سازي معنائي داده ها

هدفهاي رفتاري: دانشجو در پايان اين جلسه مي تواند:

- مدل سازي معنائي پايگاه داده ها و انواع آن را بيان کند.
- روش مدل سازي ER و مفاهيم آن را تشریح کند.
- نمودار ER و نمادهاي آن را رسم کند.



مدلسازي معنایي داده‌ها

داده‌هاي ذخيره‌شدي در پایگاه داده‌ها ابتدا باید
در بالاترين سطح انتزاع مدلسازي معنایي شوند.



انواع روشهای مدلسازی معنایی داده‌ها

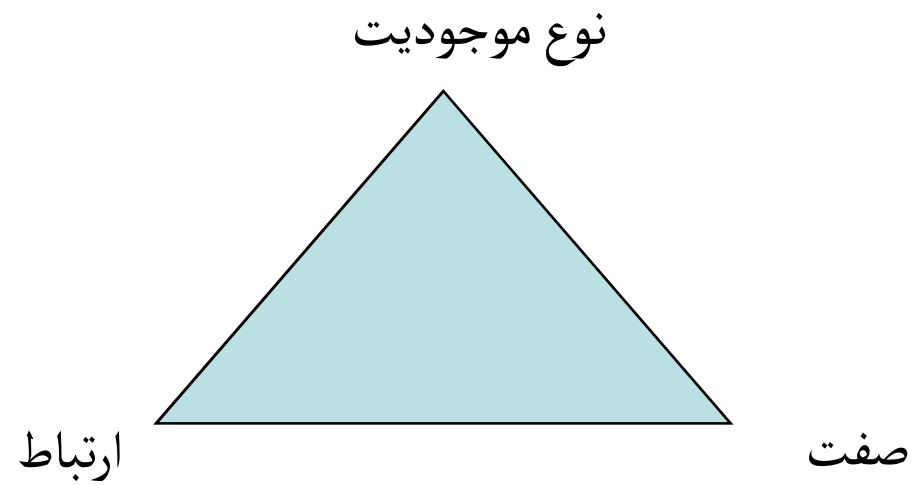
روش موجودیت-ارتباط (ER)

روش زبان عمومی مدلسازی (UML)

روش تکنیک مدلسازی شیئی (OMT)



سه مفهوم معنایی موجود در روش ER





تعریف موجودیت

مفهوم کلی شیء ، چیز ، پدیده و به طور کلی هر آنچه که
می‌خواهیم در موردش اطلاع داشته باشیم و شناخت خود را در
موردش افزایش دهیم.



سه ضابطه در رابطه با تشخیص يك نوع موجودیت

- 1- معمولاً نمونه‌هایی متمایز از یکدیگر دارند.
- 2- معمولاً بیش از يك صفت دارد و کاربر به مجموعه‌ای از اطلاعات در مورد آن نیاز دارد.
- 3- معمولاً حالت کنشگری (فاعلیت) یا حالت کنشپذیری (مفعولیت) دارد.



موجودیت مستقل و وابسته

موجودیت مستقل (قوي)، موجوديتي است که مستقل از هر موجودیت دیگر و به خودي خود، در يك محيط مشخص مطرح باشد.

موجودیت وابسته (ضعيف)، موجوديتي است که وجودش وابسته به يك نوع موجودیت دیگر است.



تعريف صفت

خصيصه يا ويژگي يك نوع موجوديت است و هر نوع
موجوديت مجموعه‌اي از صفات دارد. هر صفت يك نام ،
يك نوع و يك معنای مشخص دارد.



ساده یا مرکب

تک مقداری یا چندمقداری

شناسه یا ناشناسه موجودیت

هیچ مقدارپذیر یا ناپذیر

ذخیره شده یا مشتق

انواع صفت



مقدار صفت ساده از لحاظ معنایی تجزیه‌نشده یا اتمیک است.

صفت مرکب از چند صفت ساده تشکیل شده است.



صفت تڪمقداري ، صفتي است كه براي يك نمونه از يك نوع
موجوديت حداكثر يك مقدار از دامنه مقادير را مي گيرد

صفت چندمقداري بيش از يك مقدار از دامنه مقادير
مي گيرد.



صفت شناسه موجودیت که گاه به آن

کلید هم گفته می شود ، دو ویژگی دارد:

1- یکتایی مقدار دارد.

2- حتی الامکان طول مقادیرش کوتاه است.



صفت هیچ مقدارپذیر

هیچ مقدار یعنی مقدار ناشناخته ، مقدار غیرقابل اعمال ،
مقدار تعریف نشده.

اگر مقدار يك صفت در يك يا بیش از يك نمونه از يك نوع
موجودیت ، برابر با هیچ مقدار باشد ، آن صفت هیچ مقدارپذیر
است.



صفت ذخیره شده و مشتق

صفت ذخیره شده صفتی است که مقادیرش در پایگاه داده ها ذخیره شده باشد.

صفت مشتق ، صفتی است که مقادیرش در پایگاه داده ها ذخیره نشده باشد ، بلکه حاصل يك پردازش روی فقره هایی از داده های ذخیره شده باشد.



ارتباط

تعریف- اندرکنش (تعامل) بین دو یا بیش از دو نوع
موجودیت است و ماهیتا نوعی بستگی بین انواع
موجودیتهاست



خصوصیات نوع ارتباط

- 1- هر ارتباط يك نام دارد
- 2- هر ارتباط يك معنای مشخص دارد و این معنا با معنای هر ارتباط دیگر متفاوت است.
- 3- هر ارتباط نمونه‌هایی دارد.



نمودار ER

نموداري است که سه مفهوم اساسي مدل ER، يعني نوع
موجوديت، صفت و ارتباط نمايش داده مي شوند



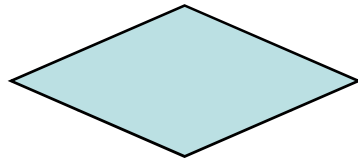
نمادهای رسم نمودار ER



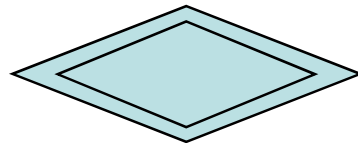
نوع موجودیت



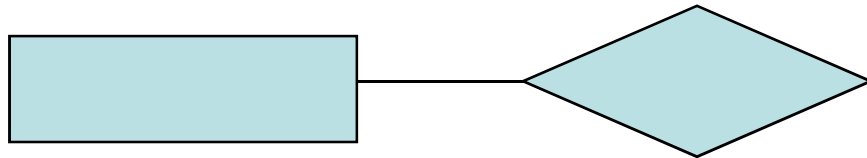
نوع موجودیت ضعیف



نوع ارتباط



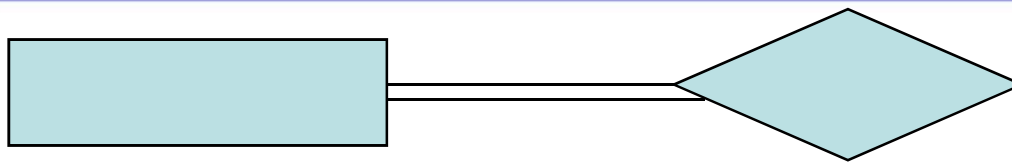
نوع ارتباط با موجودیت ضعیف



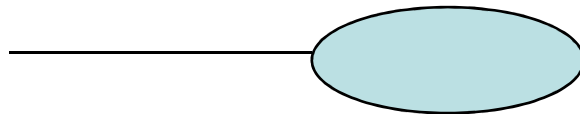
مشارکت نوع موجودیت در نوع ارتباط



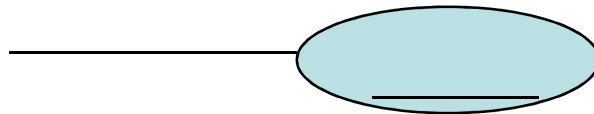
نمادهای رسم نمودار ER



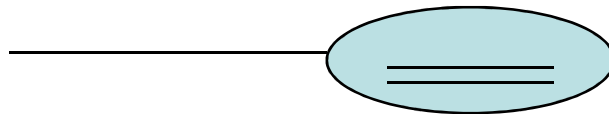
مشارکت الزامی



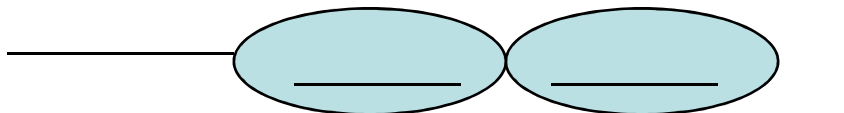
صفت



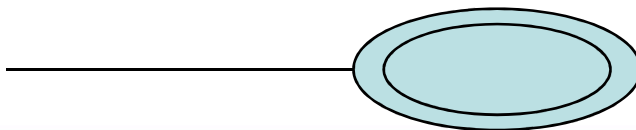
صفت شناسه اول



صفت شناسه دوم



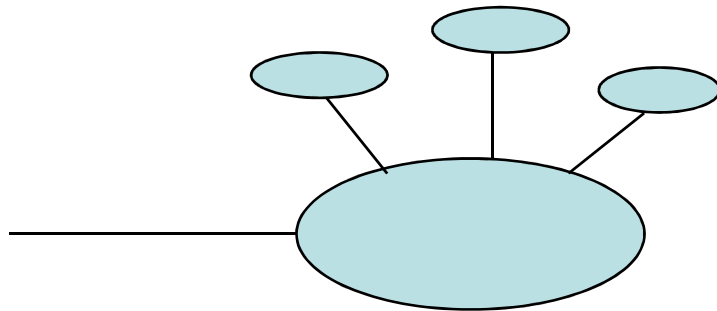
صفت شناسه مرکب



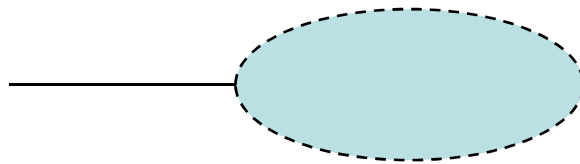
صفت چندمقداری



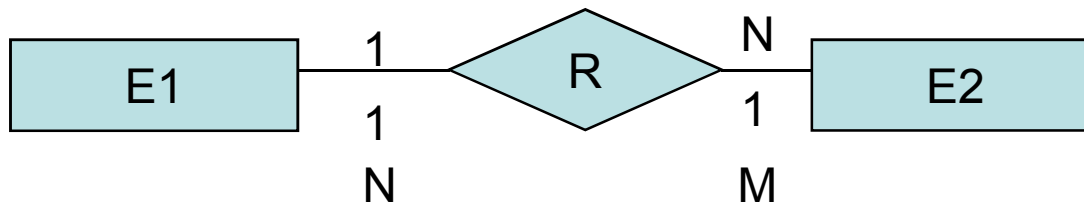
نمادهای رسم نمودار ER



صفت مرکب

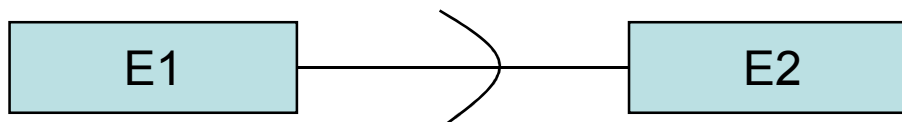


صفت مشتق



1 به N
1 به 1
N به M

چندین ارتباط



ارتباط "گونه‌ای است از"



وضع مشاركت در ارتباط

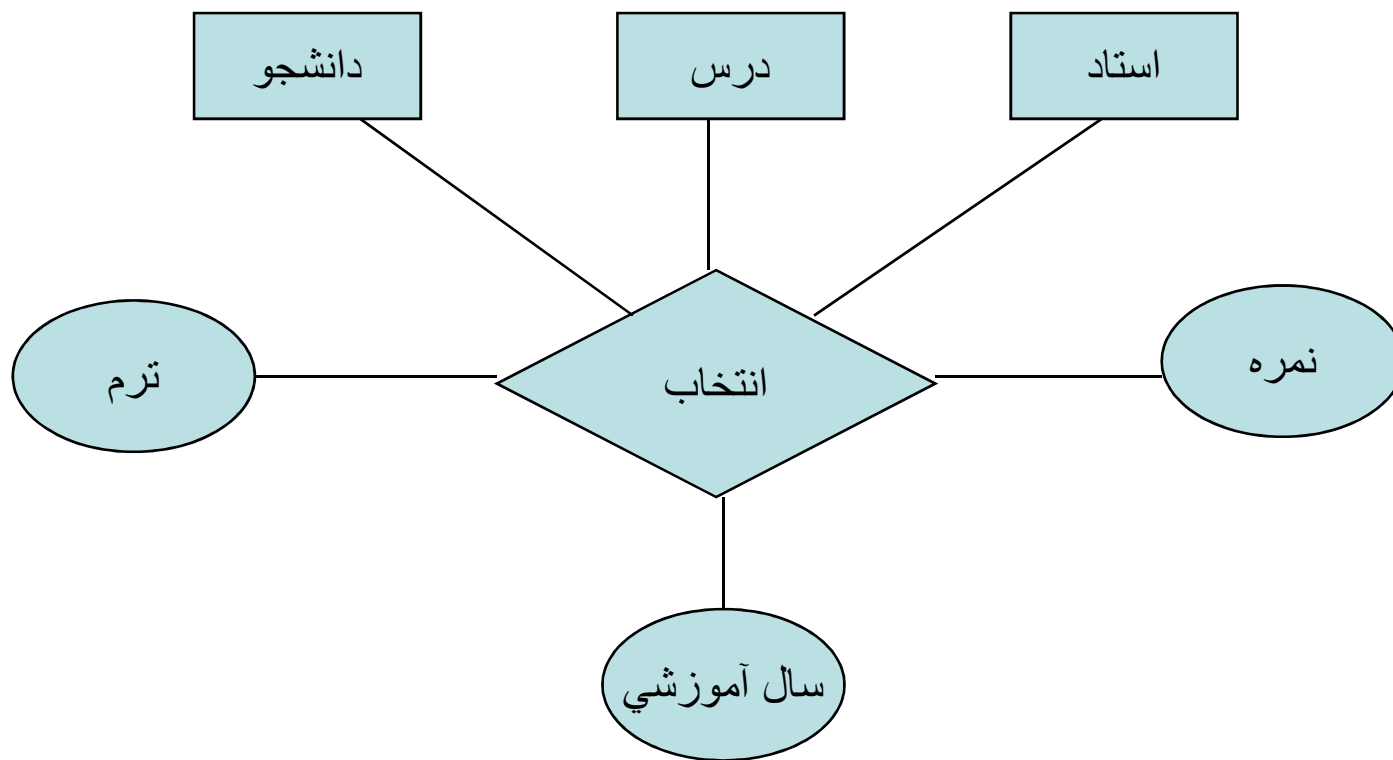
مشاركت يك نوع موجوديت در يك نوع ارتباط را الزامي گویند ، اگر تمام نمونه‌هاي آن نوع موجوديت در آن نوع ارتباط شركت كنند. در غير اين صورت مشاركت غيرالزامي است.



نمایش مشاركت الزامي



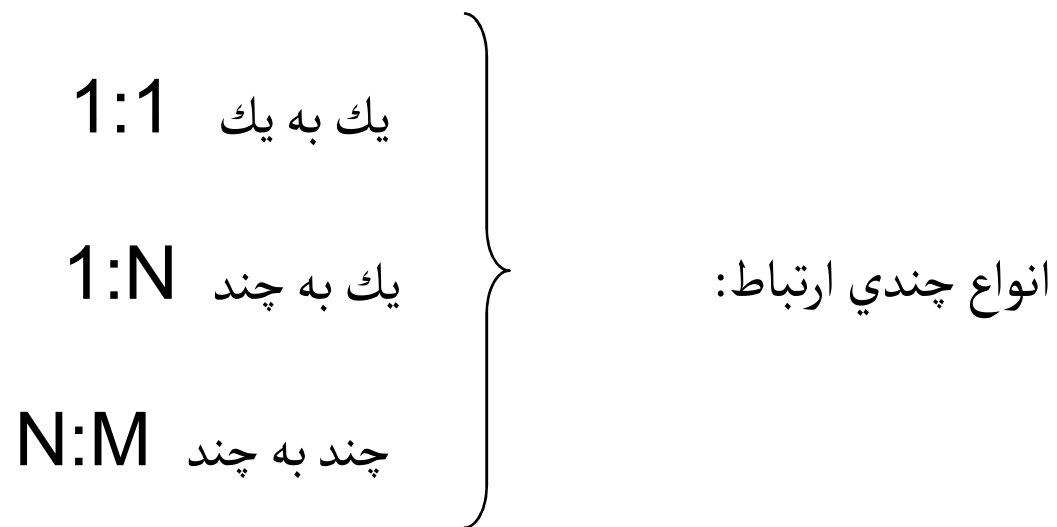
تعداد شرکت کنندگان در يك ارتباط را درجه آن ارتباط مي گویند.



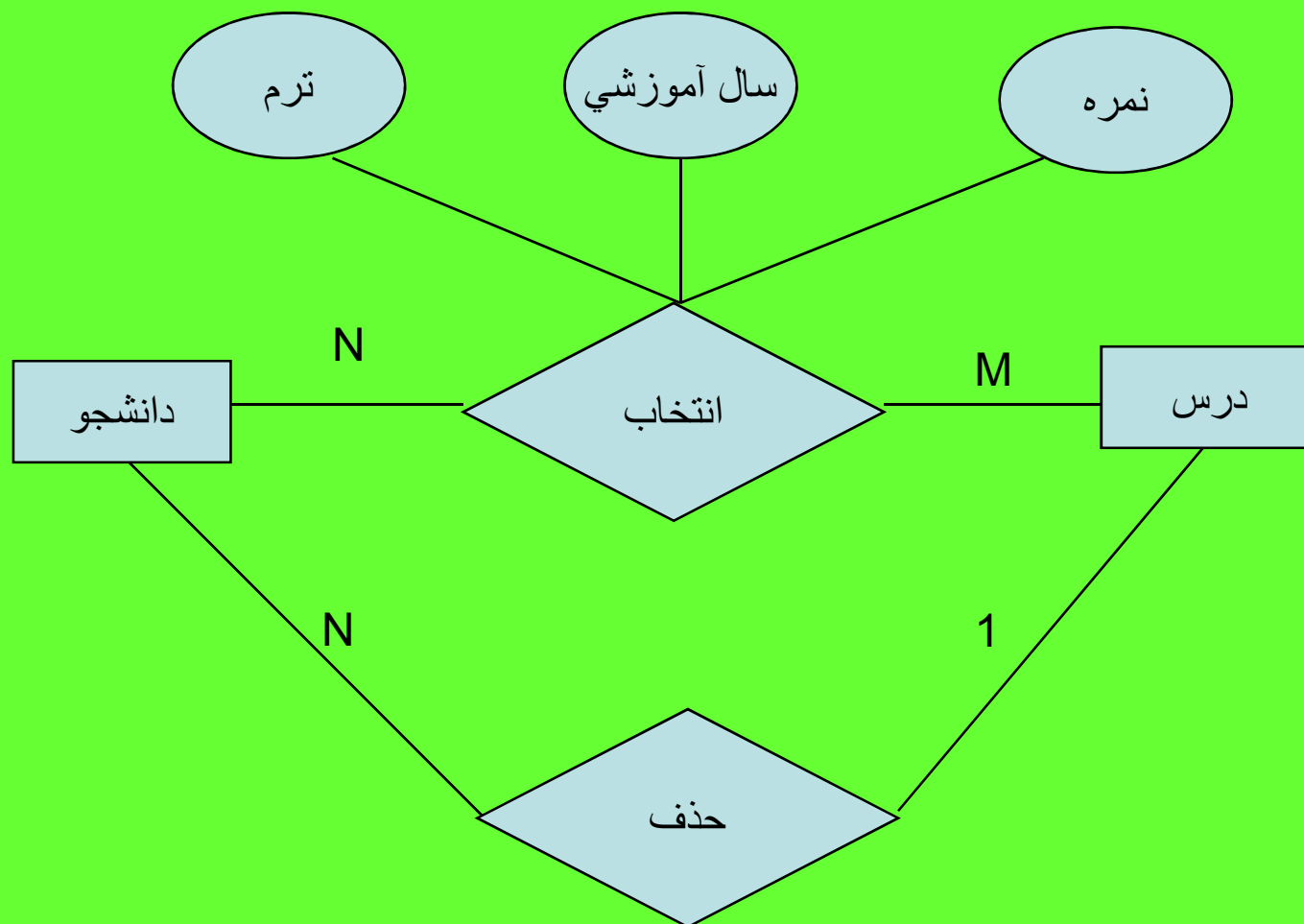
ارتباط بين سه موجوديت



چندي يا ماهيت نوع ارتباط عبارتست از چگونگي تناظر بين دو مجموعه نمونه‌هاي آن دو نوع موجوديت.



نمایش چندی ارتباط

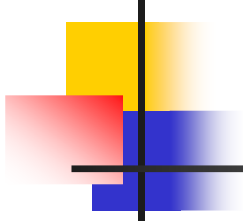


An abstract background featuring a large, textured green square on the left side. To the right, there are several orange and yellow rectangular shapes, some with dark, textured borders. The overall style is painterly and modern.

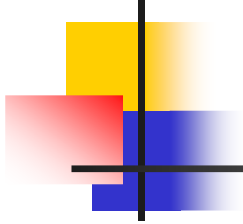
جلسه سوم

محدودیت‌های روش ER

آنچه در این جلسه می خوانید:



- 1- مشکلات روش **ER** (سه نوع دام)
- 2- تجزیه و ترکیب
- 3- تخصیص و تعمیم
- 4- تجمع
- 5- وراثت صفت
- 6- دسته بندی
- 7- مراحل مدل سازی معنایی داده ها
- 8- روش مدل سازی **UML**
- 9- نمادها
- 10- خصوصیات کلی روش مدل سازی معنایی داده ها



هدفهای کلی: مشکلات و محدودیتهای روش ER

هدفهای رفتاری: دانشجو در پایان این جلسه میتواند:

- مشکلات روش ER (انواع دامها) را تشریح کند.
- محدودیتهای روش ER را که در روش EER برطرف شدند، بیان کند.
- مراحل مدلسازی معنایی دادهها را توصیف کند.
- مدلسازی UML و نمادهای به کار رفته در آن را شرح دهد.
- خصوصیات کلی مدلسازی معنایی دادهها را بیان کند.



مشکلات روش ER

1- دام حلقه‌ای

هنگامی ایجاد می‌شود که با داشتن مثلاً سه ارتباط دو موجودیتی، وجود یک ارتباط سه موجودیتی را نتیجه بگیریم در وضعی که این استنتاج درست نباشد



مشکلات روش ER

2- دام چندشاخه (چتری)

این نوع دام وقتی ایجاد می شود که بین یک نوع موجودیت E و هر یک از دیگر انواع موجودیت F ، G و ... ارتباط $1:N$ با مشارکت الزامی وجود داشته باشد، ولی ارتباط بین مثلاً F و G ، در مدلسازی دیده نشده باشد.



مشکلات روش ER

3- دام شکاف

هنگامی ایجاد می شود که بین دو نوع موجودیت E و F ، یک ارتباط باچندی $1:N$ و مشارکت الزامی وجود داشته باشد، ولی F خود با نوع موجودیت G ، ارتباط $1:N$ با مشارکت غیرالزامی داشته باشد. به دلیل غیر الزامی بودن ارتباط بین F و G ، نمی توان همه اطلاعات دو موجودیتی در مورد ارتباط بین نمونه های دو نوع موجودیت E و G را بدست آورد



تجزیه
ترکیب
تخصیص
تعمیم
تجمع
وراثة صفت

محدودیتهاي روش ER که در روش
EER برطرف گردیدند:



تجزیه

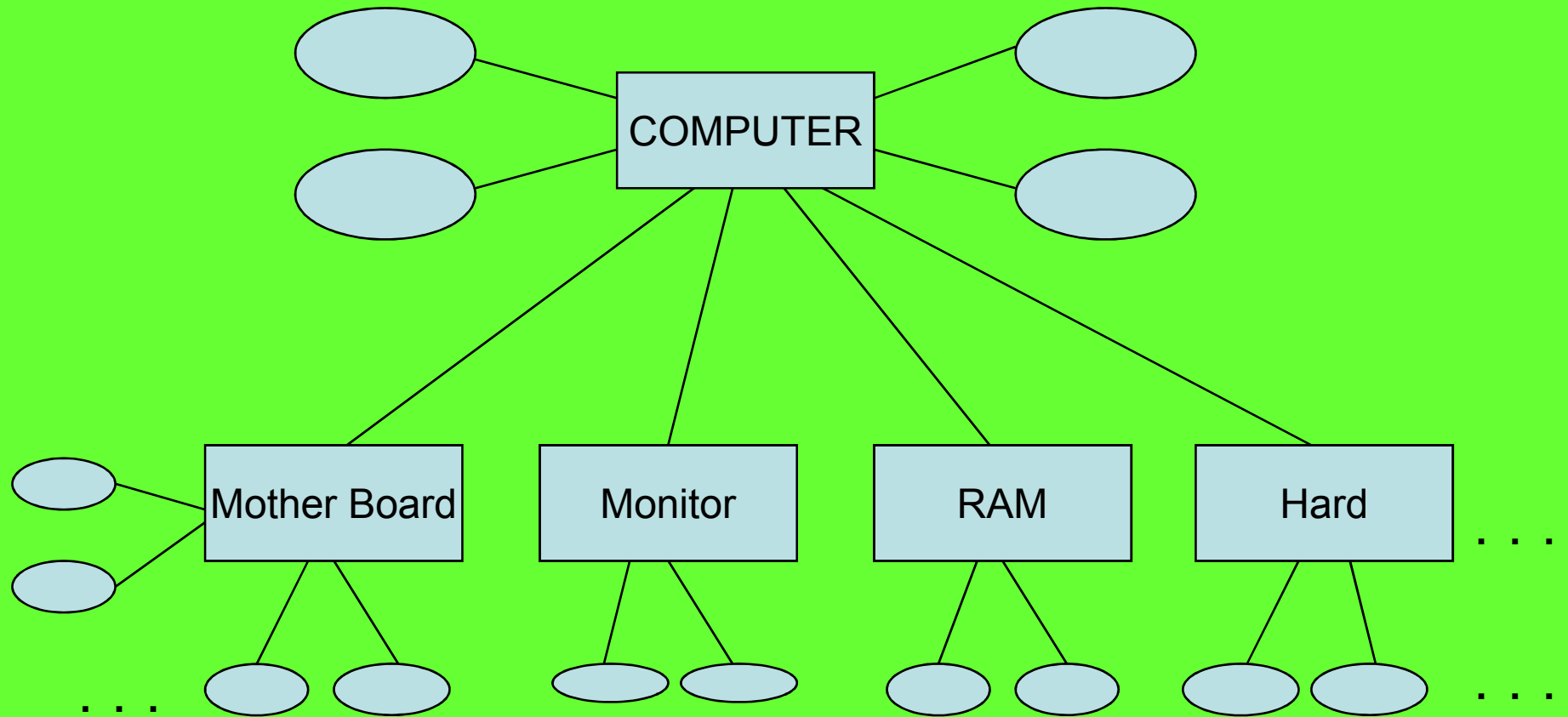
تجزیه یا جداسازی یعنی يك شیء کل را به اجزاء تشکیل دهنده آن تقسیم کنیم. شیء کل صفات ، ساختار و رفتار خود را دارد و هریک از اجزاء نیز صفات ، ساختار و رفتار خاص خود را دارند. شیء کل شامل اجزاء خود است و بین شیء کل و اجزایش ، ارتباط شمول وجود دارد. به این نوع ارتباط در **EER** ، ارتباط “جزئی است از ...” گفته می شود.



ترکیب

ترکیب ، عکس عمل تجزیه است و در این عمل ، با داشتن
 $E_i (i=1, 2, \dots)$ يك نوع موجودیت E را بازشناسي مي كنيم
به نحوي كه E_i ها اجزاء تشكيل دهنده آن باشند

مثال تجزیه و ترکیب





تخصیص

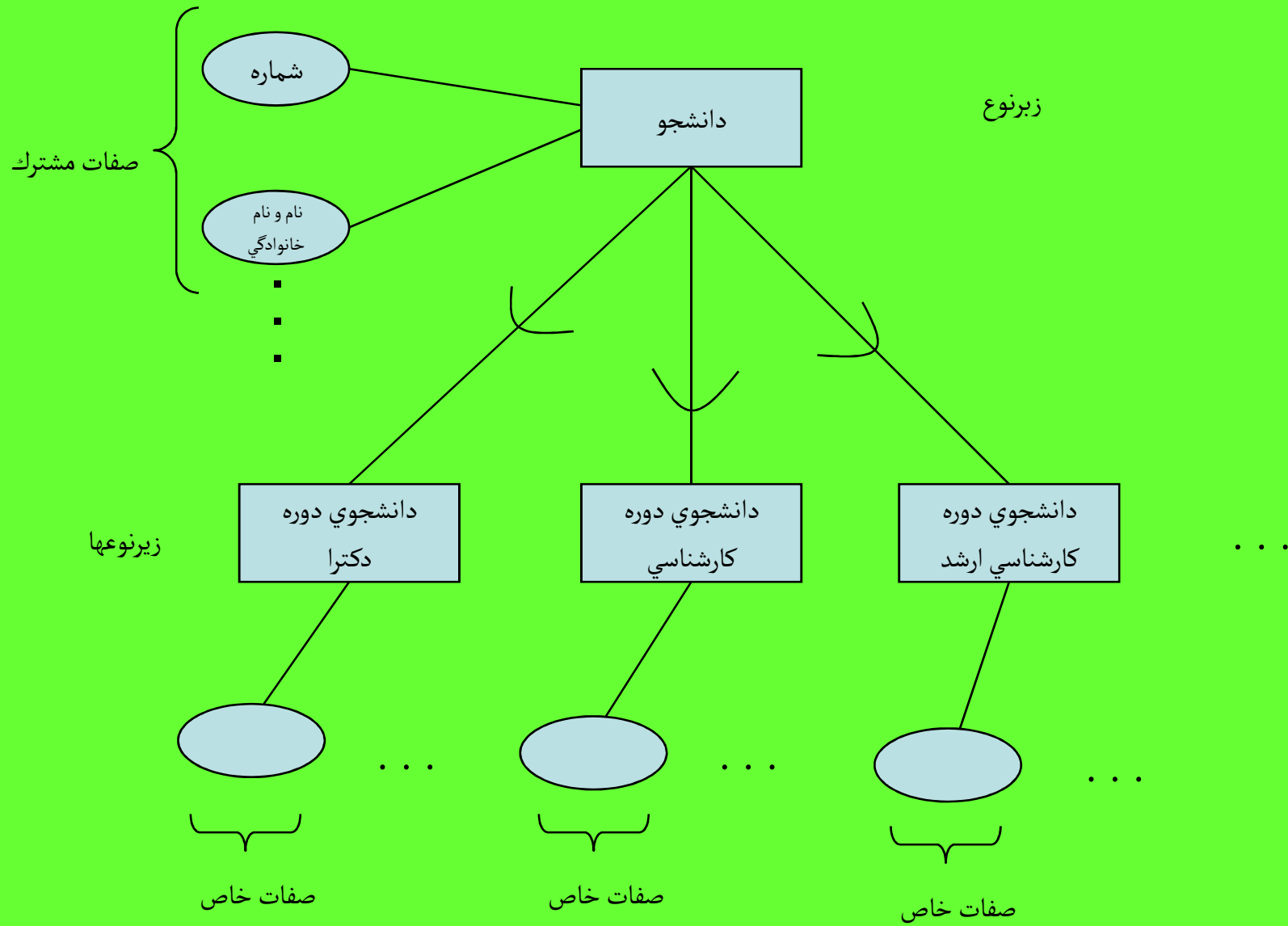
تخصیص عبارتست از مشخص کردن گونه‌های خاص يك شیء براساس يك یا چند ضابطه مشخص ، مثلاً اگر شیء موجود زنده را در نظر بگیریم ، سه گونه خاص آن عبارتند از: انسان ، حیوان و نبات. در روش **EER** هر يك نوع موجودیت می‌تواند خود زیرنوع موجودیتهایی داشته باشد. بین هر زیرنوع و زیرنوع ارتباط “گونه‌ای است از ...” وجود دارد.



تعميم

تعميم ، عكس عمل تخصيص است ، به اين معنا كه با داشتن
زيرنوعهاي خاص ، صفات مشترك بين آنها را در يك مجموعه
صفات براي يك زيرنوع موجوديت در نظر مي گيريم

مثال تخصیص و تعمیم

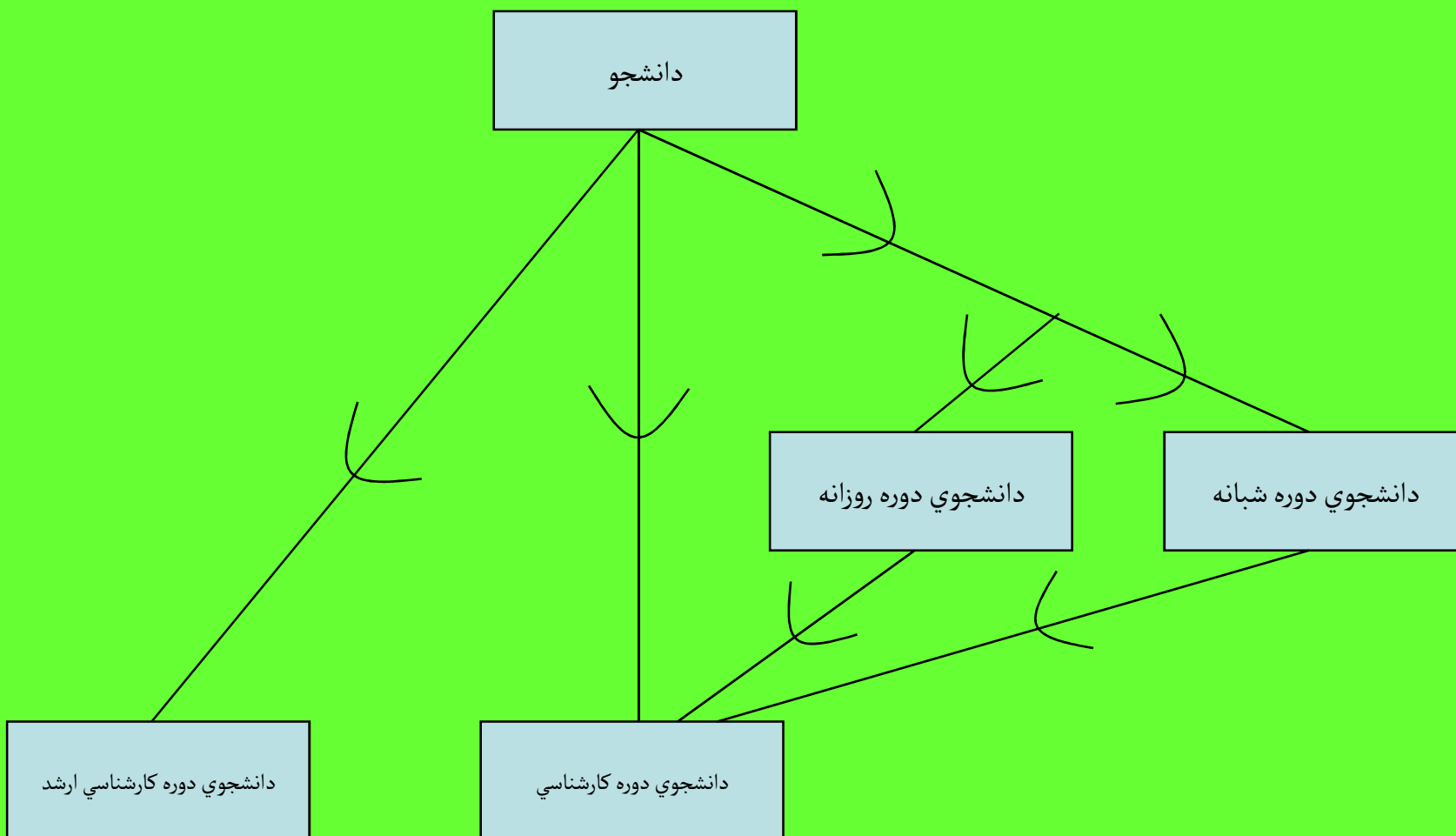




وراثت چندگانه

يك زیرنوع موجودیت ، مي تواند در عين حال زیرنوع يك
زیرنوع موجودیت ديگر هم باشد. با اين ترتيب مي توان
مفهوم وراثت چندگانه را در روش **EER** نمايش داد.

مثال وراثت چندگانه

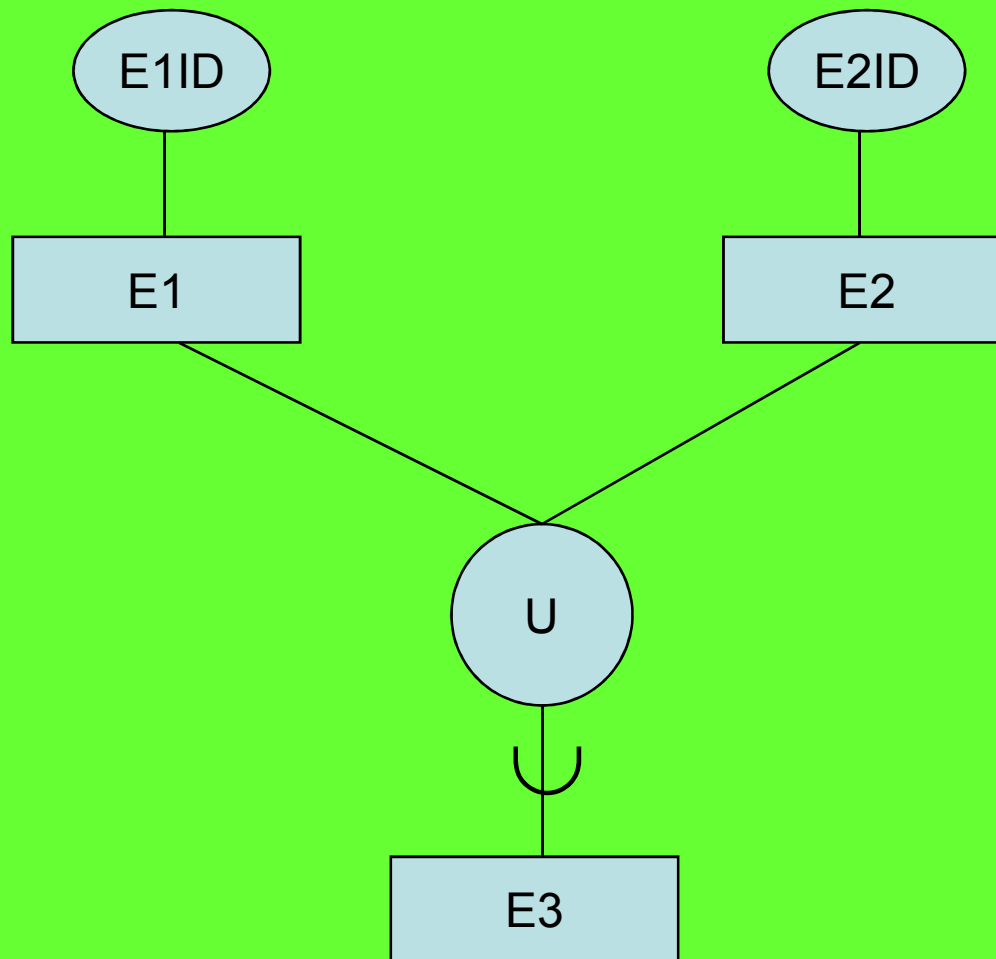




دسته‌بندی

يك زیرنوع می‌تواند زیرنوع بیش از يك زیرنوع باشد. ممکن است زیرنوع‌های این زیرنوع ، از يك نوع نباشند. به این زیرنوع اصطلاحاً دسته (طبقه) گویند. برای نمایش دسته ، از نماد **U** استفاده می‌شود.

دسته بندی



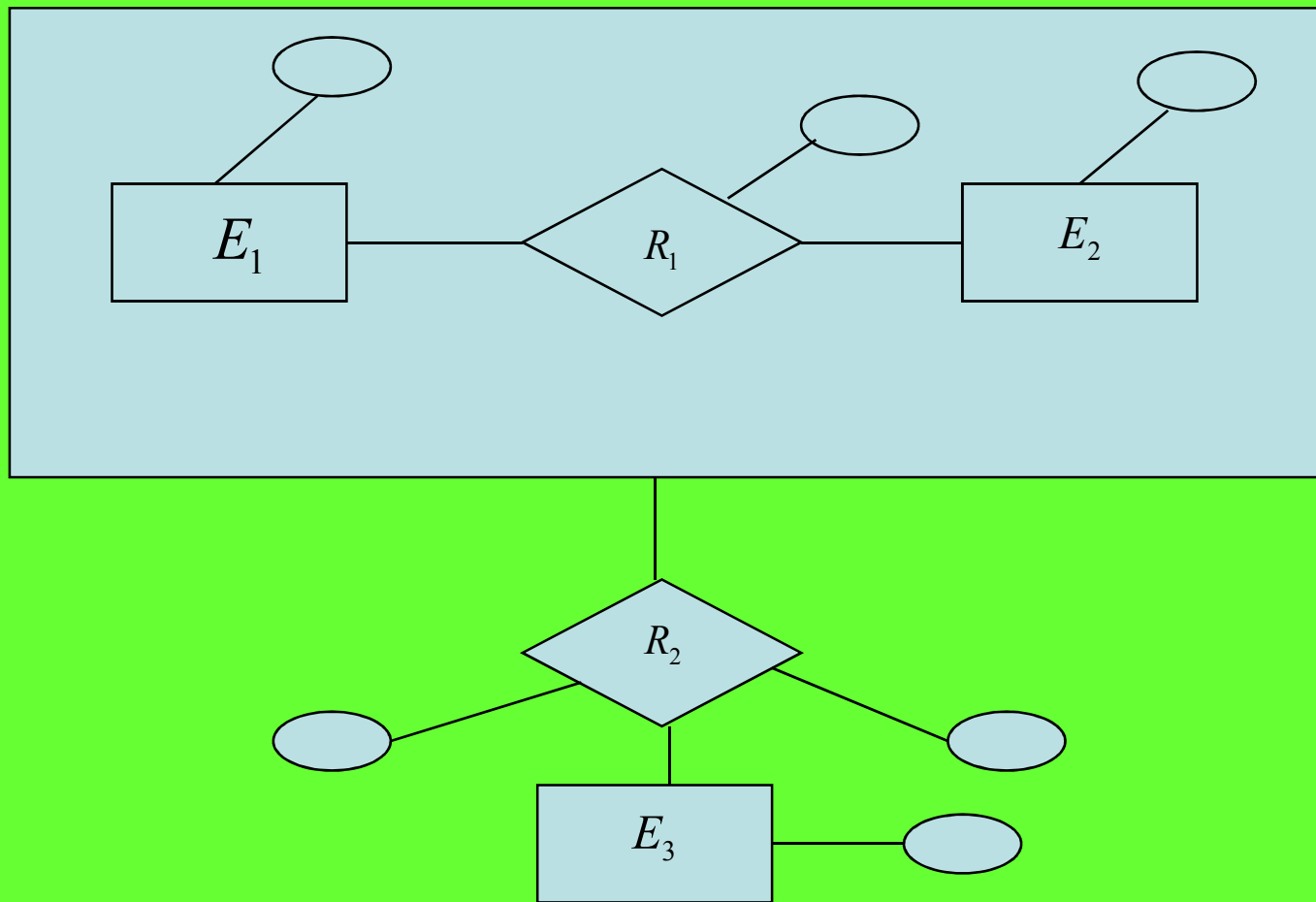


تجمع

تجمع عبارتست از ساختن يك نوع موجودیت جدید با دیدن دو یا بیش از دو نوع موجودیت ، که خود باهم در يك ارتباط شرکت دارند ، به صورت يك نوع موجودیت واحد. در واقع مجموعه‌ای از موجودیتهای مرتبط را باهم مجتمع کرده و به عنوان يك نوع موجودیت واحد ، در نظر می‌گیریم و این نوع موجودیت واحد خود می‌تواند با نوع موجودیت دیگری ارتباط داشته باشد.

نمایش تجمع

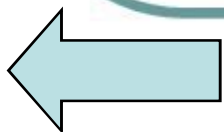
E_k





مراحل مدلسازی معنایی داده‌ها

- 1- مطالعه ، تحلیل و شناخت محیط
- 2- برآورد خواسته‌ها و نیازهای اطلاعاتی و پردازشی همه کاربران و تشخیص محدودیتهای معنایی
- 3- بازشناسی انواع موجودیتهای مطرح و تعیین وضع هر يك
- 4- تعیین مجموعه صفات هر نوع موجودیت



ادامه



مراحل مدلسازی معنایی داده‌ها

5- بازشناسی انواع ارتباطات بین انواع موجودیتهای ، تشخیص نوع مشارکت و چندی ارتباط

6- رسم نمودار ER

7- فهرست کردن پرسشهایی که پاسخ آنها از نمودار ER بدست می آید.

8- واری مدلسازی انجام شده تا اطمینان حاصل شود که مدلسازی پاسخگوی نیاز کاربران است.



روش مدل سازی UML

در این روش از چند نمودار برای نمایش مدل سازی و طراحی نرم افزار استفاده می شود

مفاهیم اصلی در این مدل سازی:

- رده
- صفت
- بستگی



تناظر بين مفاهيم UML و مفاهيم EER	
مفهوم در UML	مفهوم در EER
رده	نوع موجودیت
شیء	نمونه موجودیت
صفت	صفت
بستگی	ارتباط
پیوند	نمونه ارتباط
بستگی انعکاسی	ارتباط بازگشتی
بستگی مقید	نوع موجودیت ضعیف
میدان ساختمان	صفت مرکب
صفت پیوند	صفت ارتباط
چندی بستگی	چندی ارتباط



نمادها

- رده با يك مربع يا مستطيل نشان داده مي شود و به شكل زير به سه قسمت تقسيم مي شود:

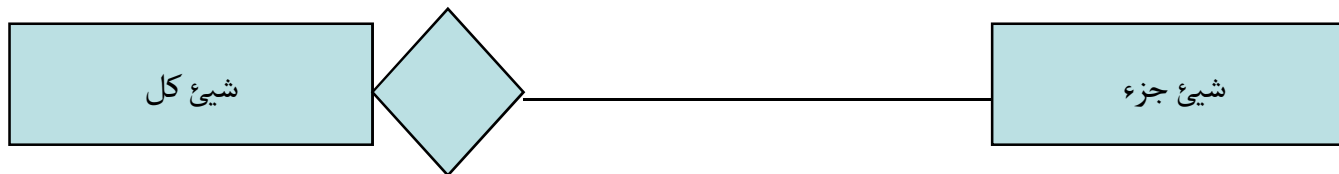
نام رده
نام صفات
پردازشها



نمادها

- بستگی بین دو رده ، به صورت يك خط متصل کننده دو رده نمایش داده می شود و نام ارتباط روی خط نوشته می شود.

- تجمع به صورت زیر نشان داده می شود:





- چندي بستگي به صورت $\min \dots \max$ نوشته مي شود. اگر به جاي $\max$ علامت ستاره باشد ، به اين معنا است كه مقدار $\max$ محدوديت ندارد.

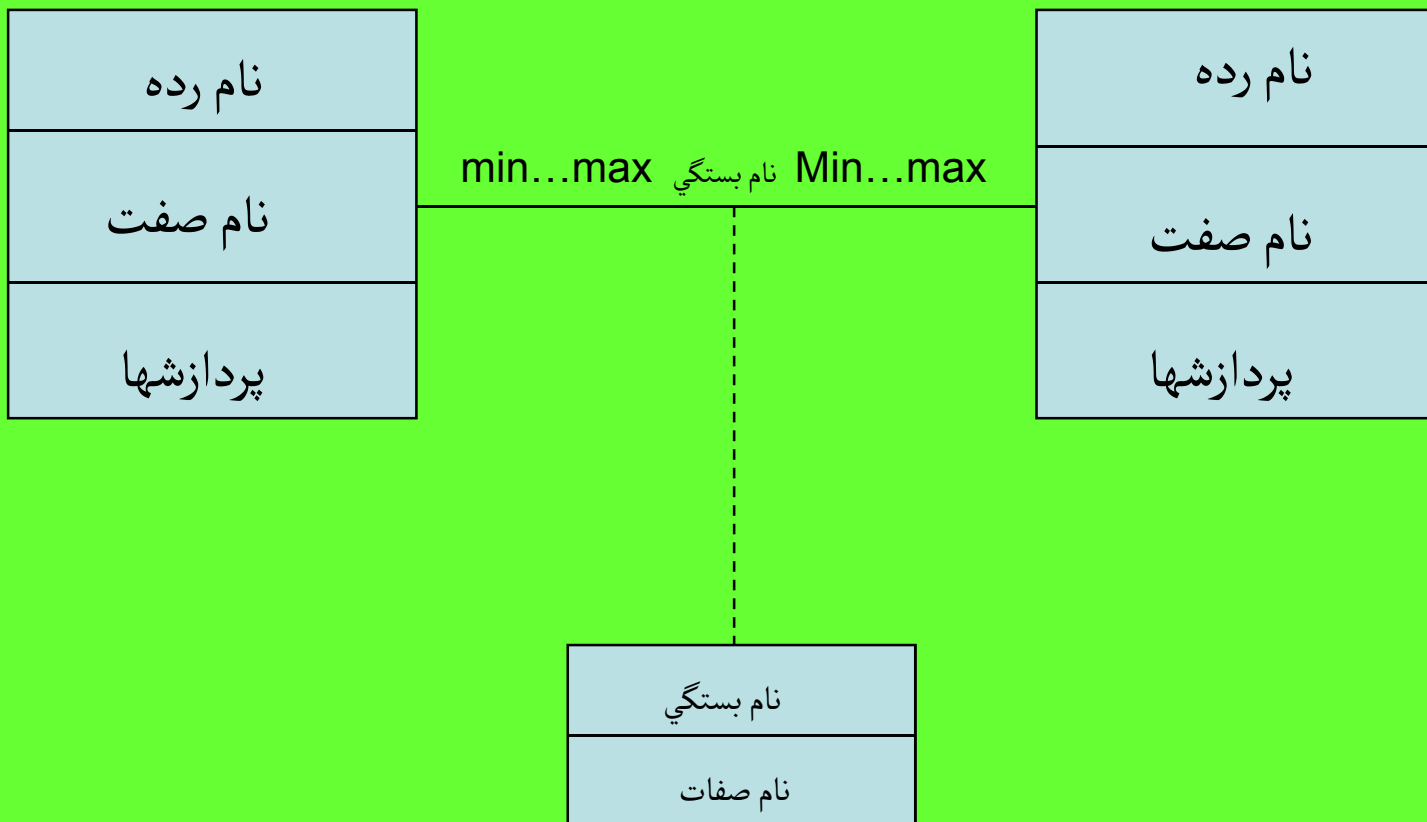
- صفت چندمقداري به صورت يك رده جداگانه نشان داده مي شود ، ولي فاقد قسمت مربوط به پردازش است.



- نام میدان بعد از نام صفت نوشته می شود و بین این دو نام علامت : گذاشته می شود.

- صفت پیوندی در يك مربع (مستطیل) نوشته می شود و این مربع با خط چین به خط نشان دهنده بستگی متصل می شود. نام پیوند و نام صفات پیوند در دو قسمت از این مربع گذاشته می شود.

نمایش صفت پیوند در UML





خصوصیات کلی روش مدلسازی معنایی داده‌ها

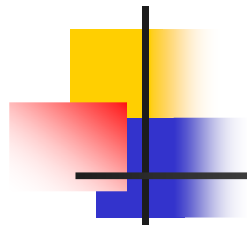
- گویایی
- صوری بودن
- سادگی مفاهیم
- قابلیت نمایش نموداری
- ایجاز
- جامع بودن مفاهیم
- قابلیت نمایش ساختار ، حالت و رفتار نوع موجودیت
- گسترش پذیری

An abstract background featuring large, textured blocks of color: a large green block on the left, a large orange block on the bottom right, and a grey block on the right. The Persian text is overlaid on the green block.

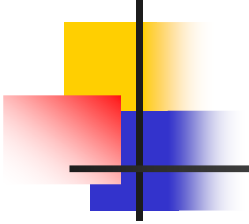
جلسه چهارم

پایگاه داده در محیط انتزاعي

آنچه در این جلسه می خوانید:



- 1- پایگاه داده‌ها در محیط انتزاعی
- 2- سطوح محیط انتزاعی
- 3- گونه‌های موجود ساختار داده‌ای
- 4- مفهوم ساختار داده‌ای در سطوح مختلف پایگاه داده‌ها
- 5- ساختار داده‌ای رابطه‌ای
- 6- شمای پایگاه جدولی
- 7- عملیات در پایگاه جدولی



آنچه در این جلسه می خوانید:

8- عملگرهای جبر رابطه‌ای

9- برخی ویژگیهای ساختار داده‌ای جدولی

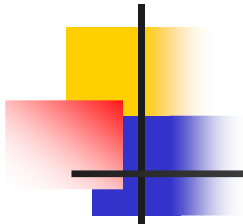
10- ساختار داده‌ای سلسله‌مراتبی

11- برخی ویژگیهای ساختار داده‌ای سلسله‌مراتبی

12- ساختار داده‌ای شبکه‌ای

13- مجموعه کوداسیل

14- برخی ویژگیهای ساختار داده‌ای شبکه‌ای



هدفهاي كلي: آشنائي با پايگاه داده در محيط انتزاعي

هدفهاي رفتاري: دانشجو در پايان اين جلسه مي تواند:

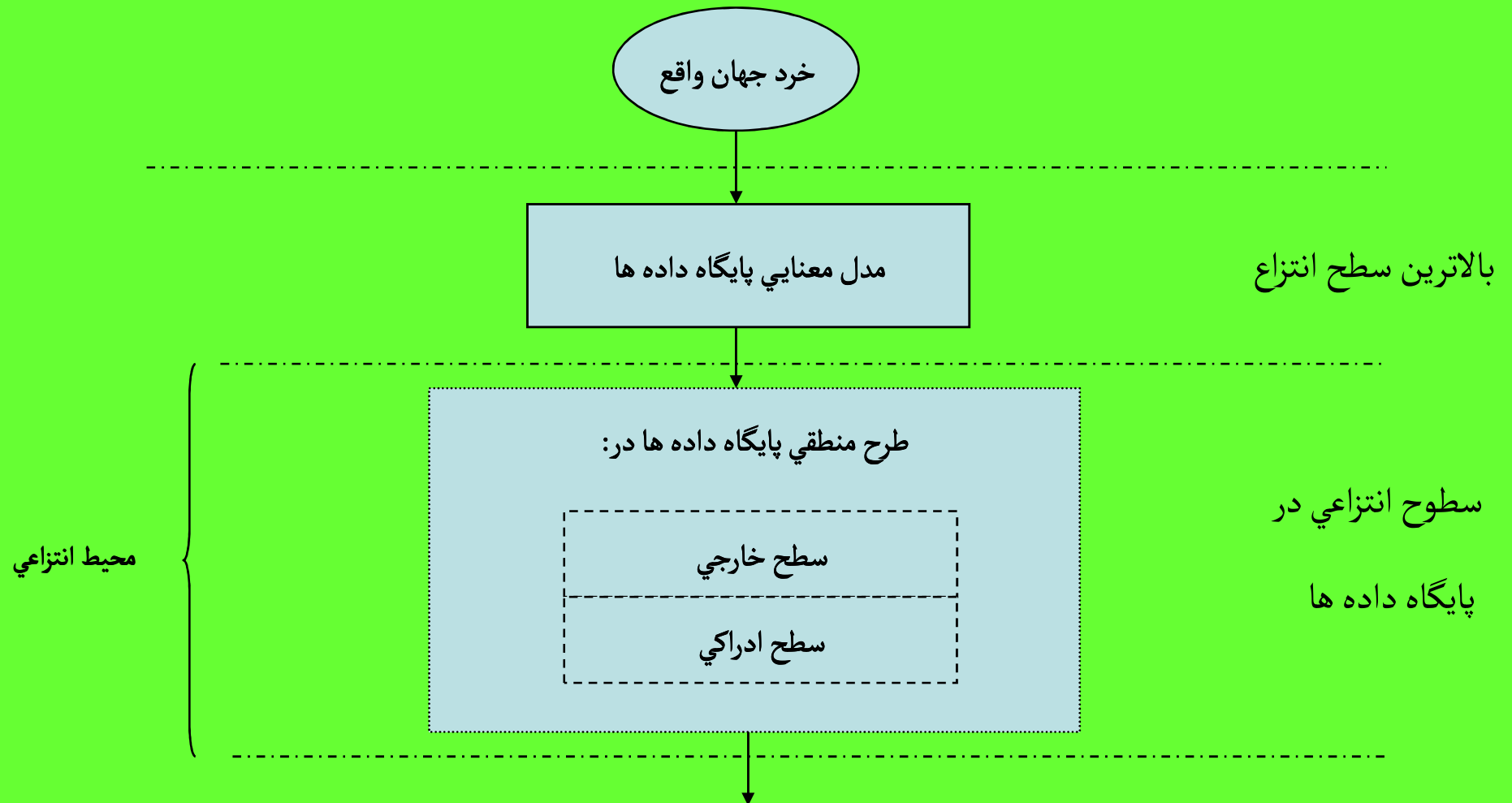
- پايگاه داده در محيط انتزاعي و سطوح اين محيط را بيان کند.
- ساختار داده‌اي رابطه‌اي و ويژگيهاي آن را بيان کند.
- ساختار داده‌اي شبکه‌اي و ويژگيهاي آن را بيان کند.
- ساختار داده‌اي سلسله‌مراتبی و ويژگيهاي آن را بيان کند.



پایگاه داده‌ها در محیط انتزاعي

در مقوله پایگاه داده‌ها وقتی از محیط انتزاعي سخن می‌گوییم ، منظور محیطی است فراتر از محیط فایلینگ منطقی و فایلینگ فیزیکی. مفاهیمی که در این محیط مطرح می‌شوند باید از جنبه‌های فایلینگ پایگاه مستقل و ماهیتا انتزاعي باشند.

سطوح محیط انتزاعي





ساختار داده‌ای امکانی است برای نشان دادن داده‌های در مورد انواع
موجودیت‌ها و انواع ارتباطات بین آنها

هر ساختار داده‌ای حداقل یک عنصر ساختاری اساسی دارد.

پایگاه داده‌ها در محیط انتزاعی ، گردایه‌ای است از نمونه‌های
متمايز عنصر (عناصر) ساختاری اساسی یک ساختار داده‌ای
مشخص.

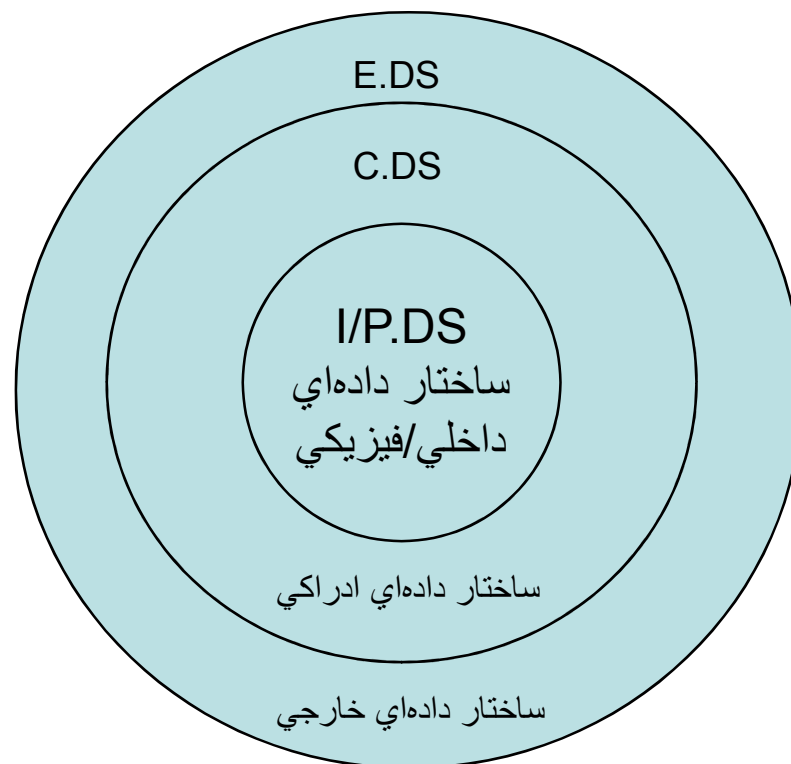


گونه‌های موجود ساختار داده‌ای

- ساختار داده‌ای رابطه‌ای
- ساختار داده‌ای سلسله‌مراتبی
- ساختار داده‌ای شبکه‌ای
- ساختار داده‌ای جعبه‌ای
- ساختار داده‌ای هایپرگرافی
- ساختار داده‌ای لیستهای وارون



مفهوم ساختار داده‌ای در سطوح مختلف پایگاه داده‌ها





ساختار داده‌ای رابطه‌ای

رابطه مفهومی ریاضی است. اما از دید کاربر، رابطه نمایشی جدولی دارد.

مفاهیم ساختار جدولی عبارتند از:

جدول	سطر	ستون
------	-----	------

عنصر ساختاری اساسی در این ساختار داده‌ای، جدول است.



شمای پایگاه جدولی

شمای پایگاه داده‌ها عبارتست از تعریف (توصیف) ساختهای انتزاعی طراحی شده و نوعی برنامه است شامل دستورات تعریف داده‌ها و کنترل داده‌ها، و دستورات عملیات در داده‌ها در آن وجود ندارد.



عملیات در پایگاه جدولی

درج (INSERT)

حذف (DELETE)

بهنگام سازی (UPDATE)

بازیابی (SELECT)



مثال درج يك نمونه دانشجو

```
INSERT  
INTO STT  
VALUES('7413673','AHMADI','bs','Comp','D222')
```

مثال حذف يك نمونه درس

```
DELETE  
FROM COT  
WHERE COID='COM777'
```



مثال حذف چند سطر

```
DELETE  
FROM STT  
WHERE STDEG='ms'
```

با این دستور مشخصات دانشجویان دوره کارشناسی ارشد از جدول STT حذف می شود

مثال بهنگام سازی تک سطر

```
UPDATE COT  
SET COID='Com303'  
WHERE COID='Com202';
```

با این دستور شماره درس Com202 عوض می شود.



مثال بهنگام سازي چند سطر

```
UPDATE COT  
SET CREDIT='1'  
WHERE COTYPE='Lab';
```

با این دستور ، تعداد واحد تمام درسهایی آزمایشگاهی ، یک می شود.

مثال بازیابی

```
SELECT DOID  
FROM STCOT  
WHERE STID='76140444'
```

با این دستور شماره درسهایی یک دانشجو بازیابی می شود.



عملگرهای جبر رابطه‌ای که برای بازیابی از محیط جدولی بکار می‌روند:

1- عملگر گزینش (تحدید) **RESTRICT**: زیرمجموعه‌ای افقی (تعدادی سطر) را بازیابی می‌کند.

2- عملگر پرتو **PROJECT**: زیرمجموعه‌ای عمودی (تعدادی ستون) را بازیابی می‌کند.

3- عملگر پیوند **JOIN**: دو جدول را بهم پیوند می‌زند، یعنی سطرهایی که شرایط مورد نظر را داشته باشند، با یکدیگر پیوند می‌شوند.



برخي ویژگیهای ساختار داده‌ای جدولی

- 1- از نظر کاربر نمایش ساده‌ای دارد.
- 2- محیطش مسطح است.
- 3- عنصر ساختاری اساسی آن جدول است.
- 4- همه چیز با فقره داده‌های ساده نمایش داده می‌شود.
- 5- ارتباطات با چندیهایی مختلف در آن قابل نمایش است.
- 6- منطق بازیابی آن ساده است.
- 7- ساختار منطقی دستور بازیابی آن ساده است.
- 8- برای پرسشهای قرینه ، رویه پاسخگویی قرینه دارد.
- 9- مبنای تئوریک قوی دارد.



ساختار داده‌ای سلسله‌مراتبی

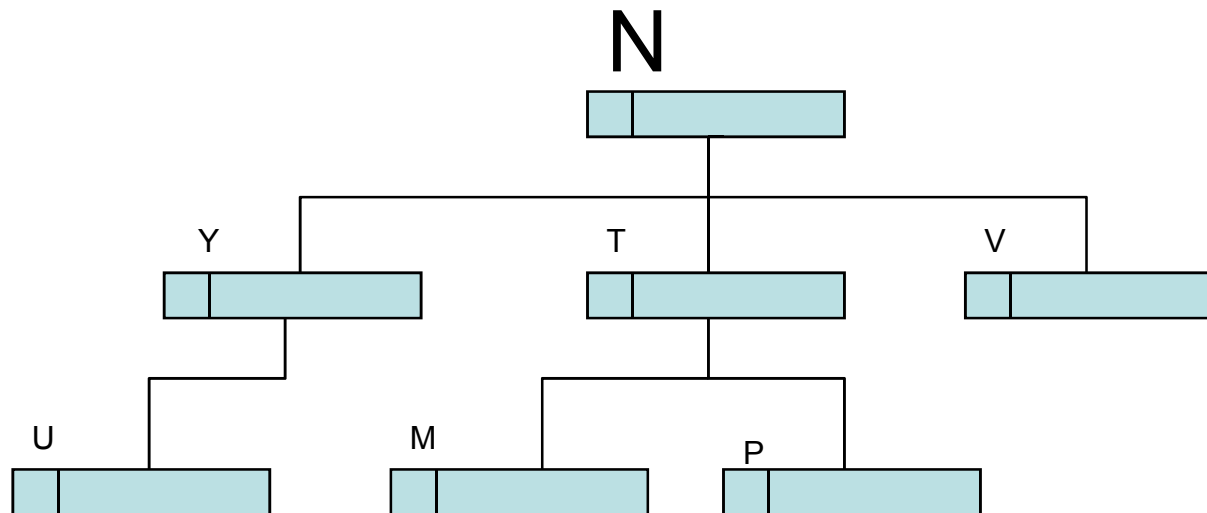
عناصر ساختاری اساسی

2- نوع پیوند پدر- فرزند

1- نوع رکورد

نوع رکورد برای نمایش نوع موجودیت به کار می‌رود.

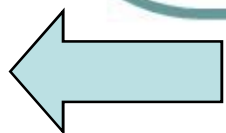
ارتباط بین دو نوع موجودیت به وسیله پیوند پدر-فرزند نمایش داده می‌شود.





برخي ویژگیهای ساختار داده‌ای سلسله‌مراتبی

- 1- سادگی نمایش ساختار جدولی را ندارد.
- 2- مبنای ریاضی ندارد.
- 3- دو عنصر ساختاری اساسی دارد.
- 4- ارتباط یک به چند را نمایش می‌دهد.
- 5- نمایش ارتباط چند به چند در آن دشوار است.
- 6- ساخت منطقی رویه بازیابی آن به سادگی منطق رویه بازیابی در ساختار جدولی نیست.



ادامه



برخی ویژگیهای ساختار داده‌ای سلسله‌مراتبی

- 7- در عملیات ذخیره‌سازی مشکلاتی دارد.
- 8- تقارن ساختار جدولی را ندارد.
- 9- تعدادی قاعده جامعیت ذاتی دارد.
- 10- در مواقعی که در ذخیره‌سازی نمونه‌های فرزند، افزونگی پدید آید، پایگاه در معرض ناسازگاری قرار می‌گیرد.



ساختار داده‌ای شبکه‌ای

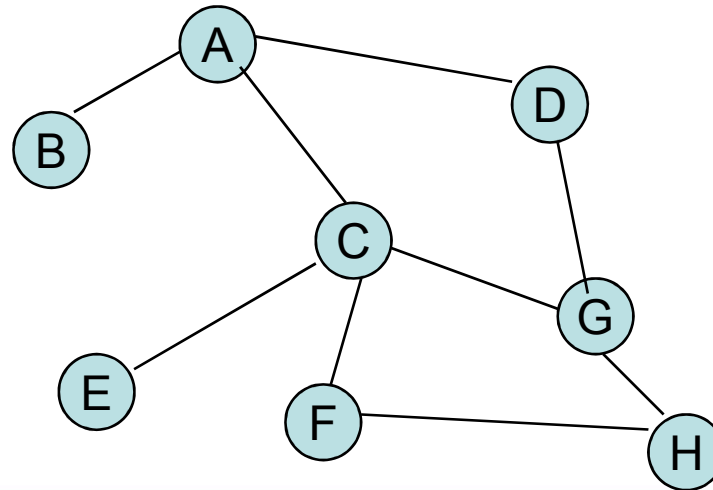
عناصر ساختاری اساسی

2- نوع مجموعه

1- نوع رکورد

نوع رکورد برای نمایش نوع موجودیت به کار می‌رود.

نوع مجموعه برای نمایش ارتباط $1:N$ بین دو (چند) نوع موجودیت به کار می‌رود.





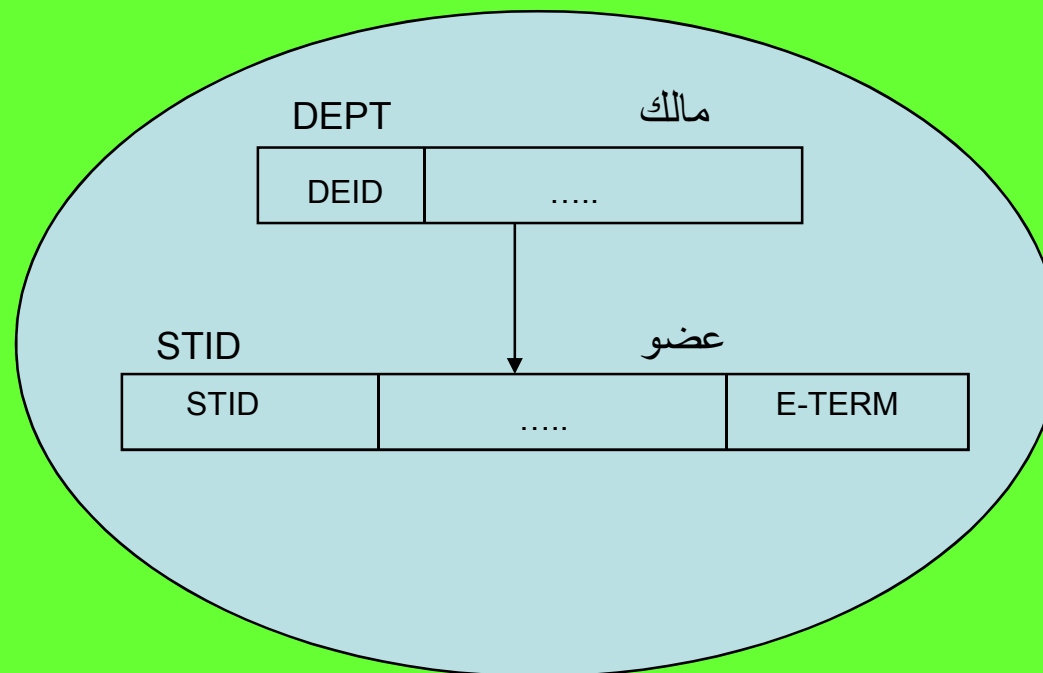
نوع مجموعه (مجموعه کوداسیل) از سه جزء تشکیل شده است:

1- نام مجموعه

2- يك نوع رکورد مالک

3- يك نوع رکورد عضو

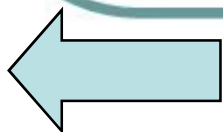
يك نوع مجموعه كوداسيل





برخی ویژگیهای ساختار داده‌ای شبکه‌ای

- 1- سادگی ظاهری ساختار داده‌ای جدولی را ندارد.
- 2- مبنای ریاضی ندارد.
- 3- دو عنصر ساختاری اساسی دارد.
- 4- ماهیتا خاص نمایش ارتباطات “یک به چند” نیست.
- 5- ساخت منطقی دستور بازیابی آن پیچیده‌تر از ساختارهای دیگر است.



ادامه



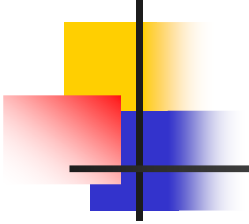
برخي ویژگیهای ساختار داده‌ای شبکه‌ای

- 6- مثل ساختار داده‌ای جدولی تقارن دارد.
- 7- خطر بروز ناسازگاری داده‌ها نسبت به ساختار سلسله‌مراتبی، کمتر است.
- 8- قواعد جامعیت ذاتی دارد.
- 9- به علت حجم زیاد اشاره‌گرها، ایجاد یا اصلاح آنها می‌تواند سبب بروز فزونکاری در سیستم شود.
- 10- بعضی آنومالیهای مدل سلسله‌مراتبی در عملیات ذخیره‌سازی را ندارد.

The background is an abstract composition. A large, textured green square occupies the upper left portion. To its right, a yellow square is partially visible, with a dark, textured rectangular shape overlapping its top edge. The rest of the background is a light, neutral tone with some subtle texture and a small, bright white triangular shape on the right side.

جلسه پنجم

معماري پایگاه داده‌ها



آنچه در این جلسه می خوانید:

1- معماری پیشنهادی **ANSI**

2- دید ادراکی (مفهومی)

3- دید خارجی

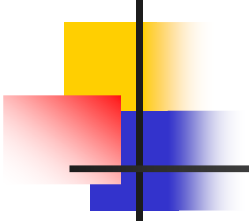
4- دید داخلی

5- سطوح معماری در محیط ناپایگاهی و محیط پایگاهی و نقش **DBMS**

در ایجاد ، مدیریت و پردازش فایلها

6- کاربر

7- زبان میزبان



آنچه در این جلسه می خوانید:

8- نقش افزایش تعداد زبانهای میزبان مورد پذیرش DBMS

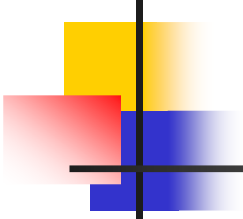
9- زبان داده‌ای فرعی

10- دستورهای DSL برای سه سطح معماری پایگاه داده‌ها

11- تقسیم‌بندی زبان داده‌ای فرعی از نظر نیاز به زبان میزبان

12- روند کلی مرحله کامپایل برنامه دوزبانی

13- ویژگیهای زبان داده‌ای فرعی



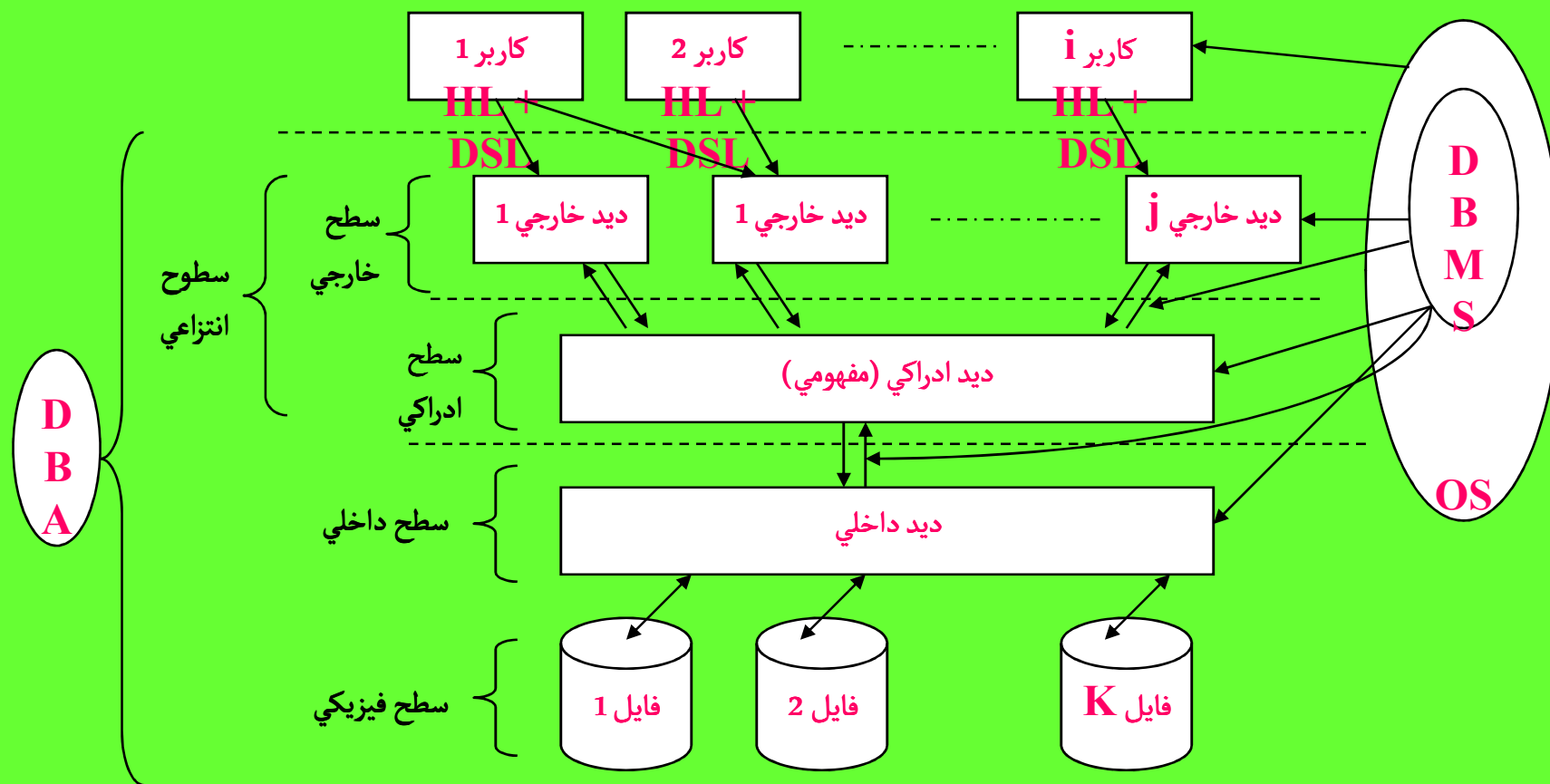
هدفهاي كلي: آشنائي با معماري پاىگاه داده‌ها

هدفهاي رفتاري: دانشجو در پايان اين جلسه مي‌تواند:

- معماري پاىگاه داده‌ها را تشریح کند.
- ديدهاي ادراكي ، خارجي و داخلي را توضیح دهد.
- سطوح معماري در محيط ناپاىگاهي و پاىگاهي را مقایسه کند.
- کاربر ، زبان ميزبان ، زبان داده‌اي فرعي و ويژگيهاي آن را بيان نمايد.

معماري پایگاه داده‌ها

معماري پیشنهادي ANSI





دید ادراکی (مفهومی)

- 1- دید طراح پایگاه داده‌ها نسبت به داده‌های ذخیره‌شده است.
- 2- جامع است.
- 3- در یک محیط انتزاعی مطرح است.
- 4- با عناصر ساختاری اساسی همان ساختار داده‌ای طراحی می‌شود.
- 5- شمایی ادراکی نوعی برنامه است حاوی دستورات تعریف و کنترل داده‌ها. سطح ادراکی در واقع همین شمایی ادراکی است.
- 6- شمایی ادراکی به سیستم داده می‌شود و در کاتالوگ سیستم نگهداری می‌شود.



دید خارجی

- 1- دید کاربر خاص نسبت به داده‌های ذخیره‌شده در پایگاه داده است.
- 2- جزئی است.
- 3- در سطح انتزاعی مطرح است.
- 4- روی دید ادراکی طراحی و تعریف می‌شود.
- 5- شمایی خارجی نوعی برنامه است حاوی دستورات تعریف و کنترل داده‌ها در سطح خارجی که توسط کاربر این سطح نوشته می‌شود.



ادامه



دید خارجی

- 6- به تعریف مجموعه دیدهای خارجی کاربر ، سطح خارجی گفته می شود.
- 7- هر کاربر می تواند تعدادی دید داشته باشد.
- 8- چند کاربر می توانند در یک دید مشترک باشند.

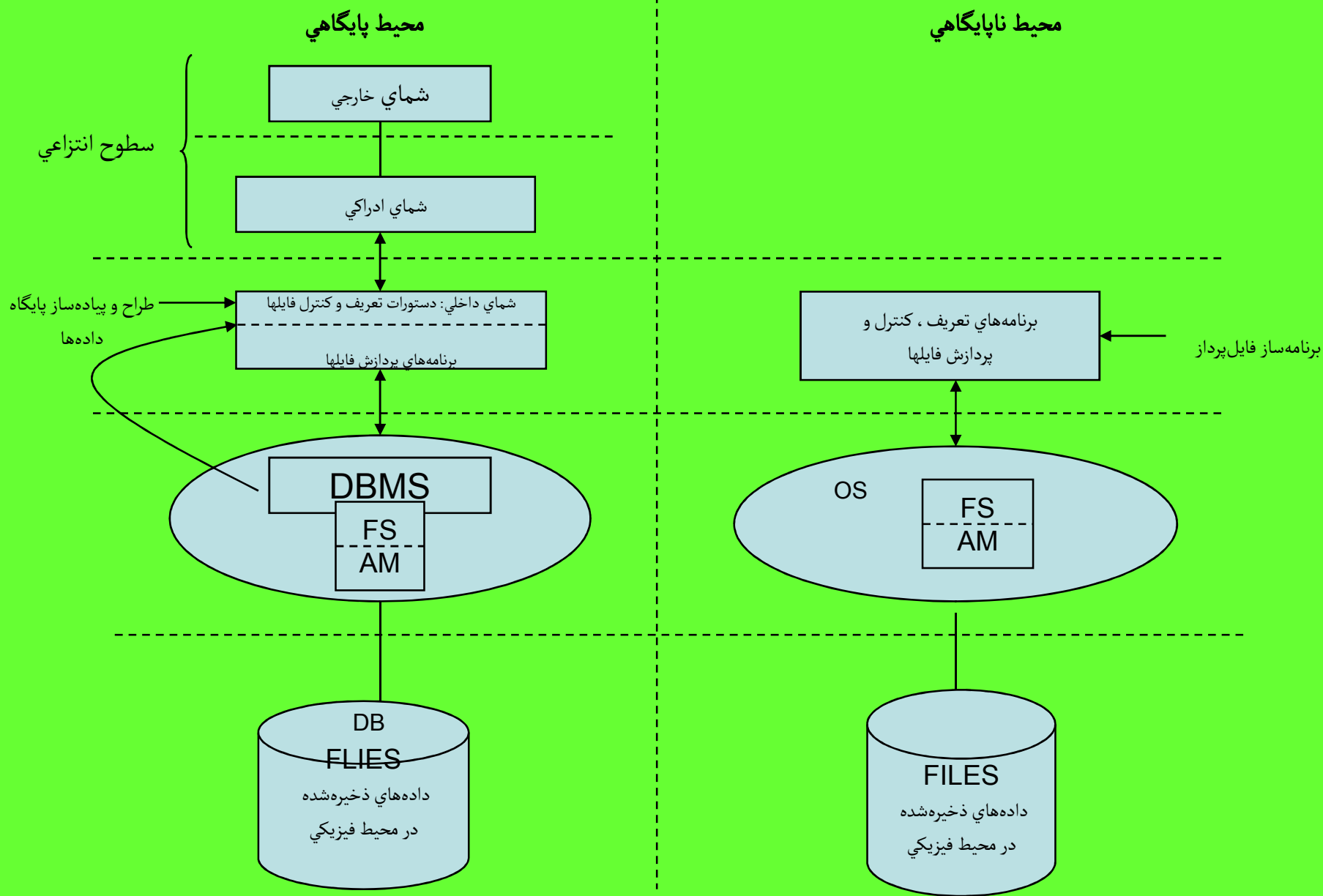


دید داخلی

- 1- دید **DBMS** و طراح پایگاه داده‌ها است و در سطحی پایین‌تر از سطح ادراکی ، نسبت به کل داده‌های ذخیره‌شده است.
- 2- در سطح فایلینگ منطقی مطرح است.
- 3- مبتنی بر یک ساختار فایل است که با نظر طراح پایگاه طراحی می‌شود و به طراحی ، طراحی فیزیکی موسوم است.
- 4- در سطح داخلی پایگاه داده‌ها ، فایلینگ منطقی تعریف می‌شود.
- 5- شمایی داخلی نوعی برنامه است که توسط خود **DBMS** تولید می‌شود و شرح فایلینگ منطق پایگاه است.

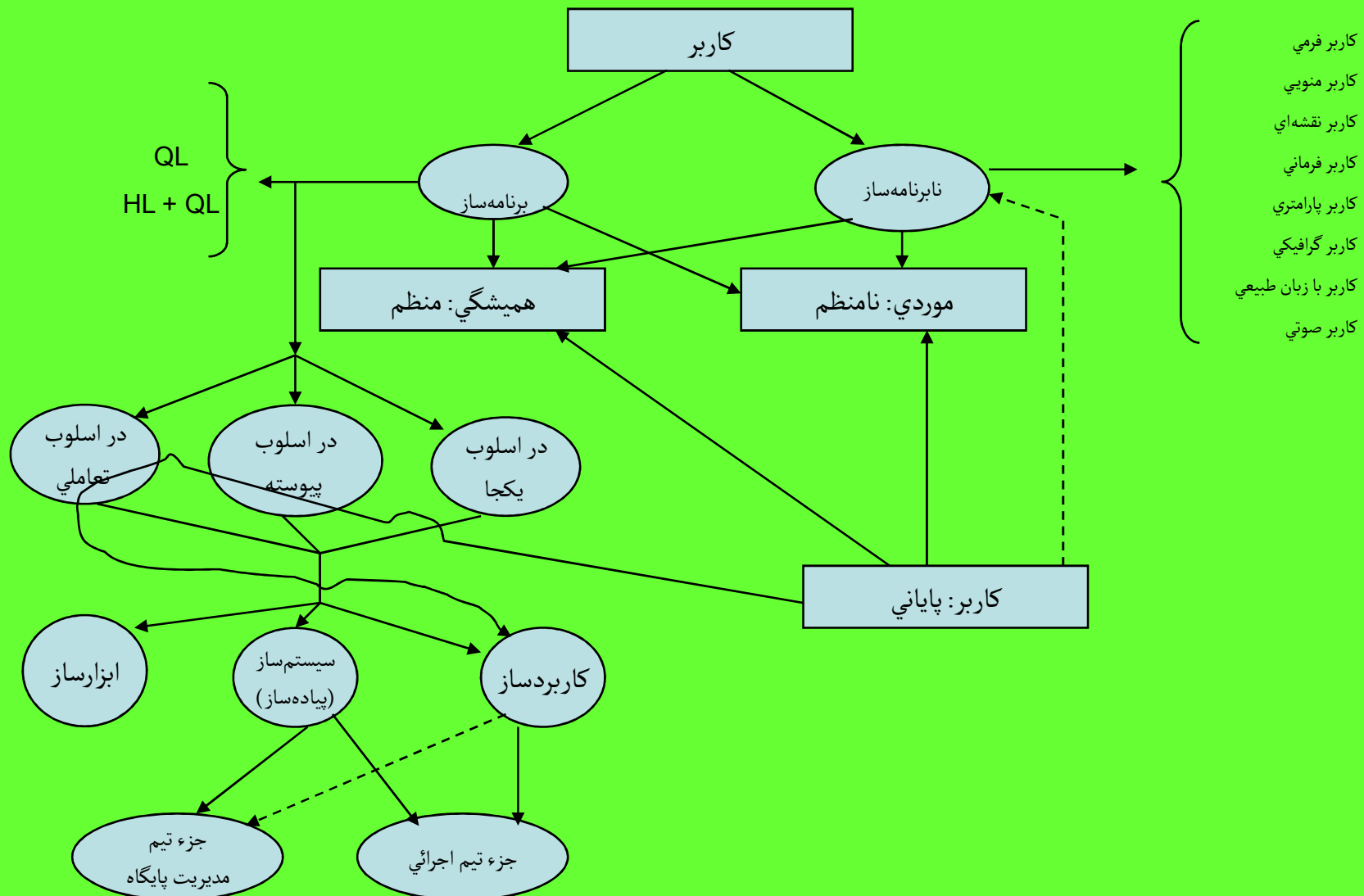
سطوح معماری در محیط ناپایگاهی و محیط پایگاهی و نقش DBMS در ایجاد ، مدیریت و پردازش

فایلها



کاربر

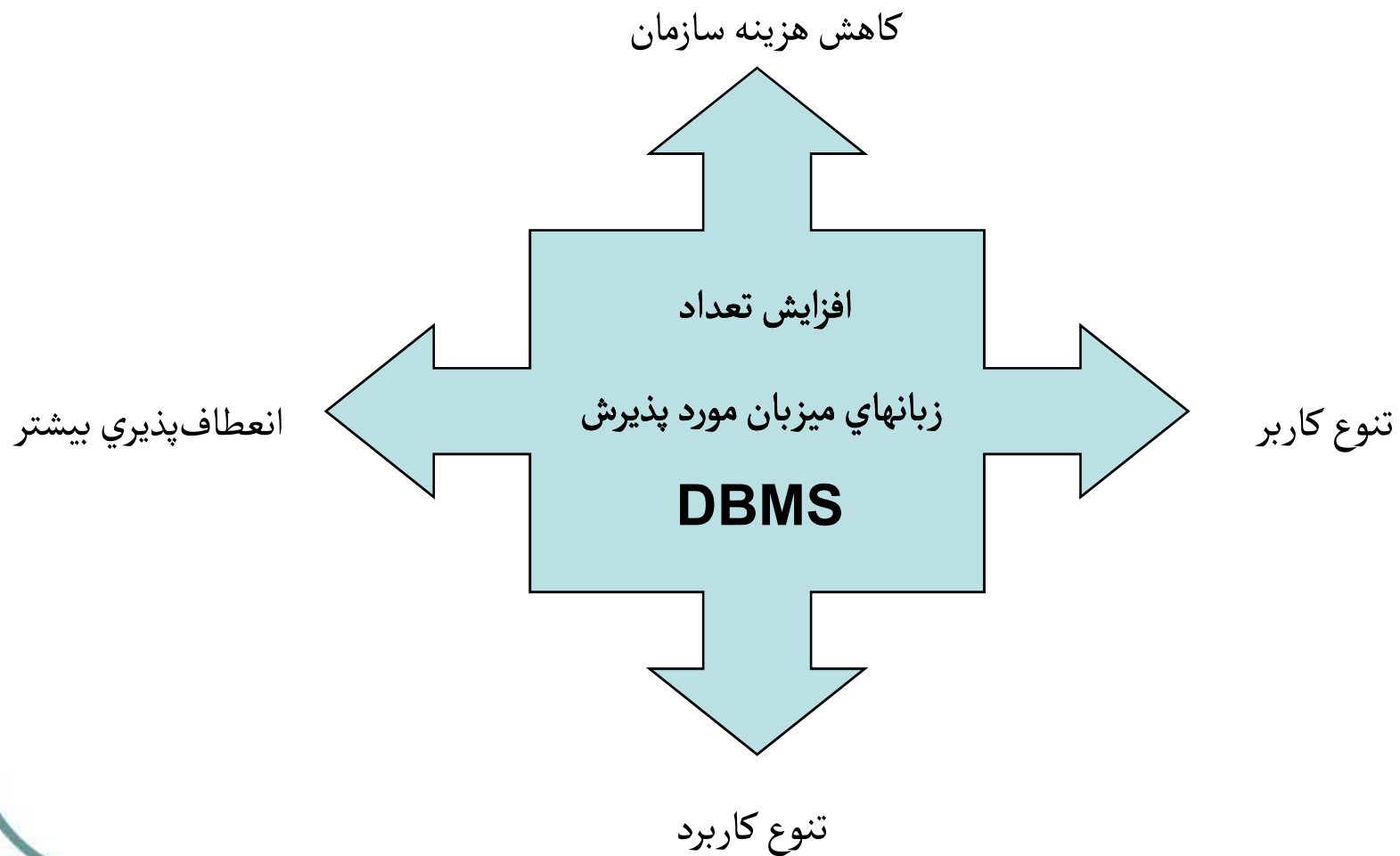
در معنای عام ، هر استفاده کننده از پایگاه داده ها را کاربر گوئیم .





زبان میزبان

یکی از زبانهای برنامه‌سازی متعارف مانند کوبول ، PL1 ،
فرترن ، پاسکال ، C و زبانهای مثل ADA ، LISP ،
JAVA و نیز زبان اسمبلی است.



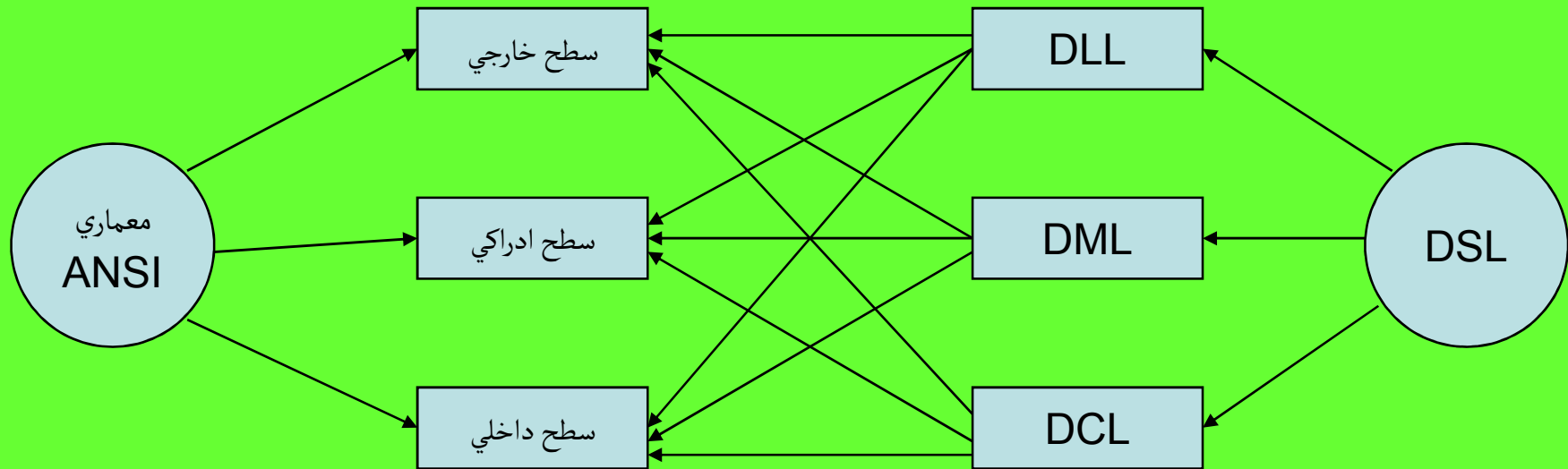


زبان داده‌ای فرعی

شامل سه دسته دستور زیر است:

- 1- دستورات تعریف داده‌ها (DDL) Data Definition Language
- 2- دستورات عملیات روی داده‌ها (DML) Data Manipulation Language
- 3- دستورات کنترل داده‌ها (DCL) Data Control Language

دستورهاي DSL براي سه سطح معماري پايگاه داده‌ها





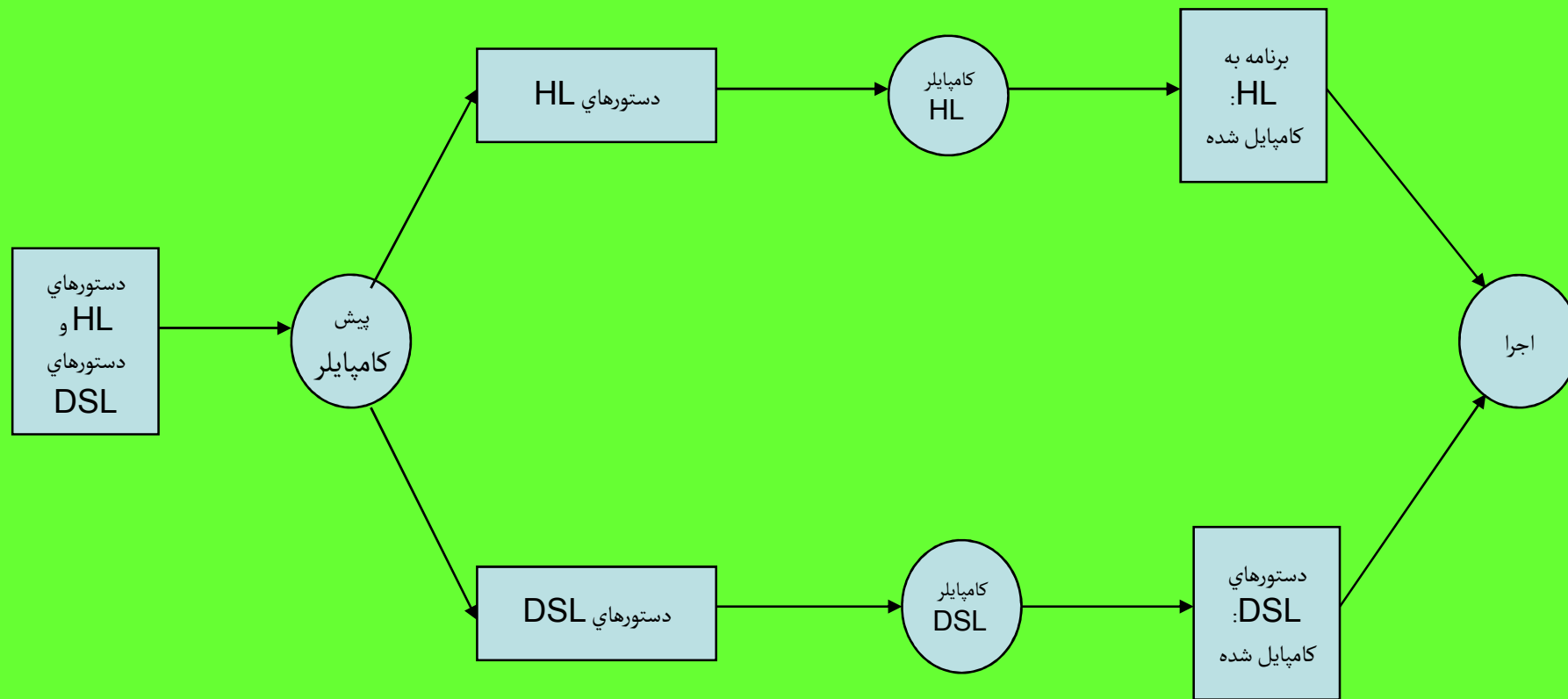
تقسیم‌بندی زبان داده‌ای فرعی از نظر نیاز به زبان میزبان

مستقل (I.DSL): به زبان میزبان نیاز ندارد و به صورت تعاملی استفاده می‌شود. در واقع یک زبان پرس و جو است.

ادغام‌شدنی (E.DSL): دستورهایش در متن برنامه‌ای به زبان میزبان به کار می‌رود و مستقلاً قابل استفاده نیست.

هم مستقل و هم ادغام‌شدنی (I/E.DSL)

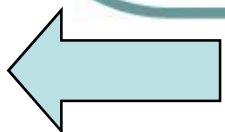
روند کلي مرحله کامپایل برنامه دوزباني





ویژگیهای زبان داده‌ای فرعی

- 1- تعداد دستورهایش باید کم باشد.
- 2- دستورهایش باید شبه زبان طبیعی باشد.
- 3- یادگیری و استفاده آن باید ساده باشد.
- 4- در طراحی آن باید اصل وحدت دستور رعایت شود.
- 5- دستورهایش باید مبتنی بر عناصر ساختاری اساسی ساختار داده‌ای طراحی شوند.



ادامه



ویژگیهای زبان داده‌ای فرعی

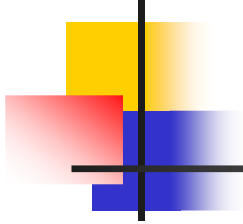
- 6- بهتر است نارویه‌ای (ناروشمند) باشند.
- 7- بهتر است کامپایلری باشد و نه مفسری.
- 8- بهتر است از نظر ساختاری کامل باشد.
- 9- بهتر است از نظر برنامه‌سازی و محاسباتی کامل باشد.
- 10- زبان باید از نظر تعداد دستورهای کنترل داده‌ها و عملکرد هردستور ، غنی و قوی باشد.
- 11- باید از نظر انواع داده‌ای و به ویژه انواع داده‌ای انتزاعی و انواع داده‌ای پیچیده غنی باشد.



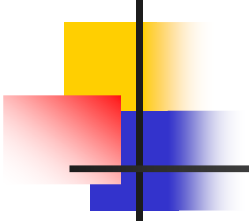
جلسه ششم

سیستم مدیریت پایگاه داده‌ها

آنچه در این جلسه می خوانید:



- 1- سیستم مدیریت پایگاه داده‌ها DBMS
- 2- DBMS به کاربر امکان می‌دهد تا
- 3- رده‌بندی سیستم‌های DBMS از نظرهای مختلف
- 4- اجزای DBMS از نمای بیرونی
- 5- نمای بیرونی (ساده‌شده) DBMS
- 6- اجزای DBMS از نمای درونی
- 7- واحدهای لایه هسته
- 8- واحدهای لایه مدیریت محیط پایگاه داده‌ها
- 9- ساختار یک سیستم پایگاه‌های



هدفهاي كلي: سيستم مديريت پاىگاه داده‌ها

هدفهاي رفتاري: دانشجو در پايان اين جلسه مي‌تواند:

- سيستم مديريت پاىگاه داده‌ها را توصيف کند.
- رده‌بندي **DBMS** را از ديدگاههاي مختلف بيان کند.
- اجزاء **DBMS** از نماي دروني و بيروني را برشمارد.
- واحدهاي لايه هسته و مديريت پاىگاه داده را نام ببرد.



سیستم مدیریت پایگاه داده‌ها

DBMS

تعریف: یکی از انواع نرم‌افزارهای واسط بین محیط فیزیکی
ذخیره و بازیابی اطلاعات و محیط منطقی برنامه‌سازی
است.



DBMS به کاربر امکان می دهد تا:

- 1- پایگاه داده های خود را تعریف کند.
- 2- در پایگاه داده های خود عملیات انجام دهد.
- 3- روی پایگاه داده های خود تا حدی کنترل داشته باشد.



رده‌بندی سیستم‌های DBMS

از نظر نوع ساختار داده‌ای

- سیستم رابطه‌ای
- سیستم سلسله‌مراتبی
- سیستم شبکه‌ای
- جز اینها



رده‌بندی سیستم‌های DBMS

از نظر محیط سخت‌افزاری

- وابسته به یک محیط خاص
- ناوابسته به یک محیط خاص



رده‌بندی سیستم‌های DBMS

از نظر رده کامپیوتر

- خاص محیط کامپیوترهای شخصی
- خاص محیط کامپیوترهای متوسط (Mini Computer)
- خاص محیط کامپیوترهای بزرگ (Main Computer)
- خاص محیط کامپیوترهای خیلی بزرگ (Super Computer)
- اجراشونده در چند رده کامپیوتر



رده‌بندی سیستم‌های DBMS

از نظر محیط سیستم عامل

- وابسته به یک سیستم عامل خاص
- اجراشونده در محیط چند سیستم عامل

از نظر نوع معماری سیستم پایگاه داده‌ها

- با توانش ایجاد پایگاه متمرکز
- با توانش ایجاد پایگاه نامتمرکز



رده‌بندی سیستم‌های DBMS

از نظر معماری مشتری - خدمت‌گزار

- با توانش ایجاد معماری چند مشتری — يك خدمت‌گزار
- با توانش ایجاد معماری چند مشتری — چند خدمت‌گزار

از نظر زبان

- سیستم دارای SQL
- سیستم فاقد SQL



رده‌بندی سیستم‌های DBMS

از نظر نوع زبان داده‌ای فرعی

- دارای I.DSL
- دارای E.DSL
- دارای E/I.DSL

از نظر ماهیت زبان داده‌ای فرعی

- با زبان رویه‌ای
- با زبان نارویه‌ای



رده‌بندی سیستم‌های DBMS

از نظر سیستم فایل

- خودکفا
- وابسته به سیستم فایل محیط سیستم عامل

از نظر نوع کاربرد

- تک منظوره
- همه منظوره



رده‌بندی سیستم‌های DBMS

از نظر قیمت

از حدود ده هزار دلار تا صد هزار دلار و گاه بیشتر

از نظر طرز برپایی

- با محدودیت برپایی یکپارچه
- دارای امکان برپایی گزینشی



رده‌بندی سیستم‌های DBMS

از نظر واسط کاربر

- با واسط زبانی
- با واسط غیر زبانی
- با هر دو واسط



رده‌بندی سیستم‌های DBMS

از نظر رفتار در قبال رویدادها

- سیستم فعال
- سیستم غیر فعال

از نظر متدولوژی زبان

- بدون متدولوژی شیء‌گرایی
- دارای متدولوژی شیء‌گرایی



رده‌بندی سیستم‌های DBMS

از نظر بهینه‌سازی پرسش

- دارای بهینه‌سازی متعارف
- دارای بهینه‌سازی مبتنی بر قاعده ، معنایی و ...

از نظر نوع تراکنش

- پذیرنده تراکنشهای ساده
- پذیرنده تراکنشهای با مدل پیشرفته



رده‌بندی سیستم‌های DBMS

از نظر نوع پردازش

- با قابلیت پردازش بی‌درنگ
- فاقد این قابلیت

از نظر رسانه ذخیره‌سازی پایگاه داده‌ها

- با قابلیت ایجاد MMDB
- فاقد این قابلیت



رده‌بندی سیستم‌های DBMS

از نظر قابلیت تعامل بین سیستم‌ها

- فاقد این قابلیت
- دارای قابلیت تعامل با سیستم‌های همگن
- دارای قابلیت تعامل با سیستم‌های ناهمگن

از نظر پردازش داده‌های زمانمند

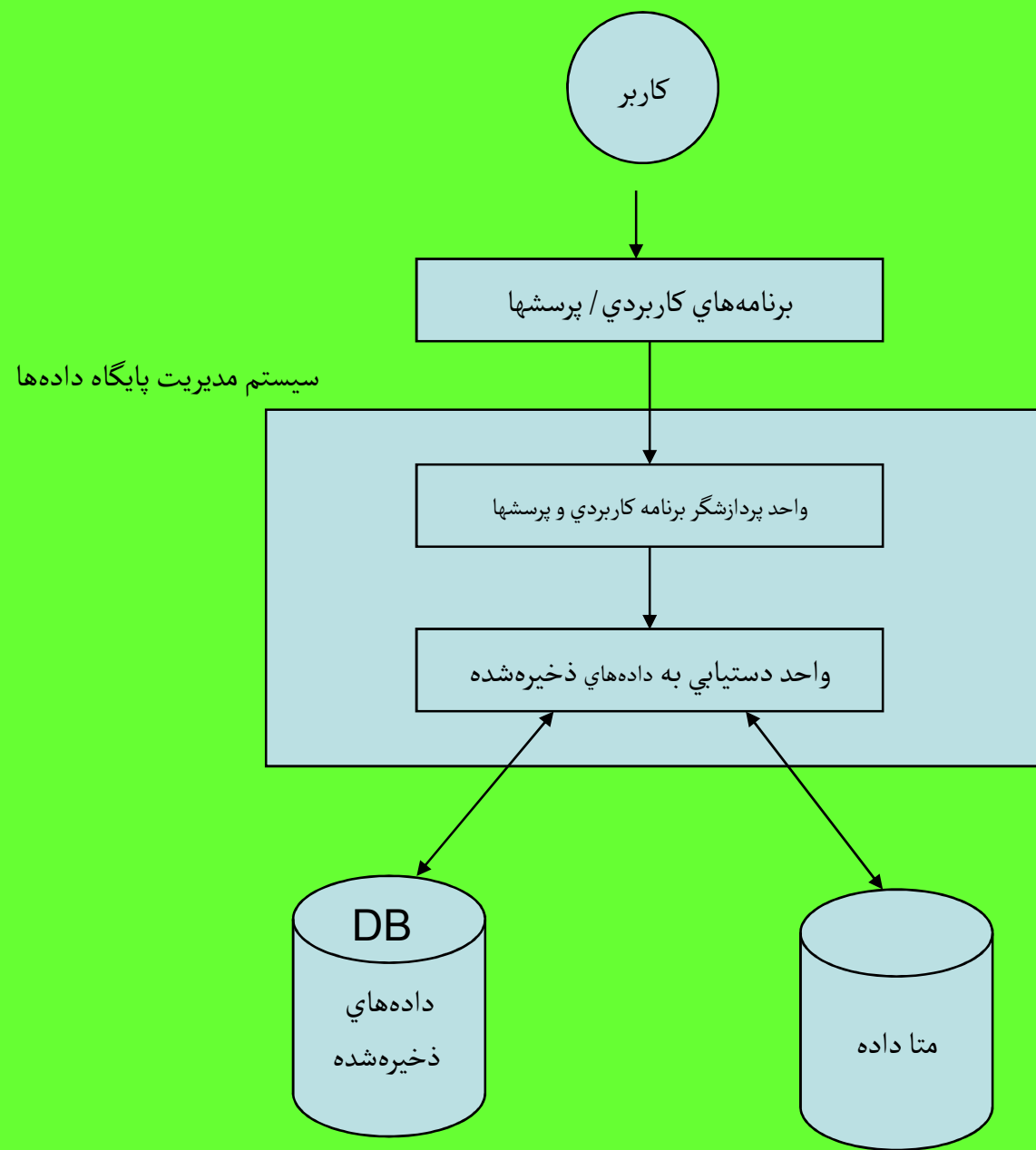
- فاقد جنبه‌های یک سیستم زمانی
- سیستم مدیریت پایگاه داده‌های زمانی



اجزای DBMS از نمای بیرونی

- واحد پردازشگر پرسش‌ها و برنامه‌های کاربردی
- واحد ایجاد و مدیریت داده‌های ذخیره‌شده

نمای بیرونی (ساده شده) DBMS





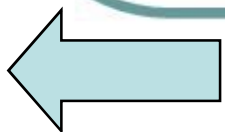
اجزای DBMS از نمای درونی

- لایه هسته (سیستم کنترل یا موتور پایگاه داده‌ها)
- لایه مدیریت محیط پایگاه داده‌ها
- لایه تسهیلات نرم‌افزاری (ابزارها)



واحدهای لایه هسته

- 1- واحد دریافت درخواست کاربر و واریسی‌های اولیه
- 2- واحد تولید شماها
- 3- پیش‌گامپایرها برای DML
- 4- گامپایرها (پردازنده DML)
- 5- پردازشگر پرسش و بهینه‌ساز پرسش
- 6- واحد مدیریت سطح داخلی



ادامه



واحدهای لایه هسته

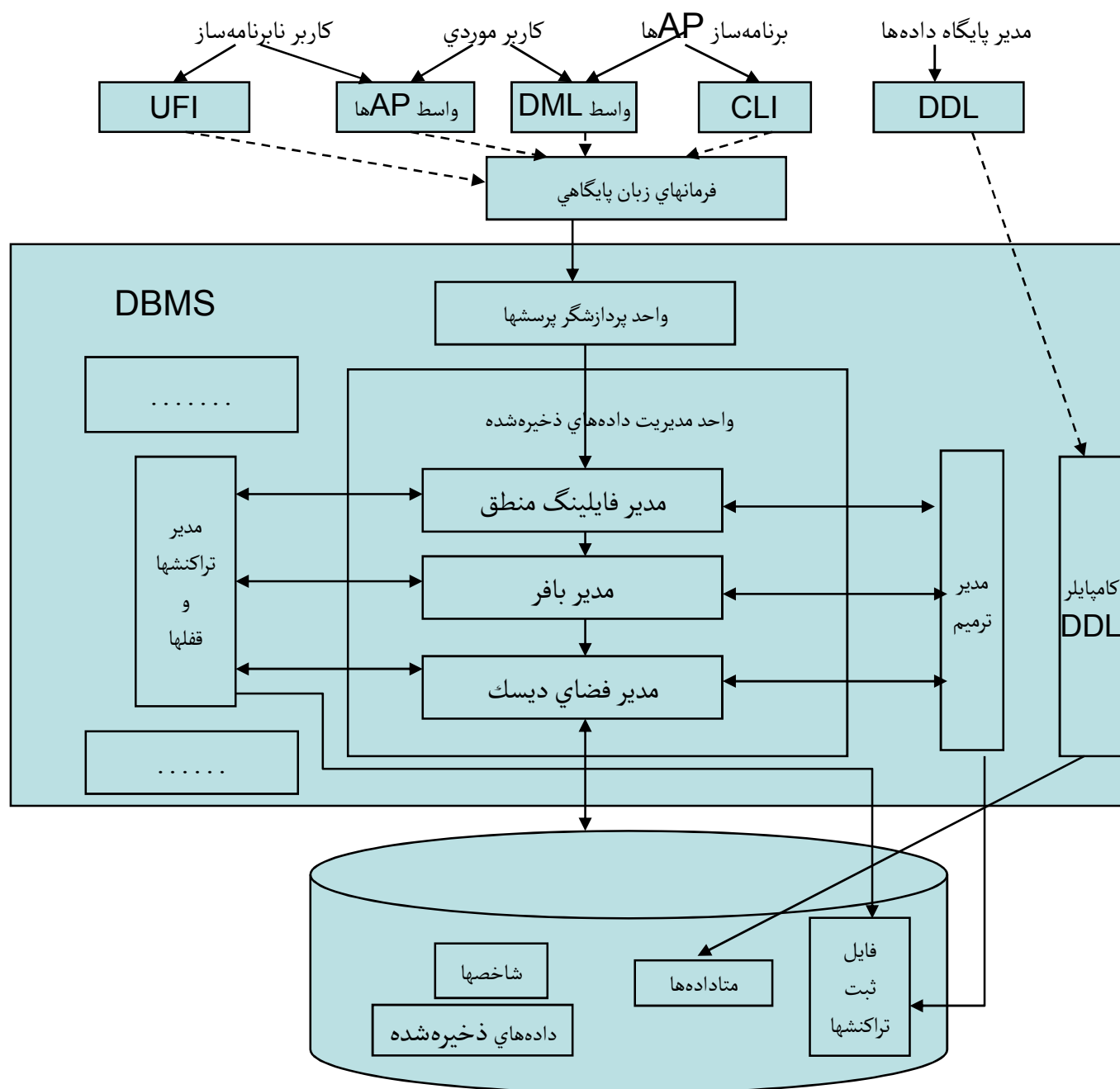
- 7- واحد مدیریت بافر
- 8- واحد مدیریت فضای دیسک
- 9- واحد ناظر زمان اجرا
- 10- واحد مدیریت همروندی تراکنش‌ها
- 11- واحد مدیریت انتقال داده‌ها
- 12- واحد مدیریت کاتالوگ



واحدهای لایه مدیریت محیط پایگاه داده‌ها

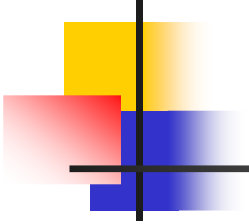
- 1- واحد کنترل جامعیت پایگاه داده‌ها
- 2- واحد ترمیم پایگاه داده‌ها
- 3- واحد ایمنی و حفاظت پایگاه داده‌ها
- 4- واحد تولید نسخه‌های پشتیبان
- 5- واحد تولید فایل‌های ثبت تراکنشها

ساختار يك سيستم پايگاهي



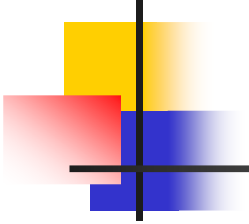
جلسه هفتم

DBMS در يك سيستم كامپيوتري



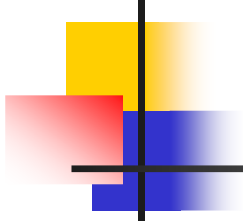
آنچه در این جلسه می خوانید:

- 1- جایگاه DBMS در يك سیستم کامپیوتری
- 2- سه طرح کاربرد DBMS در برنامه های کاربردی
- 3- اسلوبهای عملیاتی
- 4- کاتالوگ سیستم و دیکشنری داده ها: متا داده ها
- 5- اطلاعاتی که در دیکشنری داده ها نگهداری می شود
- 6- دیکشنری داده ها و استفاده کنندگان آن
- 7- پارامترهای شناخت DBMS
- 8- مدیر پایگاه داده ها



آنچه در این جلسه می خوانید:

- 9- برخی مسئولیتهای در تیم مدیریت پایگاه دادهها
- 10- پنج سرمایه در مدیریت نوین سازمانها
- 11- مزایا و معایب تکنولوژی پایگاه دادهها
- 12- مزایا و معایب سیستم تک کاربری
- 13- مزایا و معایب سیستم چند کاربری
- 14- شرایط استفاده از تکنولوژی پایگاه دادهها
- 15- کاربردهای جدید تکنولوژی پایگاه دادهها
- 16- استقلال دادهای و انواع آن



هدفهاي كلي: DBMS در يك سيستم كامپيوتري و مدير پاينگاه داده‌ها

هدفهاي رفتاري: دانشجو در پايان اين جلسه مي‌تواند:

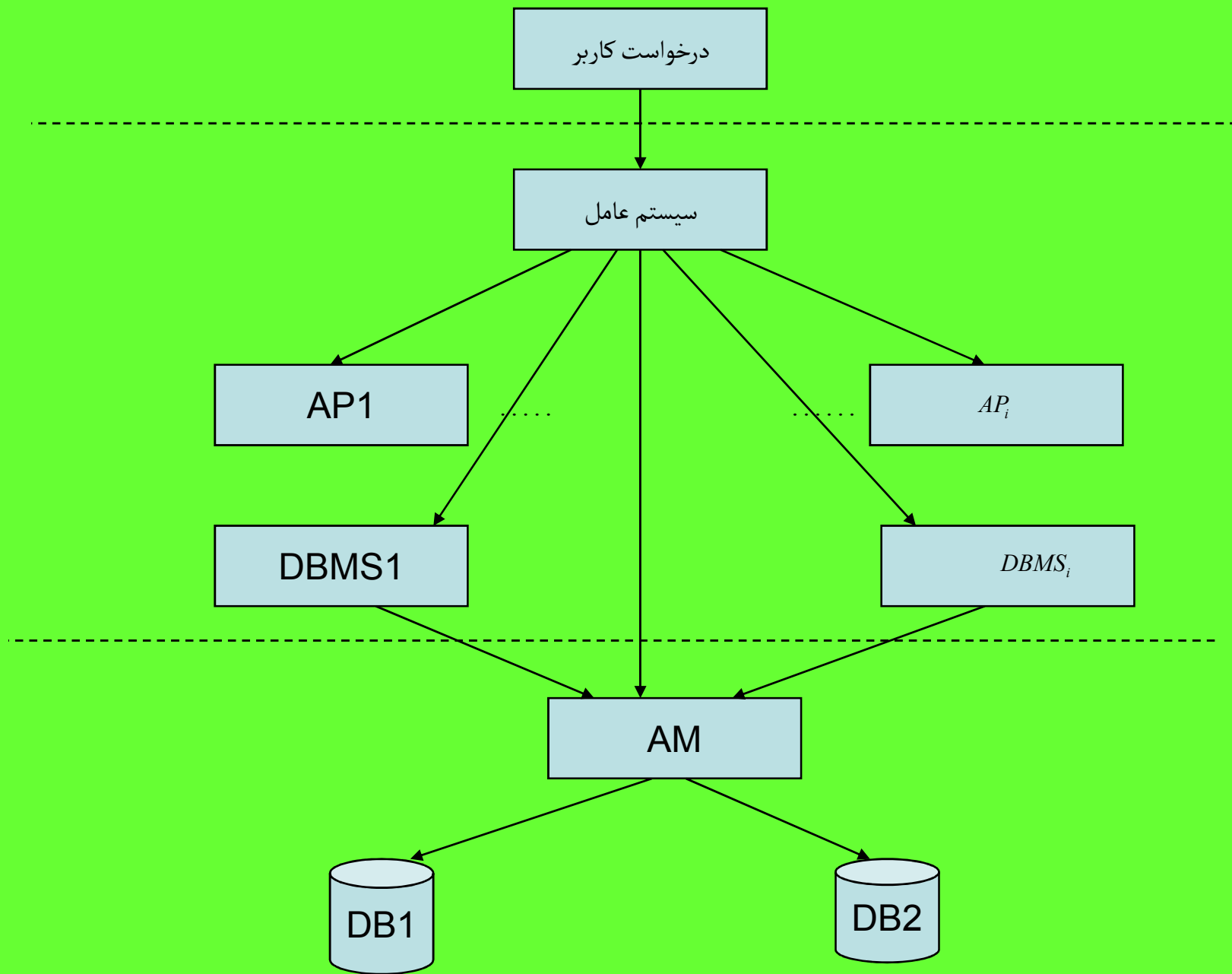
- ارتباط DBMS با ساير عناصر نرم‌افزاري را بيان کند
- اسلوبهاي عملياتي ، و پارامترهاي شناخت را نام ببرد.
- مسؤليتهاي تيم مديريت پاينگاه داده‌ها را شرح دهد.
- مزاي و معايب سيستم تك‌كاربره و چندكاربره را توصيف کند.
- استقلال داده‌اي و انواع آن را توضيح دهد.



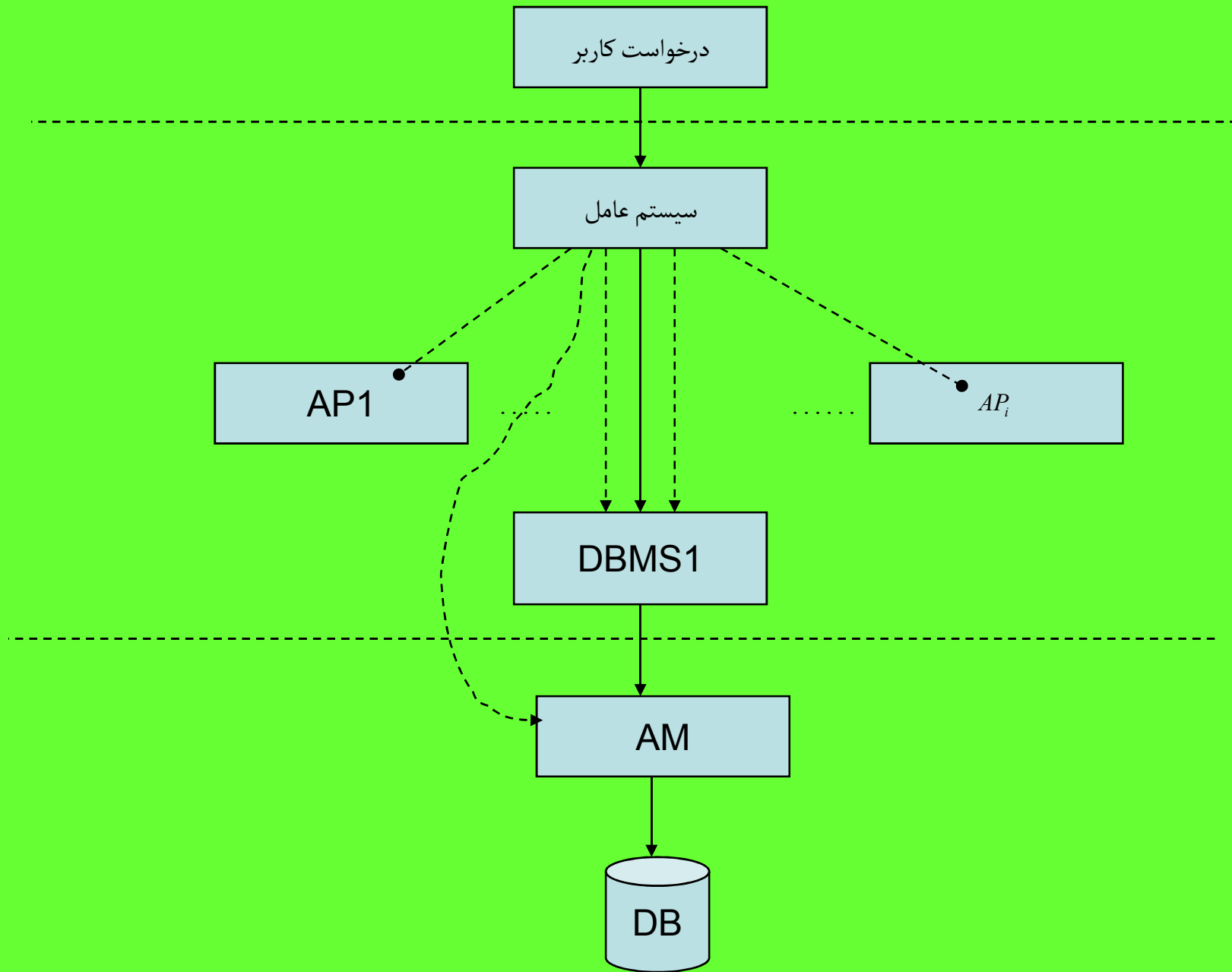
جایگاه DBMS در يك سیستم کامپیوتری



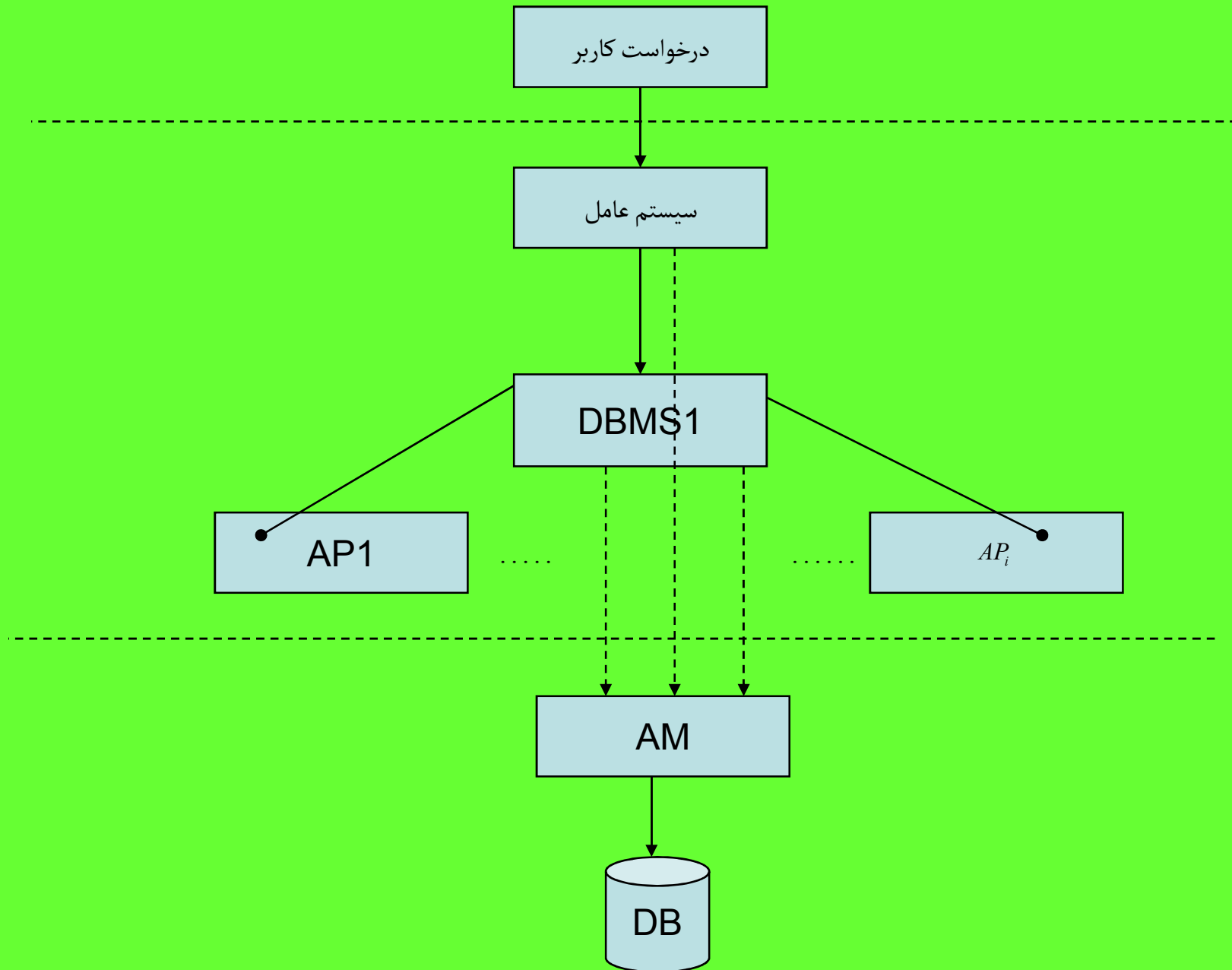
يك DBMS براي هر برنامه کاربردي



يك DBMS براي چند برنامه کاربردي تحت كنترل سيستم عامل



يك DBMS براي چند برنامه کاربردي با اجراي تحت كنترل DBMS





اسلوبهاي عملياتي

به طور كلي سه اسلوب عملياتي وجود دارد:

- اسلوب يکجا
- اسلوب برخط
- اسلوب تعاملتي



کاتالوگ سیستم و دیکشنری داده‌ها: متا داده‌ها

حاوی داده‌هایی است در مورد داده‌های ذخیره‌شده در پایگاه داده‌های کاربر و این داده‌ها به متاداده‌ها موسومند

دیکشنری داده‌ها معمولاً جزئی از خود سیستم است و به دو صورت فعال و غیرفعال تولید می‌شود.



اطلاعاتی که در دیکشنری داده‌ها نگهداری می‌شود:

- شمای خارجی
- شمای ادراکی
- شمای داخلی
- رویه‌های مربوط به تبدیلات بین سه سطح معماری
- شرح ساختار فیزیکی داده‌های ذخیره‌شده
- مشخصات و حقوق دستیابی کاربران به داده‌ها
- مشخصات برنامه‌های کاربردی تولید شده و ارتباط آنها با درخواستهای کاربران
- مشخصات پایانه‌های متصل به سیستم



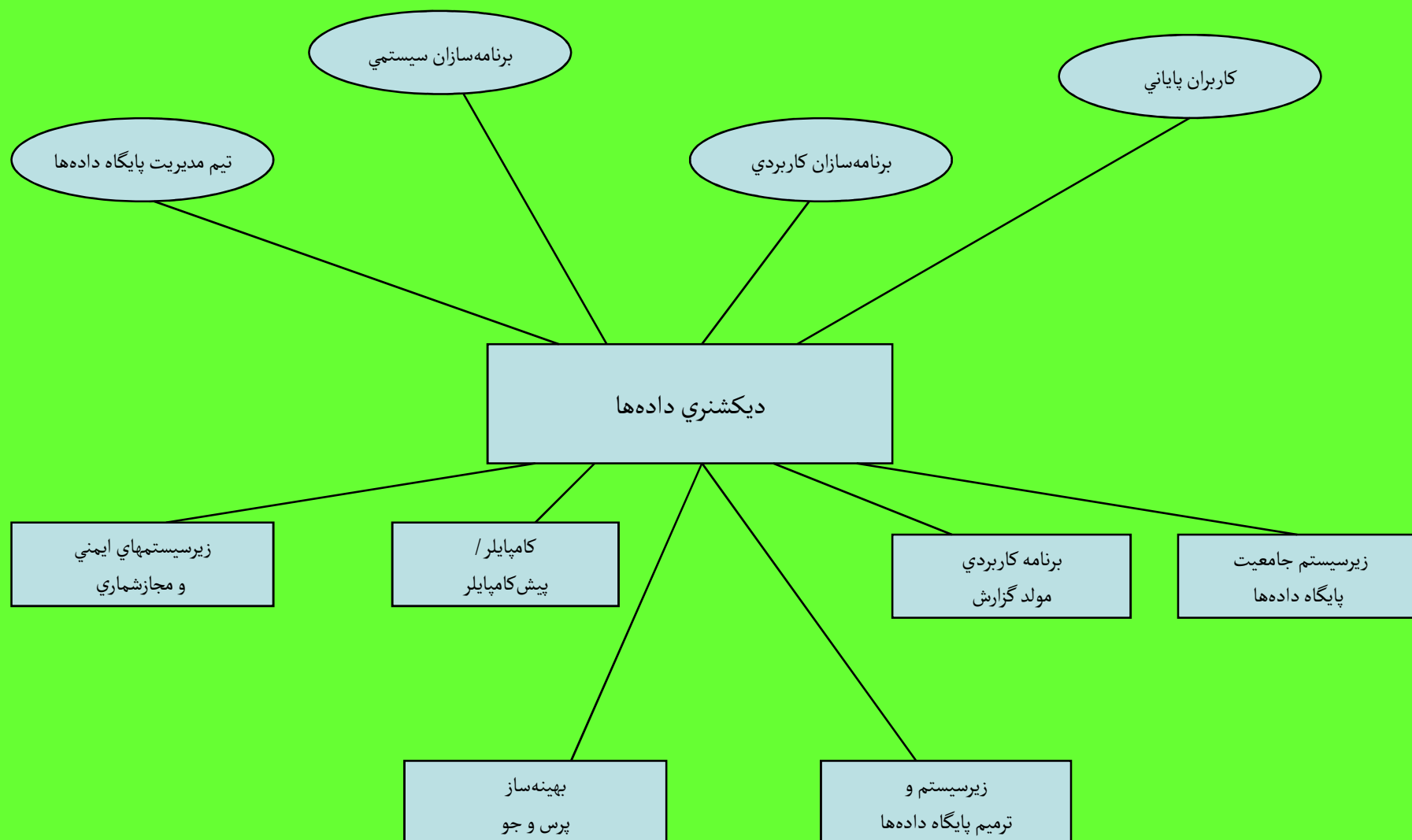
ادامه



اطلاعاتي که در دیکشنري داده‌ها نگهداري مي‌شود:

- ارتباط بين برنامه‌هاي کاربردي و داده‌هاي ذخيره‌شده
- قواعد جامعيت
- ضوابط کنترل ايمني داده‌ها
- مشخصات پیکربندي سخت‌افزاري سيستم و رسانه‌هاي ذخيره‌سازي
- اطلاعات متنوع آماري پایگاه داده و کاربران
- توابع تعريف‌شده توسط کاربران

دیکشنري داده‌ها و استفاده‌کنندگان آن





پارامترهاي شناخت DBMS

- پارامترهاي مربوط به توانش ها و كارايي سيستم
- تسهيلات و جنبه هاي ديگر
- مشخصات كلي سيستم
- پارامترهاي مربوط به معماري پايگاه داده ها
- پارامترهاي مربوط به زبان داده اي فرعي



مدیر پایگاه داده‌ها

فردی است متخصص در پایگاه داده‌ها و با مسئولیت علمی، فنی و نیز اداری در محدوده وظایفی که عهده‌دار است

این مدیر همراه با یک تیم تخصصی کار می‌کند که به آن تیم مدیریت پایگاه داده‌ها می‌گویند.



برخی مسئولیتهای در تیم مدیریت پایگاه داده‌ها

- مدیر پایگاه داده‌ها
- مدیر داده‌ها
- مدیر امور پژوهش-توسعه
- مدیر سیستمهای کاربردی
- مسئول تیمهای برنامه‌سازی
- مسئول کنترل کارایی DBMS
- مسئول کنترل کارایی خود سیستم پایگاه داده‌ها
- مسئول نظارت بر عملیات روی پایگاه داده‌ها
- مسئول تماس با کاربران زیرمجموعه‌های سازمان
- مسئول تنظیم مستندات و وضع استانداردها



در مدیریت نوین سازمانها ، هر سازمان دارای پنج سرمایه
است:

1- سخت افزار

2- نرم افزار

3- داده

4- بودجه

5- تخصص



مزایا و معایب تکنولوژی پایگاه داده‌ها

مزایای این تکنولوژی بستگی به نوع سیستم (DBMS) و معماری سیستم پایگاه داده‌ها و ماهیت کاربردها دارد.



سیستم تک کاربری

مزایا

- 1- هر بخش از سازمان ، داده های خود را نگهداری و پردازش می کند.
- 2- با استفاده از کامپیوترهای شخصی ، حجم داده های سیستم مرکزی کاهش می آید.
- 3- پایگاه داده های ایجاد شده روی کامپیوترهای شخصی معمولاً کوچک و مدلسازی ، طراحی و پیاده سازی آنها ساده است.
- 4- کار با این سیستمها و برنامه سازی در محیط آنها ساده است.
- 5- با پیشرفت کامپیوترهای شخصی ، این سیستمها می توانند بسیاری از کارهای سیستمهای کامپیوتری بزرگ را انجام دهند.



سیستم تڪ کاربري

معایب

- 1- وجود تعداد زيادي از اين سیستمها در يك سازمان باعث بروز افزونگي ، ناسازگاري داده‌ها و ناایمني آنها مي‌شود.
- 2- محدودیتهای سخت افزاري سبب محدودیت اندازه فایلها و نیز محدودیت سرعت پردازش باعث محدود شدن حجم پایگاه داده‌ها مي‌شود.
- 3- خود سیستم نمی‌تواند قوي و کارا باشد.



ادامه



سیستم تک کاربری

معایب

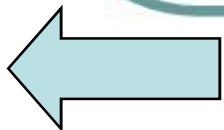
- 4- میزان ایمنی و حفاظت در آنها ضعیف است.
- 5- امکانات تولید نسخه پشتیبان در آنها معمولاً کم است.
- 6- اشتراکی کردن آنها مشکلات تکنیکی جدی دارد.
- 7- اعمال مجموعه واحدی از استانداردها در کل سازمان ناممکن است.
- 8- معمولاً کاربر این محیط مهارت کافی در مدلسازی و طراحی بهینه پایگاه داده‌ها ندارد.
- 9- ایجاد یک سیستم جامع و یکپارچه، براساس این سیستمها، دشوار و پرهزینه است.



سیستم چندکاربری

مزایا

- 1- اشتراك داده‌ها
- 2- کاهش افزونگی
- 3- تعدد شیوه‌های دستیابی به داده‌ها
- 4- اجتناب از ناسازگاری داده‌ها
- 5- تامین همروندی بهتر
- 6- تسهیل پردازش تراکنشها
- 7- تضمین جامعیت داده‌ها



ادامه



سیستم چند کاربری

مزایا

- 8- حفظ محرمانگی داده‌ها
- 9- امکان اعمال استانداردها
- 10- تعدد زبانها
- 11- کاهش حجم برنامه‌ها
- 12- تنوع کاربران
- 13- تسریع در دریافت پاسخ پرسشها
- 14- استفاده بهتر از سخت افزار و ...



سیستم چند کاربری

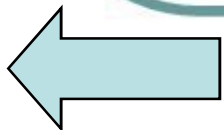
معایب

- 1- هزینه بالای نرم افزار و سخت افزار
- 2- هزینه بیشتر برای برنامه سازی
- 3- هزینه بالا برای انجام مهندسی دوباره به منظور تبدیل سیستم از مشی ناپایگاهی به مشی پایگاهی
- 4- کند شدن اجرای بعضی از برنامه های کاربردی
- 5- خطر آسیب پذیری داده ها
- 6- تاثیرات گسترده تر خرابیها و دشواری بیشتر ترمیم آنها
- 7- پیچیده بودن سیستم و نیاز به تخصص بیشتر



شرایط استفاده از تکنولوژی پایگاه داده‌ها

- 1- نیاز به ایجاد يك سیستم یکپارچه اطلاعاتي
- 2- حجم زیاد داده‌هاي سازمان و رشد پوياي آن
- 3- تغییرات مداوم در داده‌هاي ذخیره‌شده
- 4- بالا بودن بسامد درخواستهاي کاربران
- 5- نیاز به اعمال کنترل متمرکز و دقیق روی کل داده‌ها
- 6- وجود ارتباطات پیچیده بین داده‌ها



ادامه



شرایط استفاده از تکنولوژی پایگاه داده‌ها

- 7- زیاد بودن میزان داده‌های مشترک بین برنامه‌های کاربردی
- 8- مدنظر بودن صحت ، دقت و سازگاری داده‌ها
- 9- زیاد بودن گزارشها
- 10- نیاز به انجام پردازشهای تحلیلی برخط
- 11- نیاز به سیستم داده‌کاوی و کشف دانش در سازمان



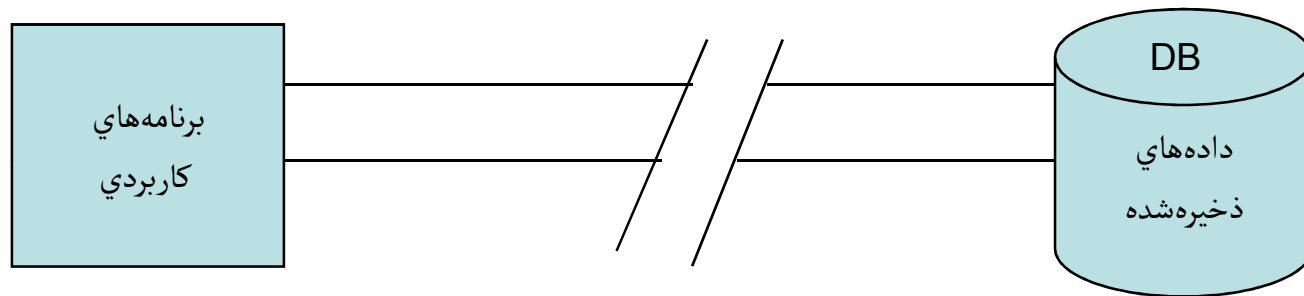
کاربردهای جدید تکنولوژی پایگاه داده‌ها

- سیستم پشتیبان تصمیم
- سیستم داده‌کاوی چندرسانه‌ای
- سیستم انبارش داده‌ها
- حیطه‌های فضایی و جغرافیایی
- پایگاه داده‌های شخصی و همراه
- سیستم پایگاه داده‌ها در شبکه جهانی اطلاع‌رسانی
- سیستم اطلاعات اجرایی
- سیستم اطلاعات طراحی
- سیستم پردازش تحلیلی برخط چندبعدی
- ...



استقلال داده‌اي

يعني وابسته نبودن برنامه‌هاي کاربردي به داده‌هاي ذخيره شده





انواع استقلال داده‌ای

- استقلال داده‌ای فیزیکی

- استقلال داده‌ای منطقی



استقلال داده‌ای فیزیکی

عبارتست از مصونیت دیدهای کاربران و برنامه‌های کاربردی در
قبال تغییرات در سطح داخلی-فیزیکی پایگاه داده‌ها



چرا استقلال داده‌ای فیزیکی در سیستم‌های رابطه‌ای جدید کاملاً تأمین است؟

زیرا:

- 1- کاربران سطح خارجی در محیطی کاملاً انتزاعی عمل می‌کنند و برنامه‌های کاربردی در این سطح با فایلینگ پایگاه داده‌ها تماس ندارند.
- 2- بین سطح خارجی و داخلی، یک سطح انتزاعی دیگر واسط است و مانع تأثیرپذیری برنامه‌های سطح خارجی از تغییرات در فایلینگ پایگاه داده‌ها می‌شود.



استقلال داده‌ای منطقی

عبارتست از مصونیت دیدهای کاربران و برنامه‌های کاربردی در قبال
تغییرات در سطح ادراکی پایگاه داده‌ها



تغییر در سطح ادراکی یعنی تغییر در طراحی منطقی پایگاه
داده‌ها و تغییر در شمای ادراکی

این تغییر معمولاً منجر به تغییر مناسب در سطح فایلینگ پایگاه
می‌شود.



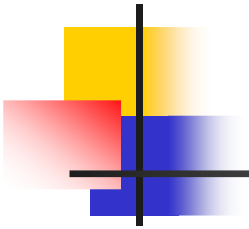
تغییر در سطح ادراکی دو وجه دارد:

1- رشد پایگاه در سطح ادراکی

2- سازمان دهی مجدد پایگاه در سطح ادراکی

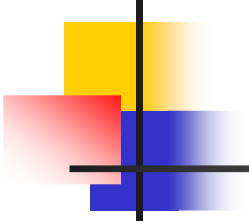
جلسه هشتم

معماري سيستم پايگاه داده‌ها



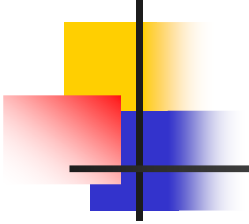
آنچه در این جلسه می خوانید:

- 1- معماری سیستم پایگاه داده ها
- 2- انواع معماری
- 3- معماری متمرکز
- 4- معماری مشتری - خدمتگذار
- 5- طرحهای معماری مشتری - خدمتگذار
- 6- مزایای معماری مشتری — خدمتگذار در مقایسه با معماری متمرکز
- 7- معماری توزیع شده



آنچه در این جلسه می خوانید:

- 8- ویژگیهای معماری توزیع شده
- 9- مزایا و معایب معماری توزیع شده
- 10- معماری با پردازش موازی
- 11- طرح های معماری با پردازش موازی
- 12- معماری سیستم چندپایگاهی
- 13- معماری سیستم پایگاههای همراه



هدفهای کلی: معماری سیستم پایگاه داده‌ها

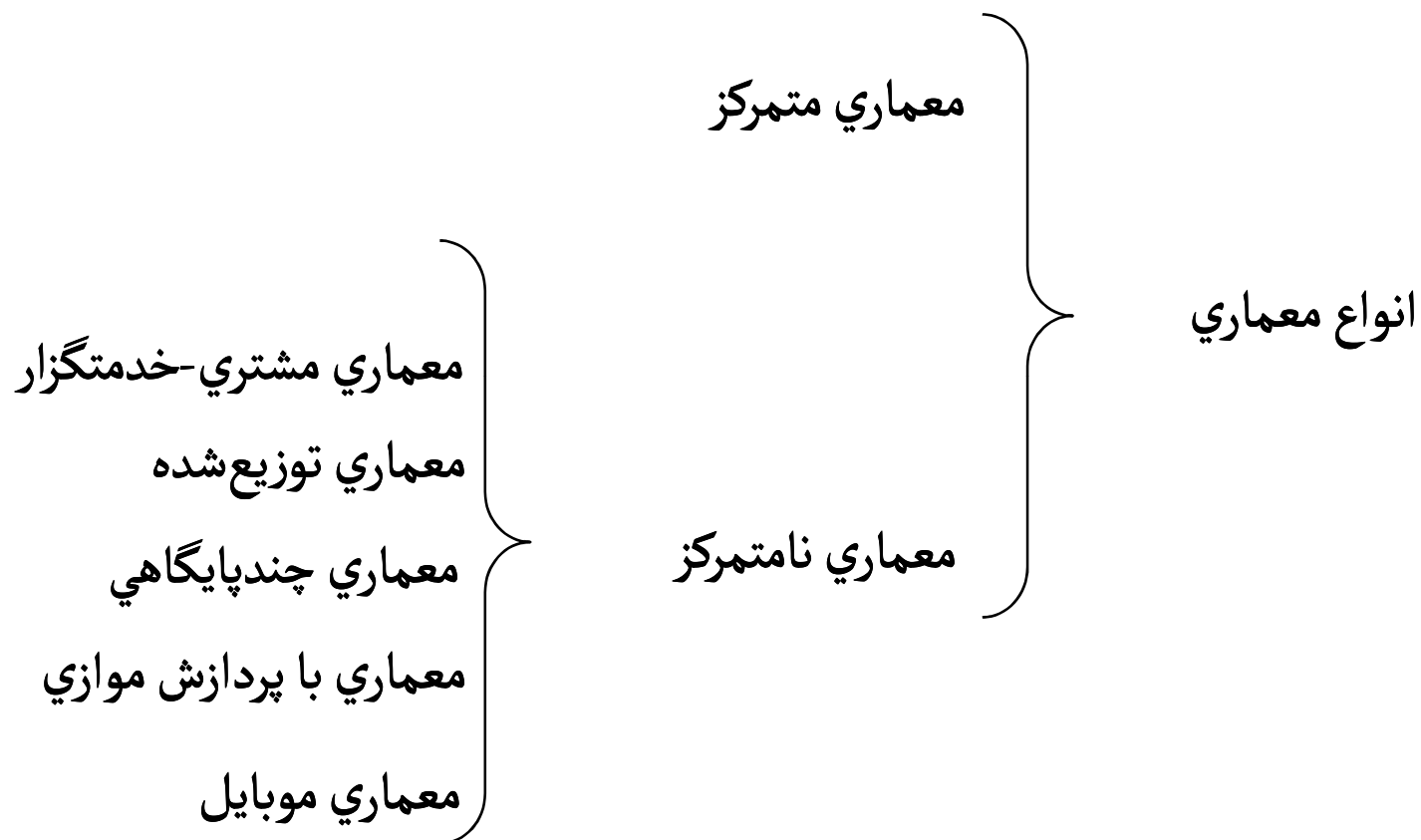
هدفهای رفتاری: دانشجو در پایان این جلسه می‌تواند:

- گونه‌های مختلف معماری پایگاه داده‌ها شامل معماری متمرکز و نامتمرکز و انواع معماری نامتمرکز و ویژگیهای هر یک را شرح دهد.



معماري سيستم پاىگاه داده‌ها

منظور ، چندي و چوني اجزاء تشكيل دهنده سيستم و نيز پيكربندي يا طرز تركيب اجزاء سيستم و چگونگي تعامل اجزاء با يكدیگر است. در اين معماری حداقل يك پایگاه داده‌ها ، يك سيستم مدیریت پایگاه داده‌ها ، يك سيستم عامل ، يك کامپیوتر با دستگاههاي جانبي و تعدادي برنامه کاربردي و کاربر وجود دارند.

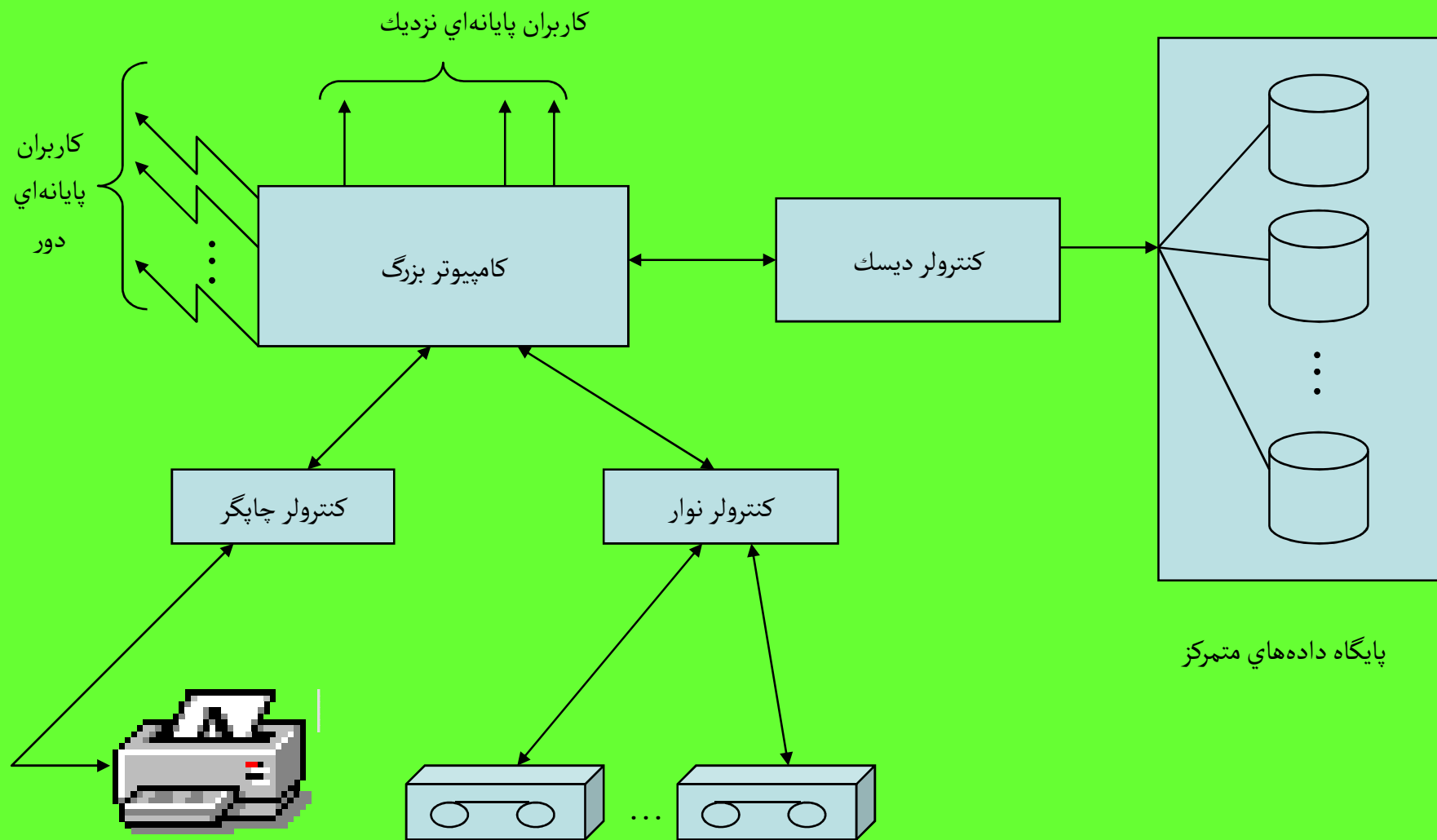




معماري متمرکز

در این معماری یک پایگاه داده روی یک سیستم کامپیوتری و بدون ارتباط با سیستم دیگر ایجاد می شود .

نمایش معماری متمرکز

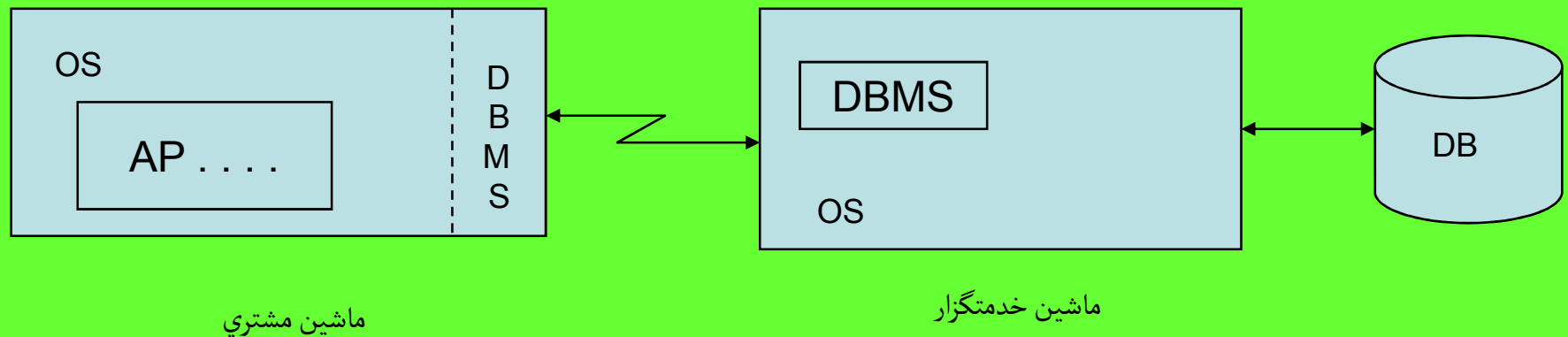




معماري مشتري - خدمتگزار

هر معماری که در آن قسمتی از پردازش را یک برنامه ، سیستم یا ماشین انجام دهد و انجام قسمت دیگری از پردازش را از برنامه ، سیستم یا ماشین دیگر بخواهد معماری مشتري خدمتگزار نامیده می شود

معماري مشري - خدمتگزار





طرحهاي معماری مشتری - خدمتگذار

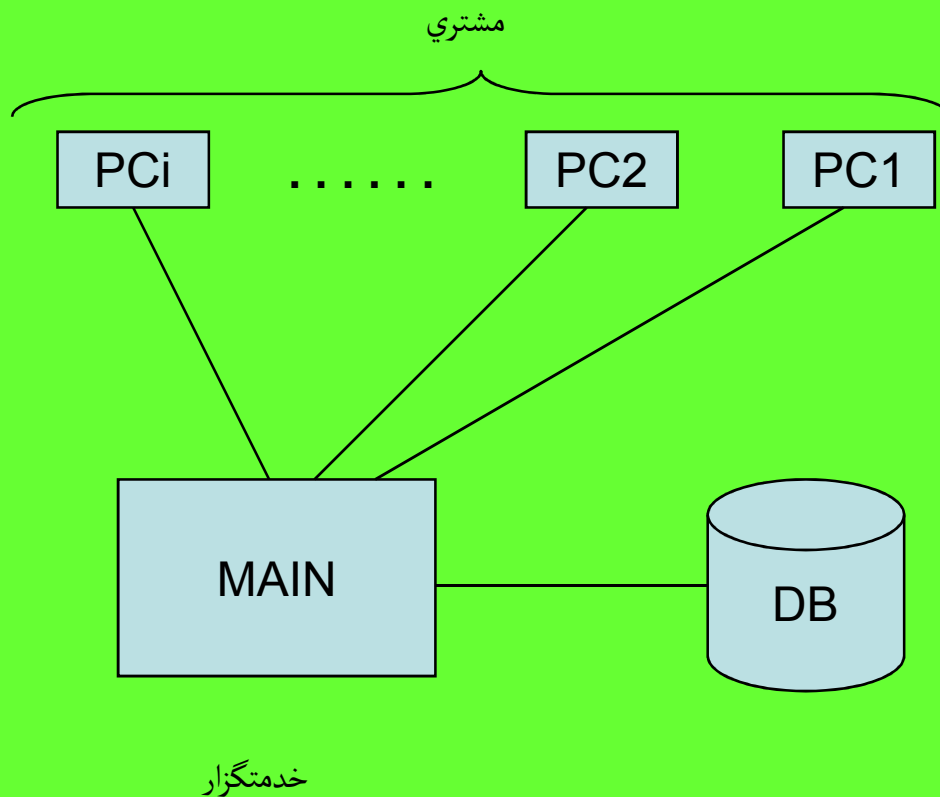
از نظر تعداد مشتری و خدمتگذار

- 1- چند مشتری — يك خدمتگذار
- 2- يك مشتری — چند خدمتگذار
- 3- چند مشتری — چند خدمتگذار

طرحهاي معماري مشتري - خدمتگذار

از نظر پيكربندي سخت افزاري

الف- معماري حول كامپيوتر بزرگ

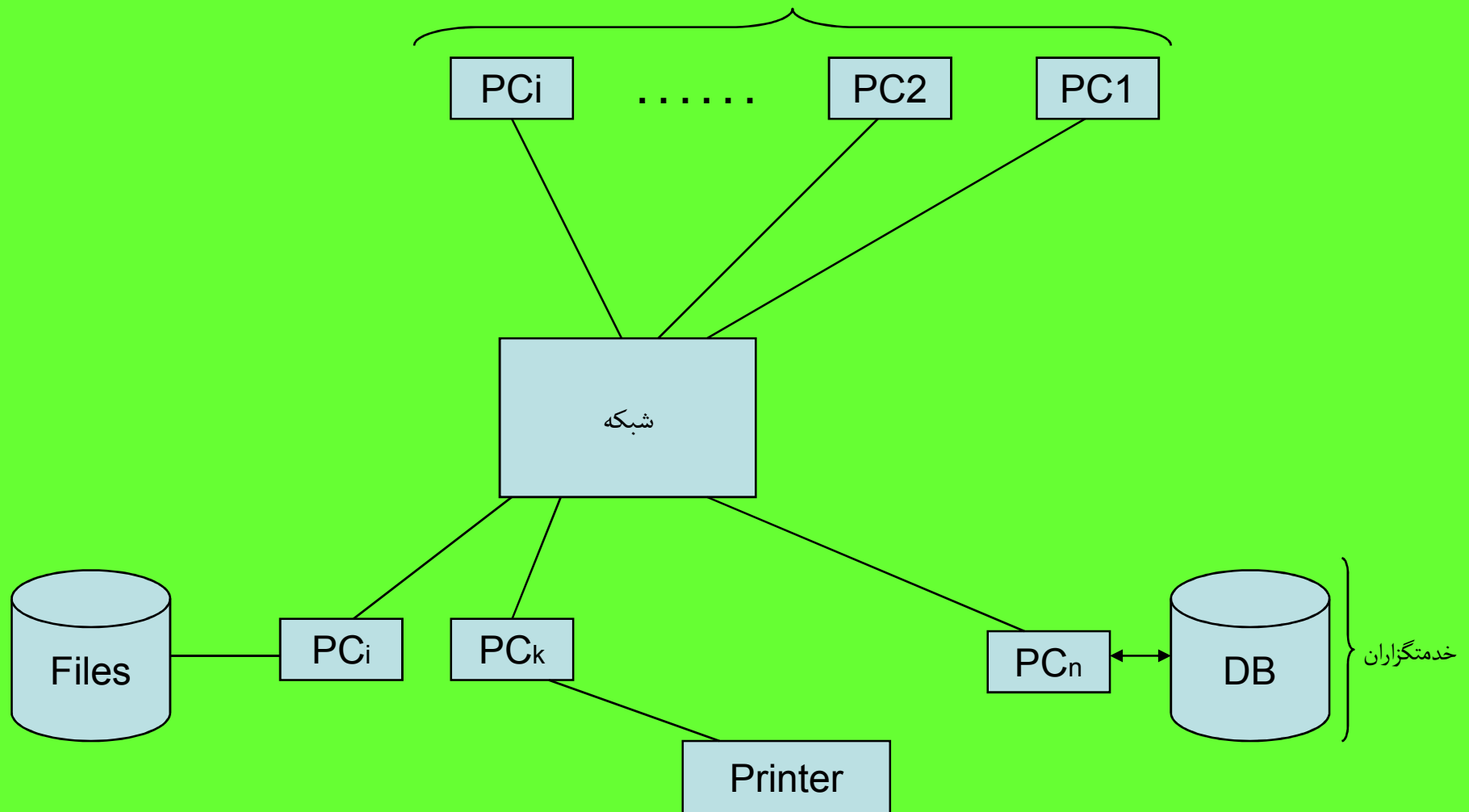


طرحهاي معماری مشتری - خدمتگذار

از نظر پیکربندی سخت افزاری

ب- معماری حول شبکه

مشتری





مزایای معماری مشتری — خدمتگذار در مقایسه با معماری متمرکز

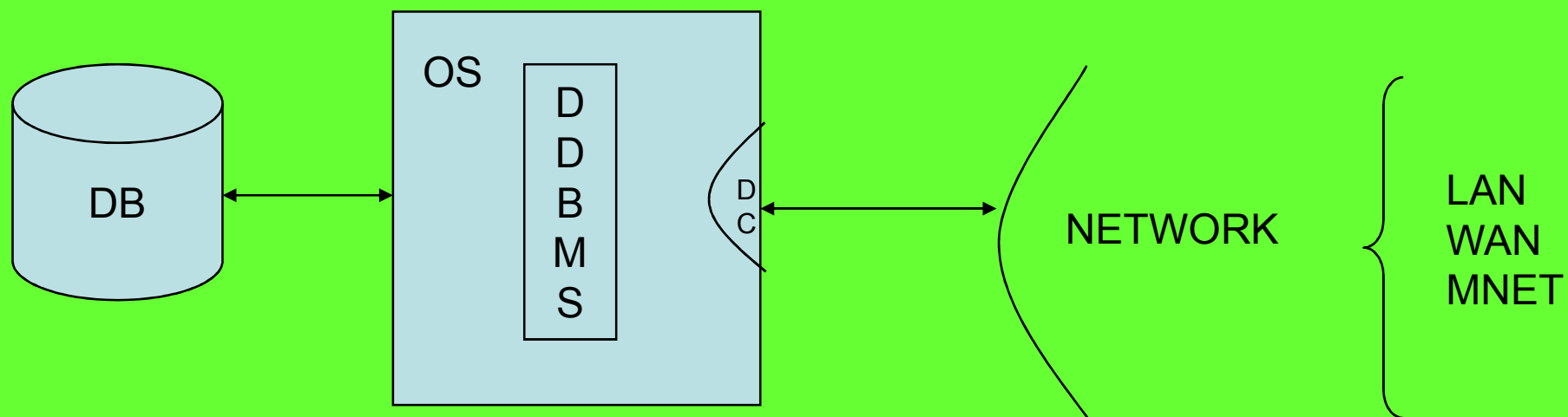
- تقسیم پردازش
- کاهش ترافیک شبکه
- استقلال ایستگاههای کار
- اشتراك داده‌ها



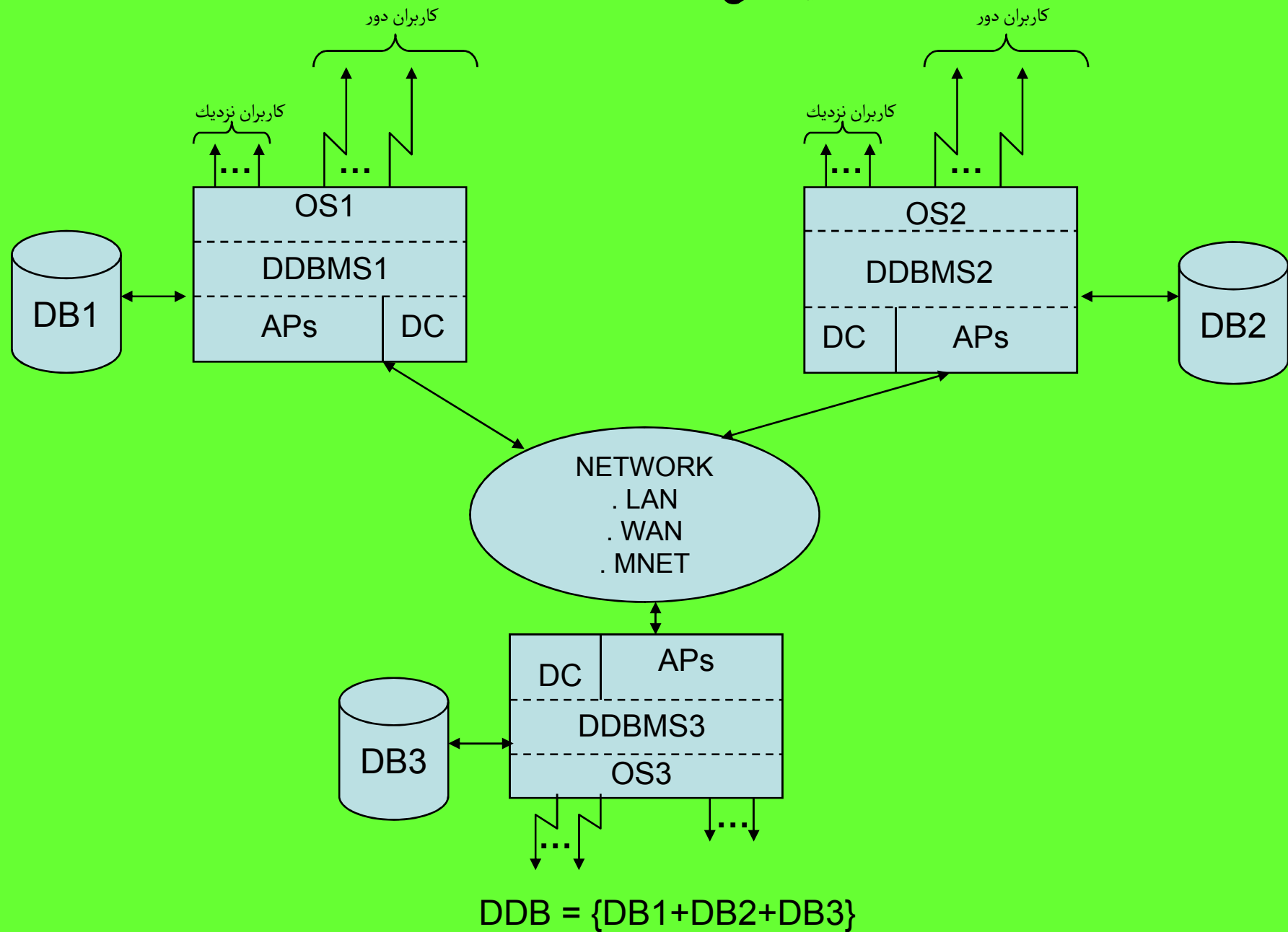
معماري توزيع شده

مجموعه اي از داده هاي ذخيره شده كه منطقا به يك سيستم تعلق دارند ولي در مانه هاي مختلف يك يا بيش از يك شبكه توزيع شده اند.

نمای یک مانه در معماری توزیع شده



معماري توزيع شده با سه مانه





ویژگیهای معماری توزیع شده

- مجموعه‌ای است از داده‌های منطقاً مرتبط و اشتراکی
- داده‌ها به بخشهایی تقسیم و در مانده‌ها توزیع شده‌اند.
- بعضی بخشها ممکن است به طور تکراری در مانده‌ها ذخیره شده باشند.
- مانده‌ها از طریق شبکه بهم مرتبط‌اند.
- داده‌های هر مانده تحت کنترل یک DBMS است.
- DMBS هر مانده ، می‌تواند برنامه‌های کاربردی محلی را
به طور خودکار اجرا کند.
- هر DBMS حداقل در اجرای یک برنامه کاربردی سرتاسری
مشارکت دارد.



مزایای معماری توزیع شده

- سازگاری و هماهنگی با ماهیت سازمانهای نوین.
- کارایی بیشتر در پردازش داده‌ها.
- دستیابی بهتر به داده‌ها.
- اشتراک داده‌ها.
- افزایش پردازش موازی.
- کاهش هزینه ارتباطات.
- تسهیل گسترش سیستم.
- استفاده از پایگاه داده‌های از قبل موجود.



معایب معماری توزیع شده

- پیچیدگی طراحی سیستم.
- پیچیدگی پیاده سازی.
- کاهش کارایی در برخی موارد.
- هزینه بیشتر.
- مصرف حافظه بیشتر.

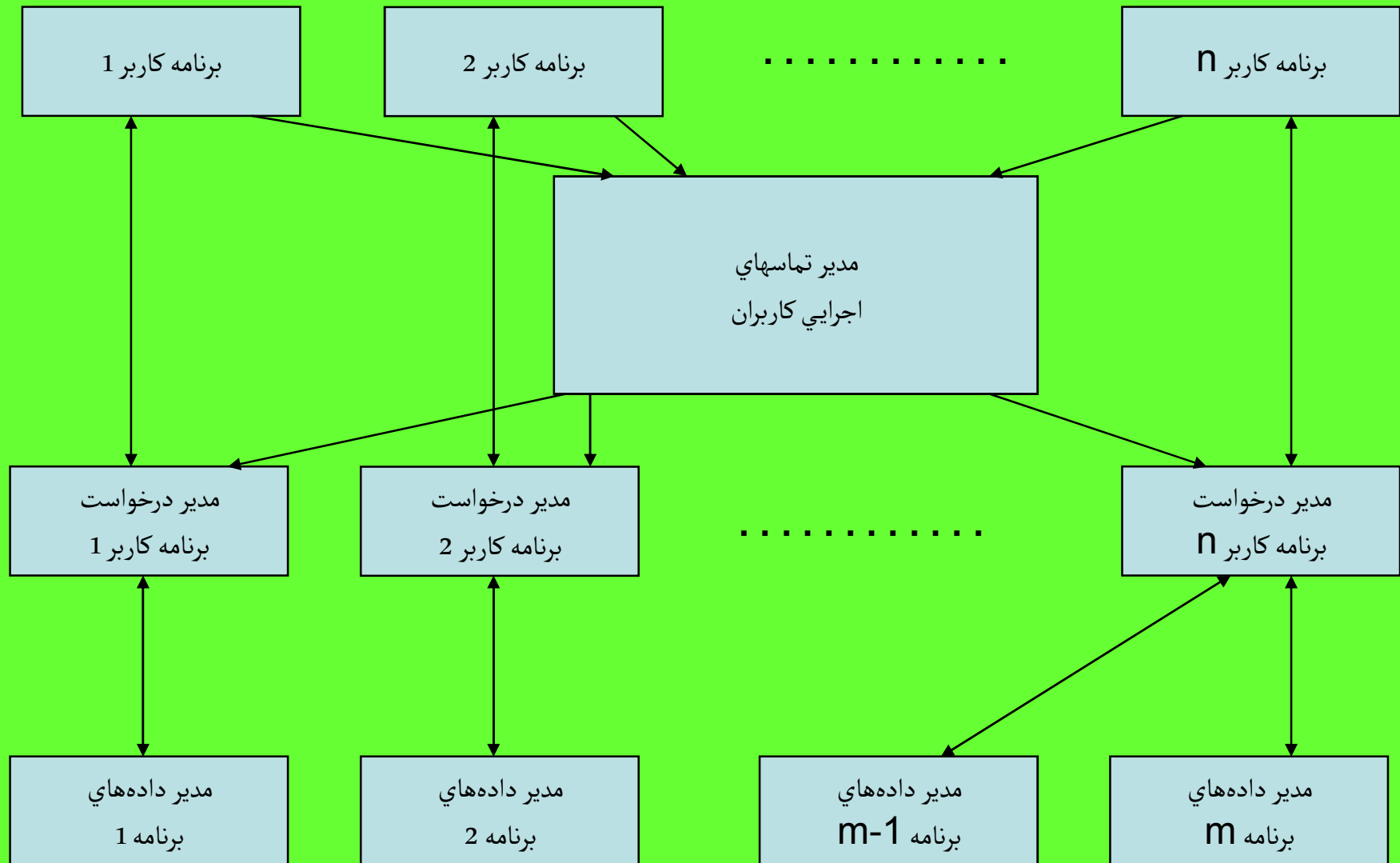


معماري با پردازش موازي

در اين گونه سيستمها معمولاً تعداد زيادي تراكنش در ثانيه و بطور موازي اجرا مي شود .

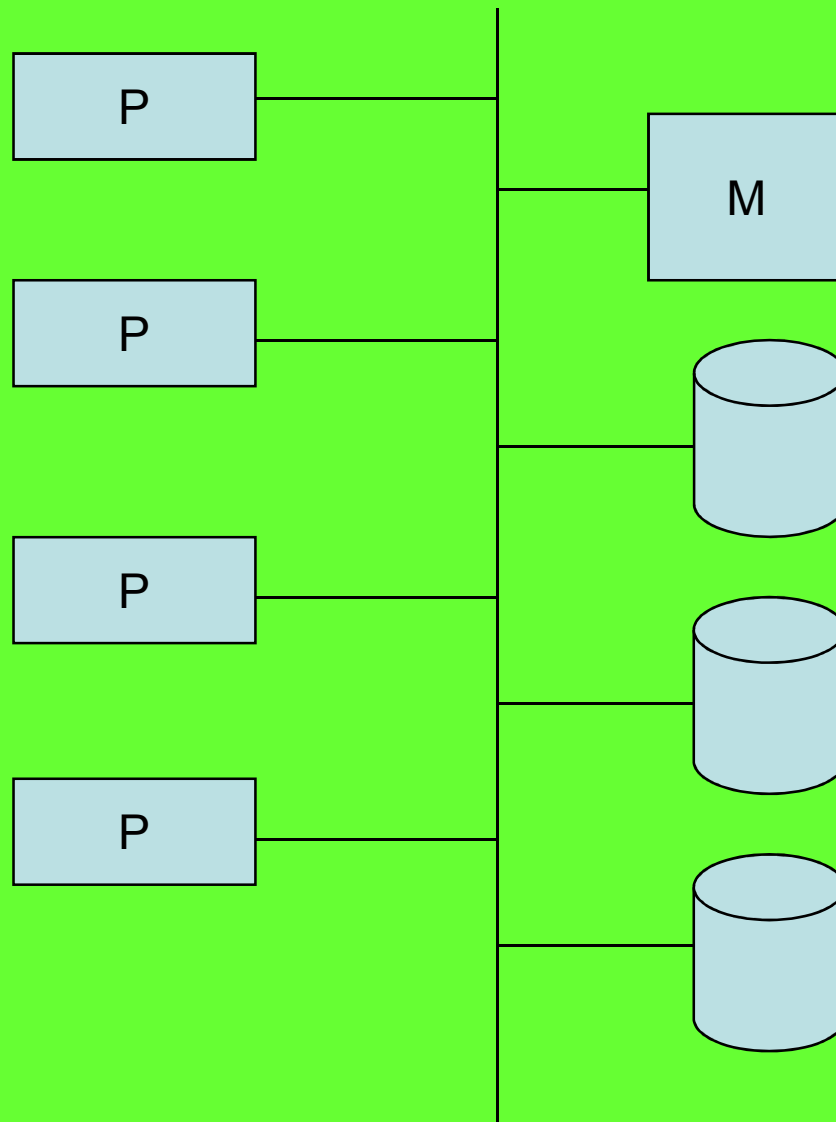
طرح کلي معماری با پردازش موازي

خدمتگذاران برنامه‌هاي کاربردي



خدمتگذار پایگاه داده‌ها

طرح هاي معماري با پردازش موازي



الف- معماری با حافظه مشترک

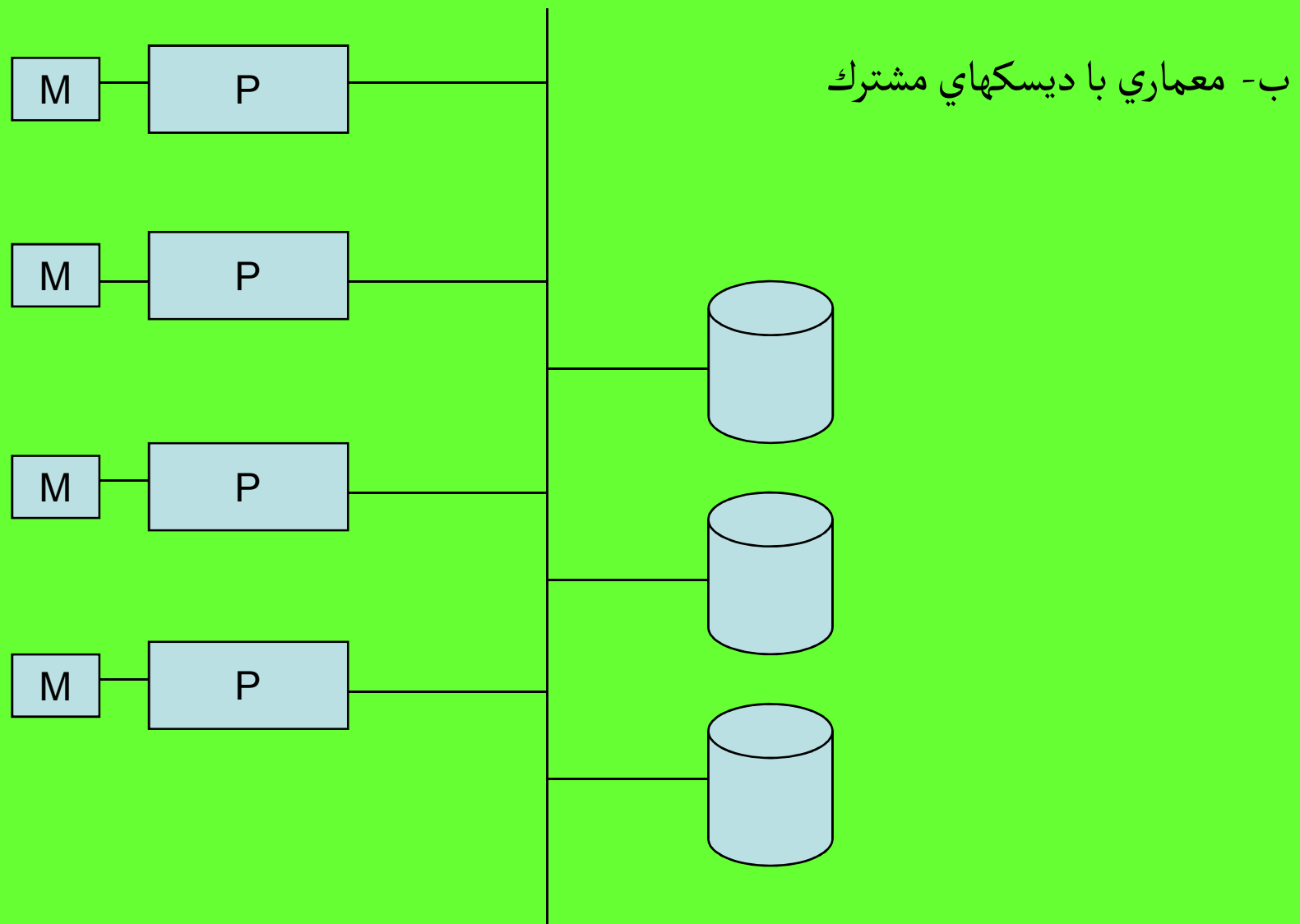


معماري با حافظه مشترك

مزيت: ارتباط بين پردازنده‌ها به طور كارا انجام مي‌شود.

عيب: نمي‌توان بيش از 32 يا 64 پردازنده داشت. زيرا احتمال بروز تنگنا در باسهاي حافظه‌اي يا شبكه ارتباطي افزايش مي‌يابد.

طرح هاي معماري با پردازش موازي





معماري با دیسکهای مشترک

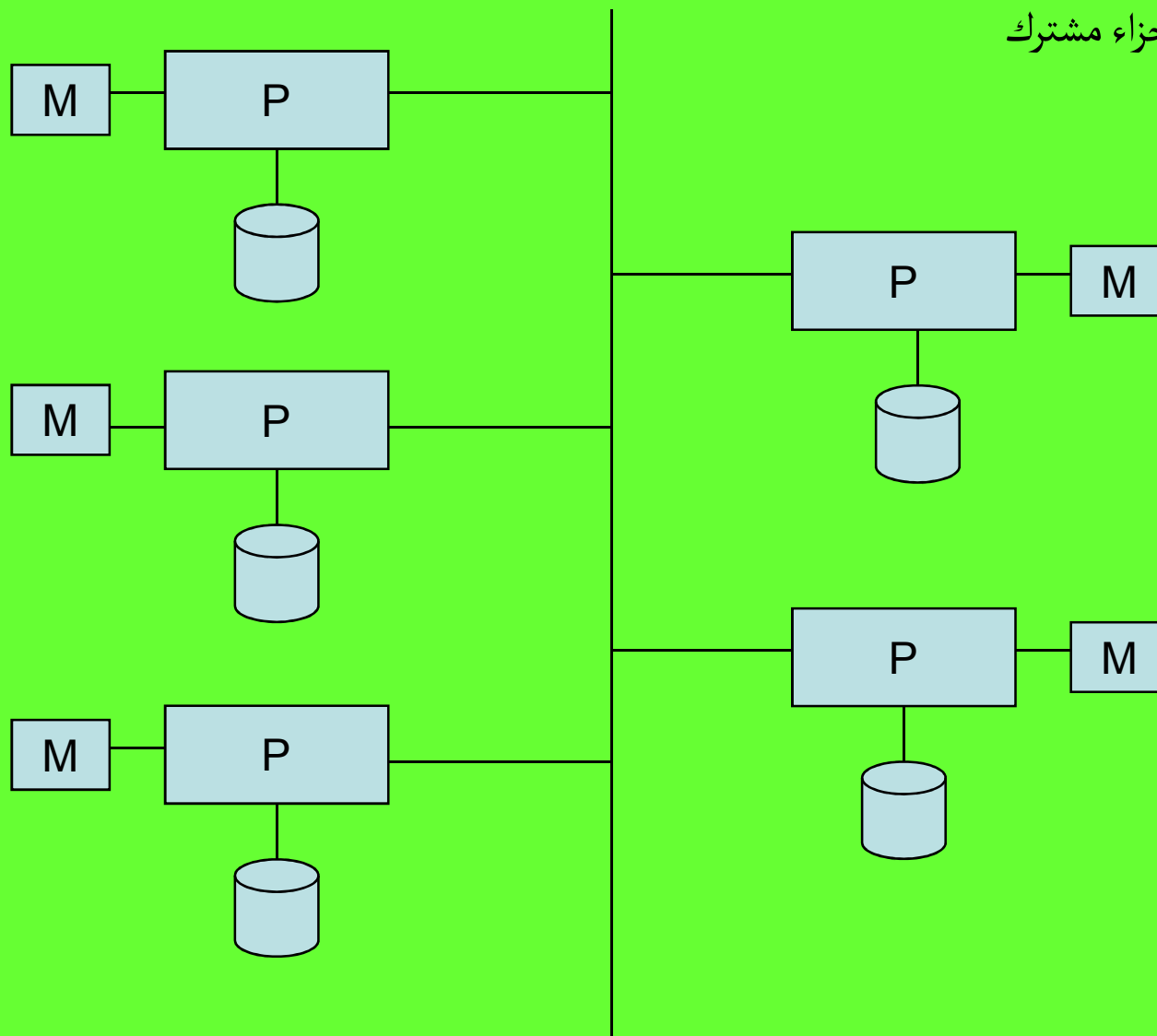
مزیت:

- 1- عدم بروز تنگنا در باسهای حافظه‌ای
- 2- تسهیل تحمل خرابی

عیب: دشواری در گسترش سیستم.

طرح هاي معماري با پردازش موازي

ج- معماری بی اجزاء مشترک





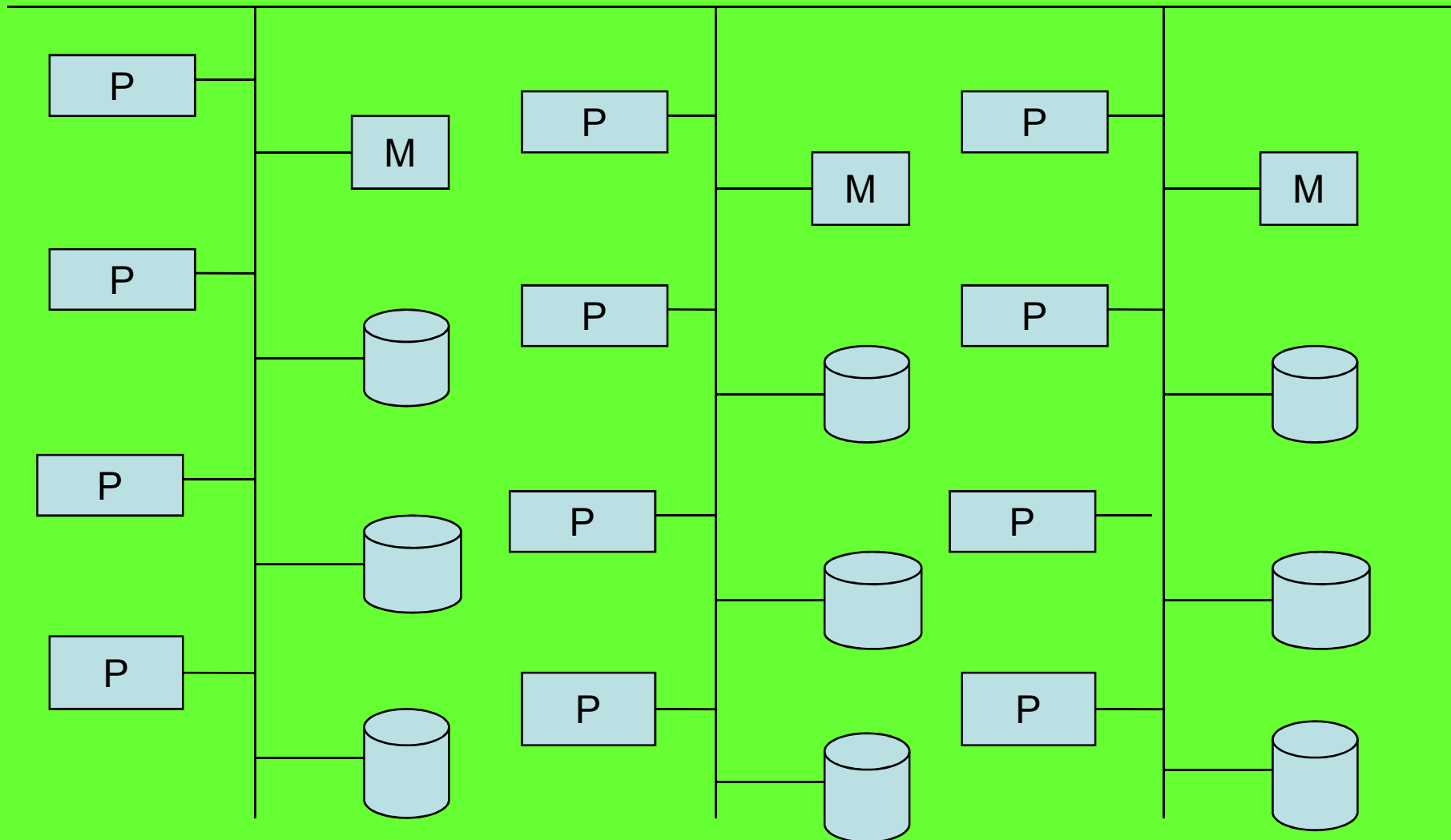
معماري بي اجزاء مشترك

مزیت: تسهیل گسترش

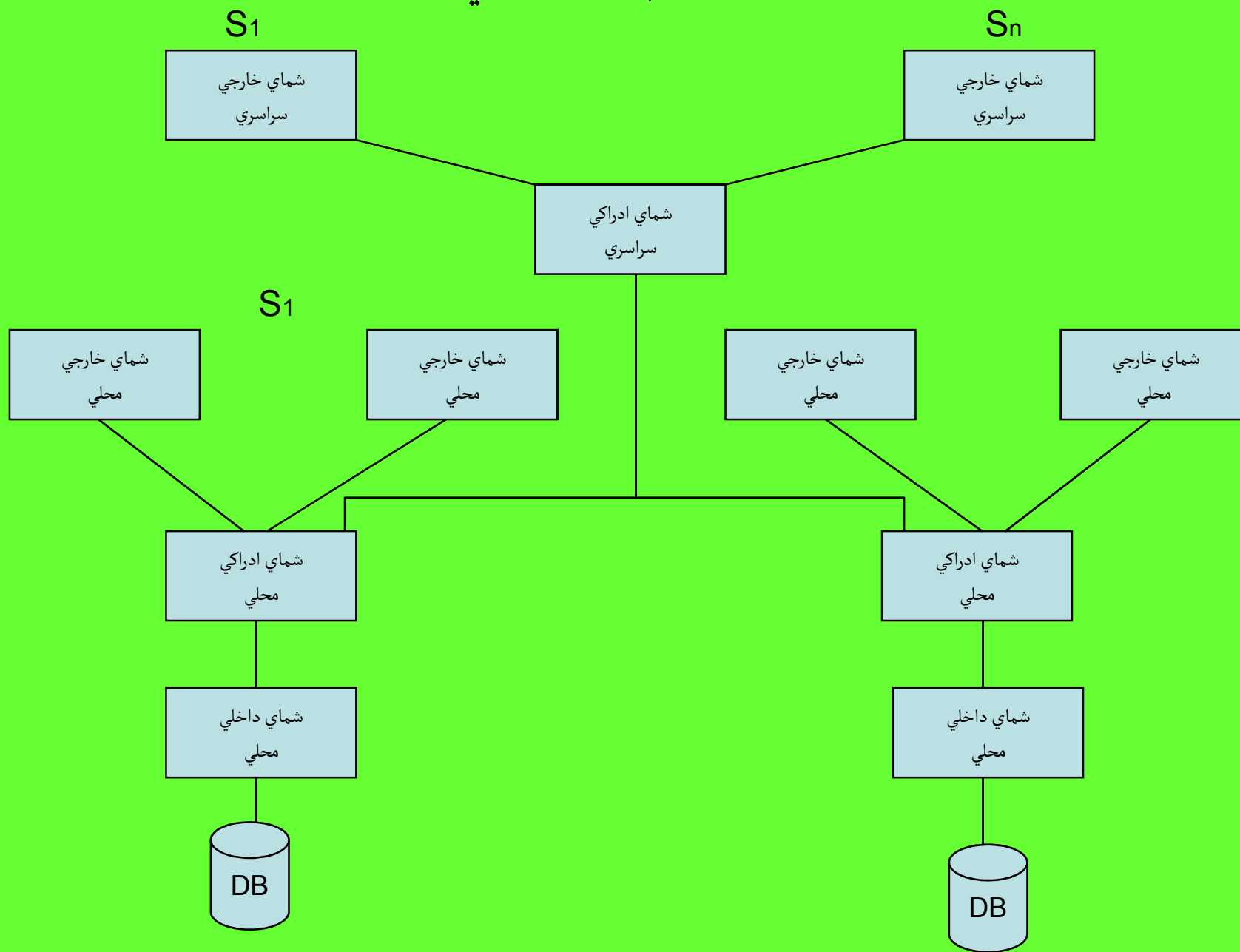
عیب: هزینه ارتباط و دستیابی های غیرمحملي زیاد است.

طرح هاي معماري با پردازش موازي

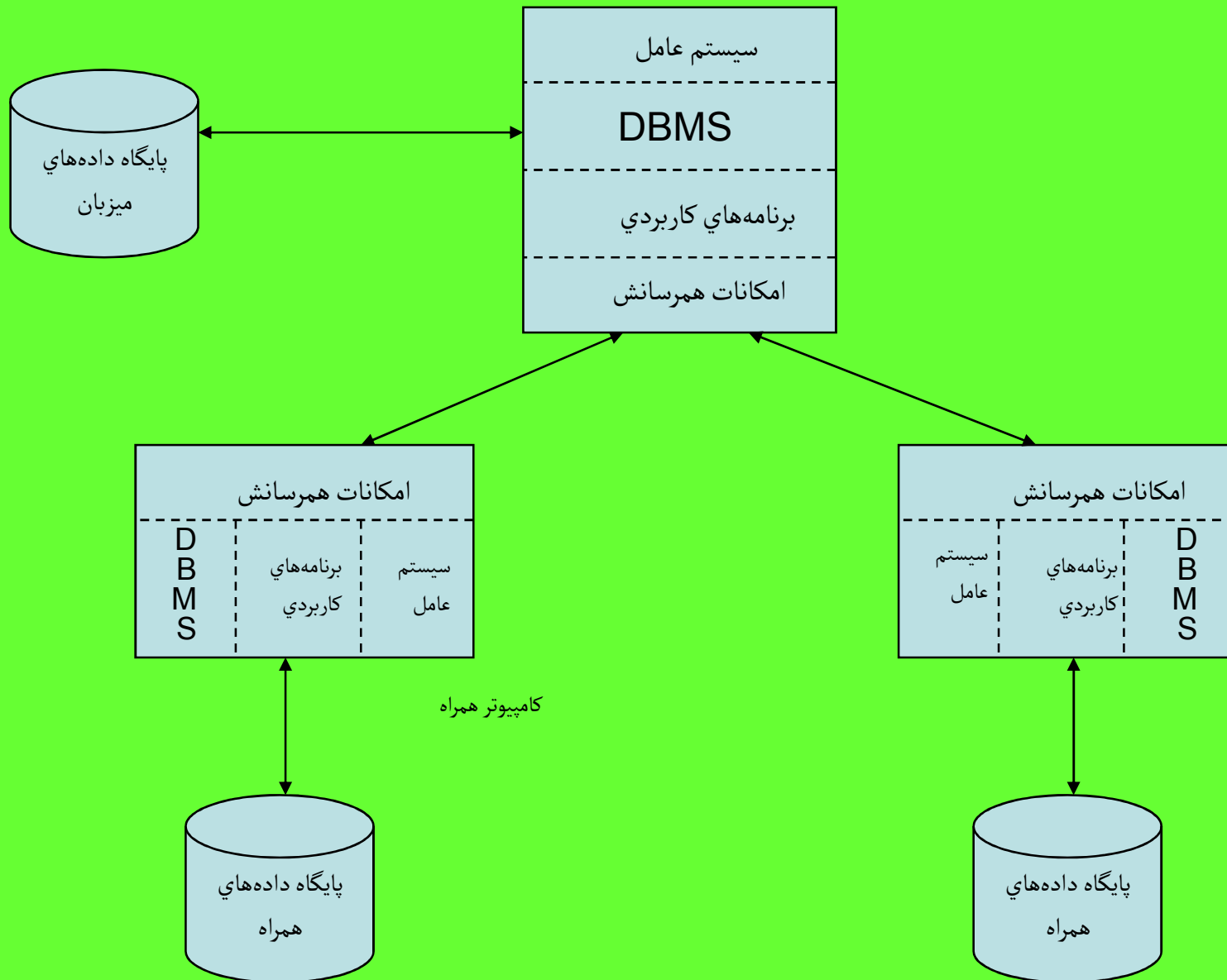
د- معماري سلسله مراتبي



معماري سيستم چندپايگاهي

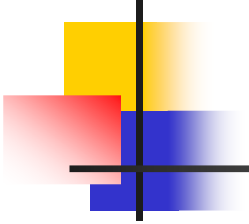


معماري سيستم پايگاههاي همراه



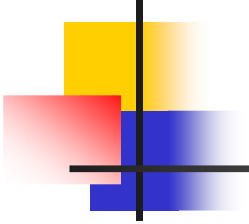
جلسه نهم

مدل رابطه‌ای



آنچه در این جلسه می خوانید:

- 1- مفاهیم اساسی مدل رابطه‌ای
- 2- بخشهای اساسی مدل داده‌ای
- 3- تعریف رابطه
- 4- تناظر بین مفاهیم رابطه‌ای و مفاهیم جدولی
- 5- ویژگیهای رابطه
- 6- انواع رابطه
- 7- میدان (دامنه)



آنچه در این جلسه می خوانید:

8- مزایای میدان

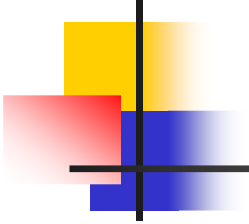
9- رابطه نرمال و غیر نرمال

10- دلیل لزوم نرمال بودن رابطه

11- معایب رابطه نرمال

12- مزایا و معایب رابطه غیرنرمال

13- انواع کلید در مدل رابطه ای



هدفهای کلی: آشنایی با مدل رابطه‌ای

هدفهای رفتاری: دانشجو در پایان این جلسه می‌تواند:

- مدل رابطه‌ای و بخشهای اساسی آن را توضیح دهد.
- رابطه را تعریف و ویژگیهای آن را بیان نماید.
- انواع رابطه را نام برده و توضیح دهد.
- روابط نرمال و غیرنرمال را شرح دهد.
- کلید و انواع آن را توضیح دهد.



مفاهيم اساسي مدل رابطه‌اي

مدل رابطه‌اي در سال 1970 توسط کاد ابداع شد.

مدل داده‌اي امکان‌ي است براي طراحي منطقي پايگاه داده‌ها ، تعريف و کنترل آن و نيز انجام عمليات در آن و امکان مي دهد تا اين هر سه عمل اساسي در محيط انتزاعي انجام شود. بنابر اين مي توان گفت که مدل داده‌اي تامين کننده محيط انتزاعي پايگاه داده هاست .



بخشهای اساسی مدل داده‌ای

- 1- بخش ساختاری
- 2- بخش عملیاتی (پردازشی)
- 3- بخش جامعیتی



بخش ساختاري ، نشان دهنده عناصر ساختاري مدل است که همان ساختار داده‌اي اصلي و مفاهيم مرتبط با آن است.

بخش عملياتي ، مجموعه امکاناتي است که به وسيله آنها عمليات مورد نظر کاربر انجام مي‌شود.

بخش جامعيتي ، از مجموعه‌اي از قواعد و محدوديتهاي جامعيتي تشکيل شده است که به وسيله آنها سيستم مدیریت پایگاه داده مي‌تواند صحت ، دقت و سازگاري داده‌ها را کنترل و تضمين کند.



تعریف رابطه

با فرض وجود n میدان D_1 تا D_n ، نه لزوما متمایز، رابطه R از دو قسمت تشکیل شده است:

1- سرآیند: مجموعه‌ای نامدار از n صفت به صورت $A_i:D_i$ که در آن هر

A_i نام یک صفت است و هر D_i نام میدان صفت

2- پیکر (بدنه): مجموعه‌ای است از m تاپل t به نحوی که t خود

مجموعه‌ای است از n عنصر هریک به صورت $A_i:v_i$ که در آن v_i مقداری

است از نوع میدان.

مقدار n را درجه (همان تعداد صفات) و مقدار m را کاردینالیتی رابطه می‌گویند



تناظر بين مفاهيم رابطه اي و مفاهيم جدولي

جدول زیر تناظر بين مفاهيم رابطه اي و مفاهيم جدولي را نشان مي دهد :

مفهوم تئوريک	مفهوم جدولي
رابطه	جدول
تاپل	سطر
صفت	ستون
ميدان	مجموعه مقادير ستون
درجه	تعداد ستونها
کاردیناليتي	تعداد سطرها



ویژگیهای رابطه

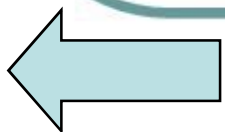
ویژگیهای رابطه عبارتند از:

- 1- رابطه تاپل تکراری ندارد
- 2- تاپلها نظم ندارند
- 3- صفات رابطه نظم مکانی ندارند
- 4- تمام صفات تکی مقداری هستند.



انواع رابطه

- 1- مبنا: استقلال وجودي دارد و از رابطه‌هاي ديگر مشتق نيست و داده‌هاي ذخيره‌شده متناظر دارد.
- 2- نامدار: با يك نام به سيستم معرفي مي‌شود.
- 3- ديد: نوعي رابطه نامدار كه مشتق از رابطه‌هاي ديگر است و ماهيتا مجازي است.
- 4- لحظه‌اي: نامدار و مشتق است ولي مجازي نيست.
- 5- مشتق: به كمك يك عبارت رابطه‌اي بر حسب رابطه‌هاي مبنا تعريف مي‌شود.



ادامه



انواع رابطه

6- عبارتي: از مجموعه‌اي از رابطه‌هاي نامدار و به وسيله يك عبارت رابطه‌اي به دست مي‌آيد.

7- نتيجه پرسش: بي‌نام و مشتق است كه حاصل اجراي يك پرسش مشخص است.

8- بينابيني: بي‌نام و مشتق است كه حاصل ارزيابي يك عبارت رابطه‌اي است كه درون يك عبارت بزرگتر جاي دارد

9- مشتق: رابطه‌اي عبارتي است كه به طور مستقيم و كارا ذخيره شده است.



میدان (دامنه)

از نظر ریاضی ، مجموعه‌ای است از مقادیر که يك یا بیش از يك صفت از آن مقدار می‌گیرند. در مدل رابطه‌ای ، این مجموعه:

- نامدار است.
- مقادیرش نوع مشخص دارند.
- مقادیرش فرمت مشخص دارند.



مزایای میدان

1. سبب ساده‌تر شدن و کوتاه‌تر شدن شمای پایگاه داده‌ها می‌شود.
2. تغییر در شمای پایگاه را تسهیل می‌کند.
3. امکانی است برای کنترل مقدری عملیات در پایگاه داده‌ها.
4. امکانی است برای کنترل معنایی درخواستها.
5. پاسخگویی به بعضی پرسشها را آسان می‌کند.
6. با استفاده از مفهوم میدان می‌توان امکانات و جنبه‌های موجود در سیستم مدیریت پایگاه داده‌های شیء‌گرا را به سیستم مدیریت پایگاه داده‌های رابطه‌ای افزود.



رابطه نرمال و غیر نرمال

- رابطه نرمال :

رابطه ای است که مقادیر هیچیک از میدانهایش ، یک مقدار رابطه ای با کاردینالیتی بزرگتر از یک نباشد

- رابطه غیر نرمال :

رابطه ای است که در آن مقادیر حداقل یک صفت ، خود مقادیر رابطه ای هستند (رابطه ای که حداقل یک صفت چند مقداری داشته باشد)



دلیل لزوم نرمال بودن رابطه

- 1- سادگی در نمایش ظاهری رابطه (جدول با سطرهای ساده‌تر)
- 2- سادگی دستورات DSL (بویژه DML و DDL)
- 3- سادگی در اجرای عملیات در پایگاه داده‌ها



معایب رابطه نرمال

- 1- بروز پدیده افزونگی که می تواند فیزیکی هم باشد.
- 2- طولانی تر شدن کلید رابطه.
- 3- عدم امکان نمایش داده های پیچیده.
- 4- دشواری در نمایش طبیعی مفهوم سلسله مراتب.
- 5- دشواری در نمایش مفهوم وراثت.



مزایا و معایب رابطه غیرنرمال

مزایا:

- 1- کاهش میزان افزونگی
- 2- کوتاه شدن کلید
- 3- امکان نمایش داده‌های پیچیده
- 4- دشواری کمتر در نمایش مفهوم سلسله‌مراتب و مفهوم وراثت
- 5- افزایش سرعت عملیاتی سیستم در بازیابی اطلاعات
- 6- عدم نیاز به نرمالترسازی رابطه

معایب:

- 1- پیچیدگی
- 2- عدم تقارن صفات



کلید در مدل رابطه‌ای

در مدل رابطه‌ای چند مفهوم در بحث کلید داریم که عبارتند از:

1. ابر کلید (super key)

2. کلید کاندید (candidate key)

3. کلید اصلی (primary key)

4. کلید دیگر (alternate key)

5. کلید خارجی (foreign key)



ابر کلید

تعریف- هر زیرمجموعه از مجموعه عنوان رابطه که یکتایی مقدار در گستره رابطه داشته باشد.



کلید کاندید

تعریف- هر زیرمجموعه از مجموعه عنوان رابطه که دو خاصیت

زیر را داشته باشد کلید کاندید رابطه است:

1- یکتایی مقدار

2- کاهش ناپذیری



کلید اصلي

تعريف- يکي از کلیدهاي کاندید رابطه که طراح انتخاب مي کند و به سيستم معرفي مي شود.

ضابطه هاي انتخاب:

- 1- از نظر کاربر ، شناسه معمول نوع موجوديت باشد.
- 2- طول کوتاهتر داشته باشد.



کلید دیگر (بدیل)

تعریف- هر کلید کاندید ، غیر از کلید اصلي ، کلید دیگر نام دارد.



کلید خارجی

تعریف- دورابطه و R_1 در نظر می گیریم. هر زیرمجموعه از صفات رابطه که هر مقدار معلومش R_2 مقدار از کلید کاندید برابر باشد، کلید خارجی در رابطه است.

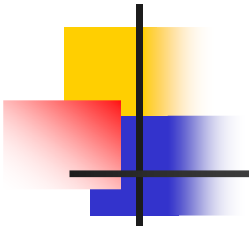
$$R_1 \quad R_2$$

کلید خارجی برای نمایش ارتباطات بین انواع موجودیتها به کار می رود.



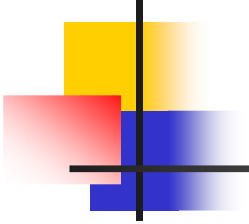
جلسه دهم

قواعد جامعیت پایگاه داده‌ها



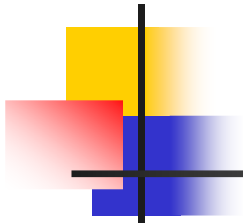
آنچه در این جلسه می خوانید:

- 1- جامعیت پایگاه داده‌ها
- 2- عوامل نقض جامعیت
- 3- انواع قواعد جامعیت
- 4- قواعد کاربری و انواع آن
- 5- متاقواعد و انواع آن
- 6- راههای اعمال قواعد جامعیت



آنچه در این جلسه می خوانید:

- 7- عملگرهای جبر رابطه‌ای
- 8- کاربردهای جبر رابطه‌ای
- 9- حساب رابطه‌ای
- 10- مزایا و معایب مدل رابطه‌ای
- 11- کاتالوگ در مدل رابطه‌ای
- 12- اشیائی که اطلاعات آنها در کاتالوگ نگهداری می‌شود



هدفهاي كلي: آشنائي با قواعد جامعيت پايگاه داده‌ها

هدفهاي رفتاري: دانشجو در پايان اين جلسه مي‌تواند:

- جامعيت پايگاه داده‌ها و عوامل نقض آن را توضيح دهد.
- انواع قواعد جامعيت را نام برده و راههاي اعمال آن را شرح دهد.
- جبر رابطه‌اي و حساب رابطه‌اي را توضيح دهد.
- مزايای و معايب مدل رابطه‌اي را نام ببرد.



جامعیت پایگاه داده‌ها

یعنی: صحت ، دقت و سازگاری داده‌های ذخیره‌شده
در پایگاه در تمام لحظات



عواملی که سبب نقض جامعیت می شوند:

- اشتباه در برنامه های کاربردی
- اشتباه در وارد کردن داده ها
- وجود افزونگی کنترل نشده
- توارد تراکنشها به گونه ای که داده نامعتبر ایجاد شود.
- خرابیهای سخت افزاری و نرم افزاری



انواع قواعد جامعیت

1- قواعد کاربري (قواعد خاص)

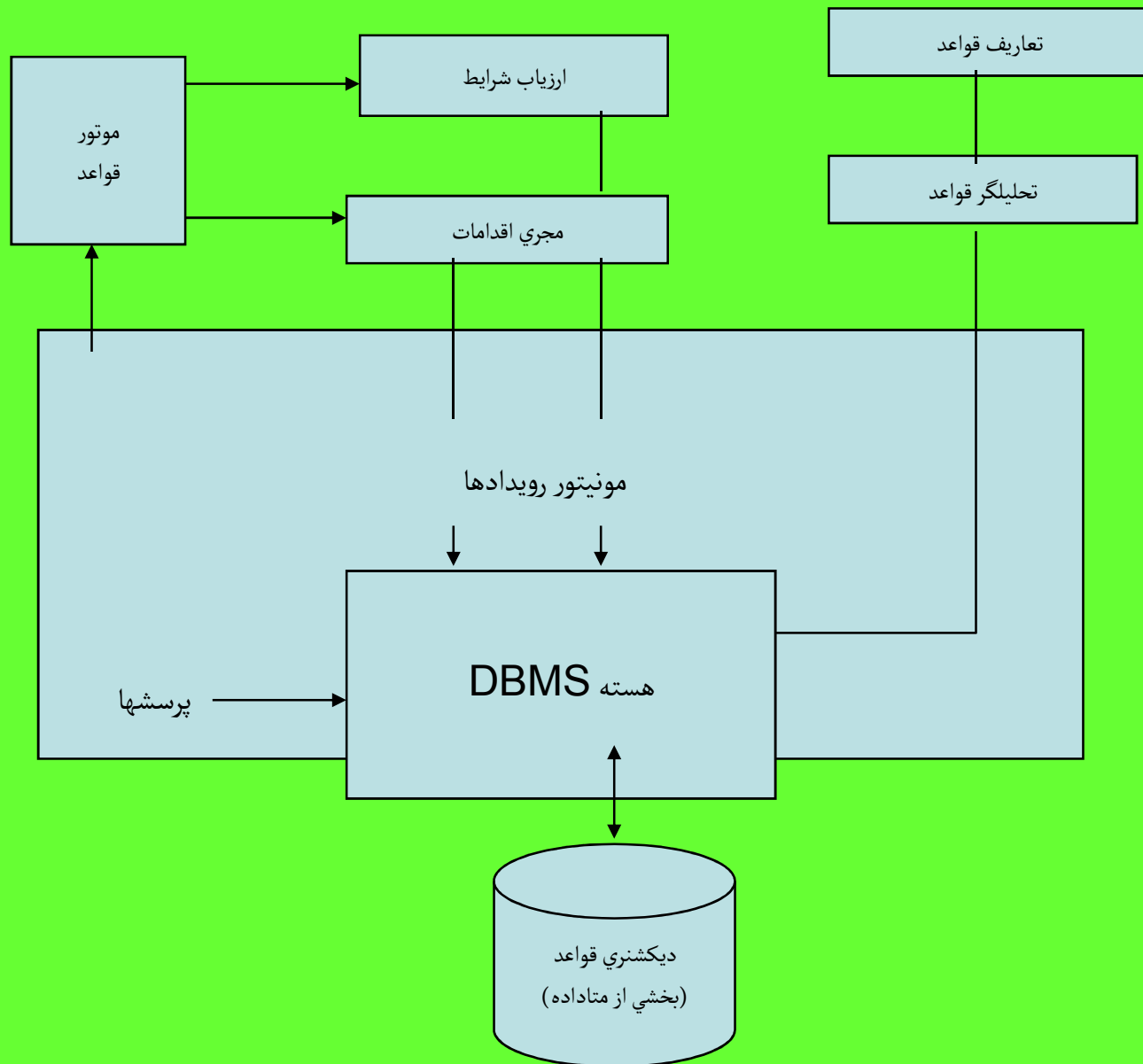
2- متا قواعد (قواعد عام)



1- قواعد کاربري

قواعدي هستند که توسط کاربر مجاز تعريف مي شوند. وابسته به داده هاي خرد جهان واقع هستند ، به اين معنا که در مورد يك پايگاه داده خاص مطرح مي شوند و عموميت ندارند. به اين قواعد ، قواعد محيطي يا وابسته به داده و يا محدوديتهاي جامعيت معنایي مي گويند.

نمای ساده شده یک DBMS فعال





انواع قواعد کاربري در مدل رابطه‌اي

1- محدوديت ميداني: ناظر بر يك ميدان است و مقادير مجاز آن را مشخص مي‌کند.

2- محدوديت صفتي (ستوني): ناظر به يك صفت است و بيان‌کننده نوع آن صفت است.

3- محدوديت رابطه‌اي: ناظر بر يك رابطه است و مقادير مجاز يك تغيير رابطه‌اي را مشخص مي‌کند.

4- محدوديت پايدگهي: ناظر بر دو يا بيش از دو متغير رابطه‌اي است و به نحوي آنها را به يکديگر مرتبط مي‌کند.



2- متا قواعد

قواعدی هستند که باید توسط هر سیستم رابطه‌ای در هر پایگاه داده رابطه‌ای اعمال شود ، ناوابسته به داده‌های خاص هستند و عمومیت دارند.



انواع مذاقواعد

1- قاعده (جامعيت موجوديتي)

2- قاعده (جامعيت ارجاعي)

 C_1

قاعده

ناظر به کلید اصلي است و چنین است:
هیچ جزء تشکیل دهنده کلید اصلي نمی تواند هیچ مقدار داشته باشد.



C_2 قاعده

ناظر بر کلید خارجی است و چنین است:

اگر صفت خاصه A_i در رابطه R_2 کلید خارجی باشد در این صورت:

A_i در R_2 می تواند هیچ مقدار داشته باشد یا اینکه باید حتما مقداری داشته باشد که در رابطه مرجع R_1 وجود دارد. به عبارت دیگر مقدار کلید خارجی يك رابطه نمی تواند در رابطه مرجع وجود داشته باشد.



به قواعد C_1 و C_2 محدودیتهای ساختاری می گویند

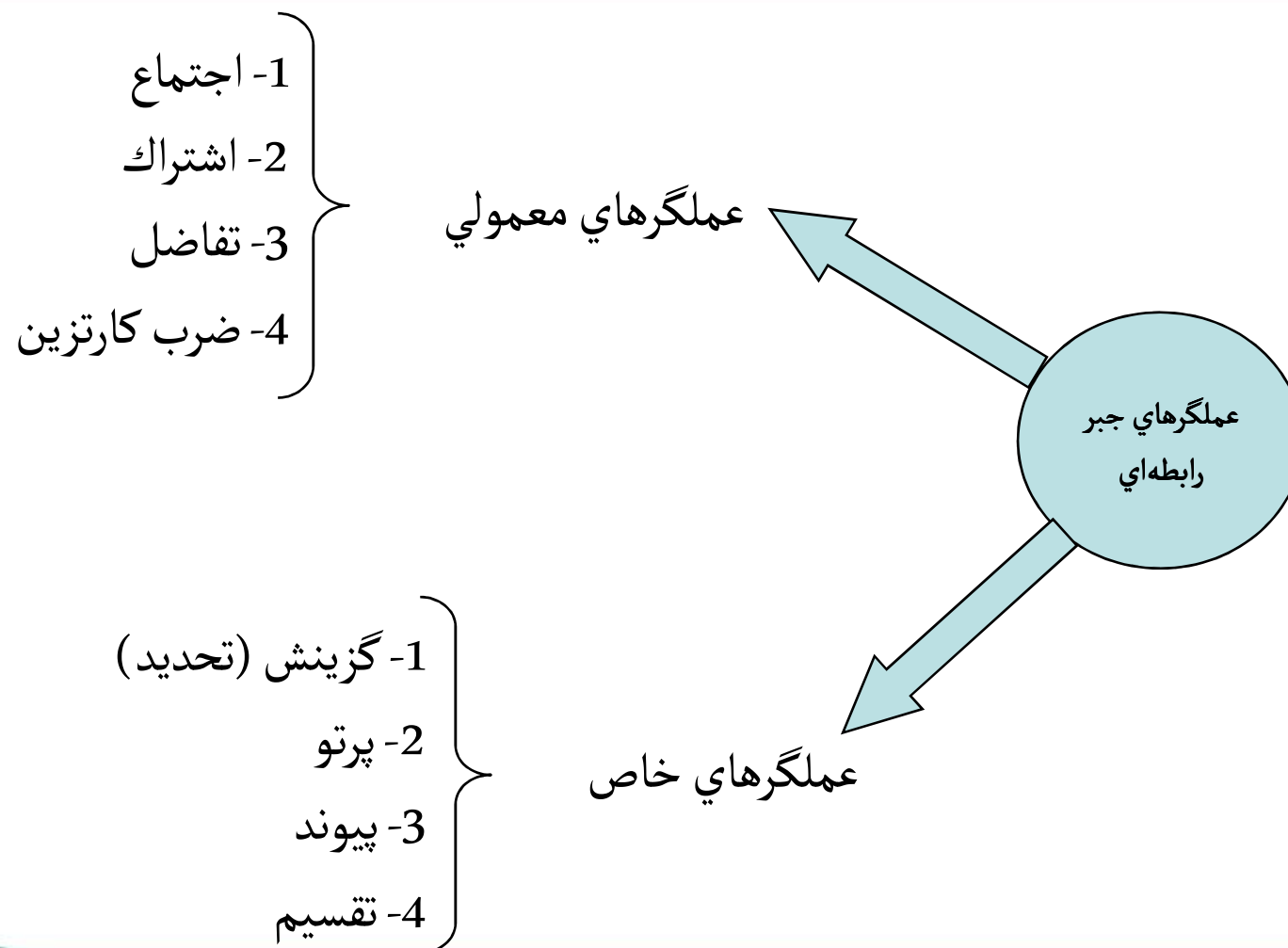
- وابستگی های تابعی
- وابستگی های چندمقداری
- وابستگی های شمول
- وابستگی زمانی
- وابستگی برابری

محدودیتهای ناساختاری



راههای اعمال قواعد جامعیت

- 1- معرفي کليد اصلي
- 2- اعلام هيچ مقدارناپذيري صفت
- 3- معرفي کليد خارجي
- 4- اعلان محدوديتهاي مورد نظر ، در شمائي پايگاه داده‌ها
- 5- نوشتن رهانا
- 6- اعلان محدوديتها با استفاده از مکانيسم اظهار





عملگر گزینش تاپلهایی از یک رابطه را گزینش می کند (سطرهایی از جدول را).
این عملگر تکه عملوندي است و به صورت زیر نوشته می شود:

R WHERE cond(s)

* cond(s) شرط یا شرایط گزینش را مشخص می کند.



علمگر پرتو ، مقادير صفت (صفاتي) از يك رابطه را به دست مي دهد. تڭ عملوندي
بوده و چنين نوشته مي شود:

PROJECT R OVER($A_1, A_2, \dots, A_i$)



عملگر پیوند دو عملوندي است و چنین نوشته مي شود:

$R_1 \text{ JOIN cond}(s) R_2$

که در آن $\text{cond}(s)$ چنین است:

$R_1.A_i \text{ theta } R_2.B_i$

A_i يك صفت از رابطه R_1 و B_i يك صفت از رابطه R_2 است که از يك ميدان مقدار مي گیرند و باید ناهمنام باشند.



کاربردهای جبر رابطه‌ای

- 1- بازیابی داده‌ها
- 2- ذخیره‌سازی داده‌ها
- 3- تعریف انواع رابطه‌های مشتق
- 4- تعریف قواعد برای کنترل پایگاه داده‌ها
- 5- تعریف داده‌ها به عنوان حیطة بعضی عملیات کنترل
همروندی تراکنشها
- 6- ضابطه تشخیص کامل بودن زبانهای رابطه‌ای



حساب رابطه‌اي

حساب رابطه‌اي ، با جبر رابطه‌اي منطقاً معادل است ، يعني براي هر عبارت جبر رابطه‌اي ، يك عبارت معادل در حساب رابطه‌اي وجود دارد و برعكس. تفاوت آنها اين است كه جبر رابطه‌اي ، دستوري است ، اما حساب رابطه‌اي توصيفي است.



- حساب تاپلي: در این حساب مفهوم مهمي به نام متغير تاپلي

وجود دارد که تنها مقادير مجازش ، تاپلهای رابطه هستند.

- حساب میداني: در این حساب ، متغير میداني وجود دارد که از

يك ميدان مقدار مي گيرد. در این حساب يك شرط اضافي به نام

شرط عضویت وجود دارد.

حساب رابطه‌اي



در حساب رابطه‌ای تاپلی دو سور وجود دارد:

1- سور وجودی: به صورت نوشته می‌شود، $\exists T(f)$ به این معنا که حداقل یک مقدار

برای متغیر T وجود دارد به نحوی که f به “درست” ارزیابی شود.

2- سور همگانی: به صورت نوشته می‌شود. یعنی به ازاء تمام مقادیر متغیر

T ، f به “درست” ارزیابی می‌شود. $\forall T(f)$



مزایای مدل رابطه‌ای

- 1- نمایش ساده
- 2- مبنای ریاضی دارد.
- 3- تأمین‌کننده استقلال داده‌ای است.
- 4- یک عنصر ساختاری اساسی به نام رابطه دارد.
- 5- کامل است.
- 6- عدم نیاز به مفهوم کلاسیک نظم.
- 7- برای به‌ترسازی طراحی ، ابزار طراحی بهتر دارد.
- .
- .
- .



معایب مدل رابطه‌ای

- 1- دشواری در نمایش داده‌های پیچیده
- 2- دشواری در نمایش مفاهیمی مانند وراثت ، تعمیم و ...
- 3- نداشتن قواعد جامعیت ذاتی
- 4- عدم امکان مدلسازی رفتار اشیاء
- 5- فقدان اکمال برنامه‌سازی و اکمال محاسباتی
- 6- دشواری در یافتن کلید کاندید
- 7- جدا بودن داده‌ها از روشها
- 8- مشکلات ناشی از وجود مفهوم هیچ مقدار
- .
- .



کاتالوگ در مدل رابطه‌ای

کاتالوگ در مدل رابطه‌ای ، تعدادی رابطه است: یک پایگاه رابطه‌ای که خود سیستم ایجاد می‌کند

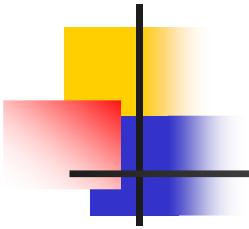


در کاتالوگ اطلاعاتی در مورد اشیاء ذیل نگهداری می شود:

- میدان
- رابطه
- صفت
- دید
- قواعد جامعیت
- توابع و رویه های تعریف شده توسط کاربر
- رهانهای تعریف شده توسط کاربر
- کاربر
- شاخص و ساختارهای مرتبط با آن
- ضوابط ایمنی
- پرسشها
- برنامه های کاربردی

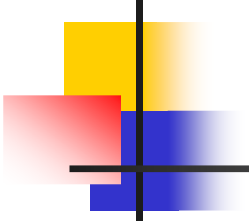
جلسه یازدهم

زبان SQL



آنچه در این جلسه می خوانید:

- 1- انواع زبانهای رابطه ای
- 2- امکانات مهم زبان **SQL**
- 3- دستورات تعریف داده ها ، دستور ایجاد و حذف جدول و دید
- 4- دستور مجازشماری
- 5- دستورات پردازش داده ها
- 6- توابع جمعی (گروهی)



آنچه در این جلسه می خوانید:

7- امکان LIKE و NOTLIKE

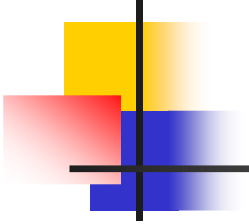
8- آزمون تست وجود هیچمقدار در یک ستون

9- امکان UNION و UNION ALL

10- امکان GROUP BY: (گروه بندی)

11- امکان HAVING

12- امکان BETWEEN



هدفهاي كلي: آشنائي با زبان SQL

هدفهاي رفتاري: دانشجو در پايان اين جلسه مي تواند:

- امکانات زبان SQL را بيان کند.
- انواع دستورات تعريف داده ها و پردازش داده ها ياد گرفته و مي تواند استفاده کند.



آشنایی با یک زبان رابطه ای : SQL

انواع زبانهای رابطه ای

- SQUARE •
- SEQUEL •
- SQL •
- QUEL •
- QBE •
- DATALOG •



امکانات مهم زبان SQL

- دستورات تعریف داده‌ها
- دستورات مجازشماری
- دستورات پردازش داده‌ها
- دستورات پردازش داده‌ها به طور ادغام‌شدنی
- دستورات نوشتن ماژول و رویه
- دستورات کنترل جامعیت
- دستورات کنترل تراکنشها



1. دستورات تعریف داده‌ها

1-1- تعریف شما :

CREATE SCHEMA

AUTHORIZATION USER

**{base – table definition , view definition ,
grant-operation }**



1. دستورات تعریف داده‌ها

1-2- انواع داده‌ای:

CHARACTER[(length)]

INTEGER

DECIMAL [(precision[,scale])]

SMALLINT

DOUBLE PRECISION

REAL

FLOAT [(precision)]

NUMERIC [(precision[,scale])]



1. دستورات تعریف داده‌ها

1-3- دستور ایجاد جدول :

CREATE SCHEMA
AUTHORIZATION ...
CREATE TABLE ...



1. دستورات تعریف داده‌ها

1-4- دستور حذف جدول:

DROP TABLE table-name



1. دستورات تعریف داده‌ها

1-5- دستور ایجاد و حذف دید:

CREATE VIEW ...

DROP VIEW ...



1. دستورات تعريف داده‌ها

1-6- دستور مجاز شماری:

GRANT , REVOKE

با این دستور حق انجام يك يا بیش از يك عمل به کاربر (کاربرانی) داده می‌شود.



2- دستورات پردازش داده ها

2-1- دستور بازیابی (SELECT):

SELECT [ALL | DISTINCT] item(s)-list

FROM table(s)-name

[WHERE condition(s)]

[GROUP BY column(s)]

[HAVING conditions(s)]



توابع جمعي (گروهي)

- **COUNT:** تعداد مقادير اسكالر را به دست مي دهد.
- **SUM:** حاصل جمع مقادير اسكالر در يك ستون را به دست مي دهد.
- **AVG:** ميانه مقادير اسكالر در يك ستون را به دست مي دهد.
- **MAX:** بيشترين مقدار اسكالر در يك ستون را به دست مي دهد.
- **MIN:** كمترين مقدار اسكالر در يك ستون را به دست مي دهد.



يك مثال در مورد استفاده از توابع جمعي

```
SELECT MIN(GRADE) , MAX(GRADE)
FROM STCOT
WHERE TR='2'
AND
YRYS='78-79'
AND
COTD='COM222';
```

بالاترين و پايين ترين نمره در درس COM222 در ترم دوم 79-80 را بازيايي مي کند.



امكان LIKE و NOTLIKE

با این دو امکان می‌توان داده‌های مورد نظر را با دادن یک رشته کاراکتری به عنوان نشانوند جستجو و بیان شرایط مورد نظر، بازیابی کرد.



مثال: مشخصات استادانی را بدهید که نام آنها با **AR** شروع شده باشد.

```
SELECT *  
FROM PROF  
WHERE PRNAME LIKE 'AR%';
```



آزمون تست وجود هیچ مقدار در يك ستون

مي توان با امکان **IS NULL** وجود هیچ مقدار در يك ستون را تست کرد.



مثال: شماره دانشجویانی را بدهید که نمره آنها در درس **SOC333** در ترم دوم 78-79 هنوز اعلام نشده است ؟

```
SELECT STID
FROM STCOT
WHERE COID='SOC333'
AND
TR='2'
AND
YRYS='78-79'
AND
GRADE IS NULL
```



امکان UNION و UNION ALL

با این امکان ، می توان عملکرد عملگر **UNION** جبری را برنامه سازی کرد. حاصل این عملکرد ، جدولی است که سطرهای تکراری ندارد. اگر از **ALL** گزیدار استفاده شود ، سیستم دیگر سطرهای تکراری را در صورت وجود ، حذف نمی کند.



امكان **GROUP BY**: (گروه بندي)

با این امکان می‌توان سطرهای جدول را حسب مقادیر یک ستون ساده گروه‌بندی کرد به نحوی که در هر گروه ، مقدار آن ستون یکسان باشد.



مثال: جدول STCOT را برحسب مقادیر ستون COID منطقاً گروه‌بندی کنید.

```
SELECT COID, AVG(GRADE) AS  
AVGGR  
  
FROM STCOT  
  
GROUP BY COID;
```



امکان **HAVING**

با این امکان می‌توان شرط (شرایطی) ناظر به گروه سطرها اعلان کرد. نقش این امکان در واقع همان نقش **WHERE** در سطر است. توجه داشته باشید که **HAVING** همیشه با **GROUP BY** می‌آید.



مثال: شماره درسهایی را بدهید که در ترم دوم 78-79 کمتر از 10 دانشجو در آنها ثبت نام کرده باشند.

```
SELECT STCOT.COID  
FROM STCOT  
WHERE TR='2'  
AND  
YRYR='78-79'  
GROUP BY COID  
HAVING COUNT(*)<10
```



BETWEEN امکان

شکل کلی این امکان چنین است:

Scalar-expression [NOT] BETWEEN Scalar-expression AND Scalar-expression



مثال: شماره دانشجویانی را بدهید که نمره آنها در درس **HIS444** در ترم اول 77-78 بین 15 و 19 باشد.

```
SELECT STCOT.STID
```

```
FROM STCOT
```

```
WHERE TR='1'
```

```
AND
```

```
YRYSR='78'79'
```

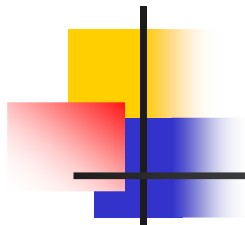
```
AND
```

```
GRADE BETWEEN '15' AND '19'
```

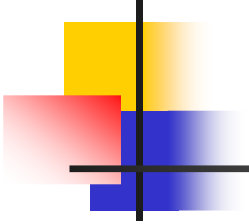
جلسه دوازدهم

ادامه زبان SQL

آنچه در این جلسه می خوانید:



- 1- دستورات عملیات ذخیره سازی (درج ، حذف و بهنگام سازی)
- 2- جمع بندی در مورد امکانات نسخه **SQL1**
- 3- دستوراتی که در نسخه **SQL2** تغییر کردند یا به آن اضافه شدند
- 4- امکانات جامعیتی
- 5- امکانات ایمنی
- 6- امتیازها



آنچه در این جلسه می خوانید:

7- دستوراتی که در نسخه **SQL3** تغییر کردند یا به آن اضافه شدند

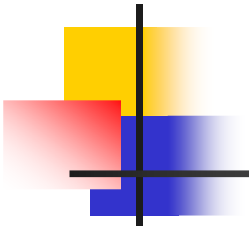
8- نوع داده مجرد

9- تعریف رهانا و مزایا و موارد کاربرد آن

10- تراکنش

11- طرحهای اجرای تراکنشها

12- تکنیکهای کنترل همروندی



هدفهاي كلي: آشنائي با زبان SQL

هدفهاي رفتاري: دانشجو در پايان اين جلسه مي تواند:

- از دستورات ذخيره سازي استفاده کند.
- امکانات اضافه شده در نسخه SQL2 را بيان کند.
- امکانات اضافه شده در نسخه SQL3 را بيان کند.



2- دستورات پردازش داده ها

2-2- دستورات عملیات ذخیره سازی :

برای انجام عملیات سه گانه ذخیره سازی سه دستور **DELETE** ،
INSERT و **UPDATE** وجود دارد.



UPDATE دستور

شکل کلی این دستور چنین است:

UPDATE table-name

SET assignment-commalist

[WHERE Condition(s)]



مثال: تعداد واحد درسهایی عملی را یک واحد کاهش دهید.

```
UPDATE COT  
SET CREDIT=CREDIT-1  
WHERE COTYPE='p';
```



DELETE دستور

شکل کلی این دستور چنین است:

DELETE

FROM table-name

[WHERE Cond(s)]



مثال: درسهایی دانشجویی با شماره 78110555 را در ترم دوم سال 78-79 حذف کنید

DELETE

FROM STCOT

WHERE STID='78110555'

AND

TR='2'

AND

YRYR='78-79';



دستور INSERT

این دستور دو شکل کلی دارد:

```
INSERT INTO table-name  
Values(one row);
```

```
INSERT INTO table-name  
Subquery;
```

در شکل دوم ، تعدادی سطر در جدول درج می شوند.



مثال: اطلاعات درسي زير را در جدول درج نماييد.

<78110888,COM888,2,78-79,12>

INSERT INTO STCOT

VALUES <'78110888', 'COM888', 2, 78-79, 12>;



جمع‌بندی در مورد امکانات نسخه SQL1

- 1- نارویه‌ای است.
- 2- دارای کمال ساختاری است.
- 3- زبان استاندارد سیستم‌های رابطه‌ای موجود است.
- 4- تمام انواع داده‌ای ساده را دارد.
- 5- عملگرهای بسیار قوی دارد.
- 6- یادگیری آن ساده است.
- 7- استقلال داده‌ای را تأمین می‌کند.
- 8- هم به صورت مستقل و هم به صورت ادغام‌شده قابل استفاده است.



دستوراتي که در نسخه SQL2 تغییر کردند یا به آن اضافه شدند:

1. دستور تعریف داده‌ها
2. دستور تعریف میدان
3. انواع داده‌ای
4. دستور تغییر میدان
5. دستور حذف میدان
6. دستور ایجاد جدول
7. دستور تغییر جدول
8. دستور حذف جدول
9. تعریف جدول موقت
10. دستور بازیابی
11. امکانات جامعیتی
12. امکانات ایمنی
13. SQL پویا



امکانات جامعیتی

کاربر می تواند زمان واریسی جامعیت پایگاه را (براساس محدودیتها و قواعد داده شده) خود مشخص کند. می تواند درخواست کند که این واریسی بلافاصله باشد یا با تاخیر. برای این منظور از امکان زیر استفاده می شود:

```
SET CONSTRAINTS (constraint-name)  
{DEFERRED | IMMEDIATE}
```



امکانات ایمنی

کاربر می تواند امتیاز دستیابی به اشیاء تعریف شده در شما را به منظور انجام عمل
مشخص ، به کاربر دیگری بدهد.

برای اعطای امتیاز از دستور **GRANT** و برای لغو آن از دستور **REVOKE**
استفاده می شود.



امتیازها

امتیازهایی که یک کاربر می تواند به یک کاربر دیگر بدهد:

- **USAGE:** اجازه استفاده از یک میدان
- **SELECT:** اجازه دستیابی به تمام ستونهای یک جدول
- **INSERT(X):** اجازه درج یک مقدار در ستون **X** از یک جدول
- **INSERT:** اجازه درج تمام ستونهای یک جدول
- **UPDATE(X):** اجازه بهنگام سازی ستون **X** از یک جدول
- **UPDATE:** اجازه بهنگام سازی تمام ستونهای یک جدول
- **DELETE:** اجازه حذف سطر(ها) از یک جدول
- **REFERENCES(X):** اجازه ارجاع به ستون **X** از یک جدول در تعریف محدودیتهای جامعیتی



دستوراتی که در نسخه SQL3 تغییر کردند یا به آن اضافه شدند:

1. تعریف نوع داده مجرد
2. تعریف زیرنوع
3. امکان تعریف زیرجدول و زبرجدول
4. امکان ارتزبری ستونهای جدول جدید از ستونهای جدول نامدار موجود
5. رویه ای شدن زبان
6. رهانا
7. نوشتن رویه
8. SQL ادغام شده
9. واسط درخواست فراخوان
10. دستورات کنترل تراکنشها



نوع داده مجرد

شاید مهمترین قابلیت که در زمینه انواع داده‌ای در SQL3 افزوده شد، امکان تعریف نوع داده مجرد (ADT) باشد. با این نوع داده می‌توان مجموعه‌ای از صفات و روالها را تعریف کرد. نوع داده مجرد نمونه‌هایی دارد و هر نمونه هم به نوبه خود، همه صفات و روالهای تعریف شده را دارد.



ساخته‌های زبانی افزوده شده به SQL3 ، که در تعریف توابع و رویه‌ها استفاده می‌شود:

- احکام مرکب (تودرتو)
- احکام حلقه گردانی
- احکام شرطی
- احکام انتساب
- احکام اعلان اشتباهات در برنامه



رهانا

تعریف- قاعده (محدودیت) یا قواعدی است که قبل یا بعد از بروز یک رویداد در پایگاه داده‌ها (معمولاً یک عمل تغییردهنده داده‌ها) باید اعمال شود. این قاعده در سطح برنامه‌سازی ، به صورت یک رویه از پیش تعریف شده است که به طور شرطی یا غیرشرطی ، قبل یا بعد از انجام یک عمل در پایگاه داده‌ها ، به طور اتوماتیک اجرا می‌شود.



رهانا

مزایا

1. اجرای آن تحت کنترل متمرکز سیستم و نظارت مدیر پایگاه داده‌ها است.
2. با وجود آن دیگر نیازی نیست که محدودیتهای مورد نظر در هر برنامه کاربردی اعمال شوند و کنترل‌های لازم انجام شود.
3. برای معماری **C/S DB** بسیار مناسب است.



کاربرد رهانا

1. اعمال قواعد جامعیت.
2. اعمال قواعد ایمنی مبتنی بر مقادیر
3. درج رکوردهای ثبت عملیات در فایل ثبت تراکنشها
4. تولید نسخه‌ای از داده‌های ذخیره‌شده در یک سایت دیگر.
5. در انتشار اتوماتیک عملیات در پایگاه داده‌های توزیع‌شده.
6. در بهنگام‌سازی دیدها در سیستم‌های رابطه‌ای



تراکنش

تعریف: به بیان ساده عملی است که تغییری در پایگاه داده‌ها ایجاد کند. این عمل که طبقاً به صورت یک برنامه یا بخشی از یک برنامه است، یا باید به تمامی اجرا شود و یا اصلاً اجرا نشود و می‌گوییم که تراکنش خاصیت تجزیه‌ناپذیری دارد.

خواص

1. تجزیه‌ناپذیری
2. سازگاری
3. جدایی (انفراد)
4. مانایی (دوام)



طرحهای اجرایی تراکنشها

- طرح اجرایی متوالی
- طرح اجرایی همروند



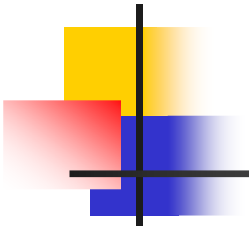
تکنیکهای کنترل همروندی

- تکنیک قفل گذاری
- تکنیک زمانمهر
- تکنیک چند نسخه سازی
- تکنیک تایید (تصدیق)



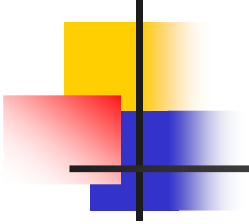
جلسه سیزدهم

دید در مدل رابطه‌ای



آنچه در این جلسه می خوانید:

- 1- دیدهای رابطه‌ای
- 2- دید در SQL
- 3- مزایا و معایب دید
- 4- عملیات در دیدهای رابطه‌ای
- 5- انواع دید از نظر پذیرش عملیات ذخیره‌سازی
- 6- انواع دیدهای پذیرا
- 7- انواع دیدهای ناپذیرا
- 8- ویژگیهای دیدهای قابل بهنگام‌سازی



هدفهاي كلي: آشنائي با ديد در مدل رابطه‌اي

هدفهاي رفتاري: دانشجو در پايان اين جلسه مي‌تواند:

- ديد را تعريف کند و دستورات **SQL** مرتبت با آن را بکار برد.
- مزايای استفاده از ديد را برشمارد.
- عمليات قابل انجام روي ديد را شرح دهد.
- انواع ديد از لحاظ پذير و ناپذير را نام ببرد.



دیدهای رابطه‌ای

دید در مدل رابطه‌ای نوعی رابطه است ، بنابراین می‌توان آن را به
كمك يك عبارت جبر رابطه‌ای یا حساب رابطه‌ای به شکل زیر
تعریف کرد:

View name= Relational expression



دید در SQL

دستور ایجاد دید

```
CREATE VIEW Viewname [(Column-name(s))]  
AS Subquery
```

-
-
-

```
[WITH [CASCADE|LOCAL] CHECK OPTION];
```



مثال

```
CREATE VIEW MAPHSTUD( STNUM, STLEV, STAREA)  
AS SELECT STID, STDEG, STMJR  
FROM STT  
WHERE STMJR='Math' OR STMJR='Phys'  
WITH LOCAL CHECK OPTION;
```



دید در SQL

دستور حذف دید

```
DROP VIEW Viewname {restrict | cascade}
```



مثال:

DROP VIEW MAPHSTUD CASCADE;



مزایای دید

1. تامین کننده محیط انتزاعی برای کاربران سطح خارجی
2. تامین کننده پویایی بالا در تعریف پایگاه توسط کاربر
3. تسهیل کننده واسط کاربر برنامه ساز با پایگاه
4. امکانی است برای کوتاه نویسی یا ماکرونویسی پرسشها
5. تامین کننده اشتراك داده ای
6. تامین کننده نوعی مکانیسم خودکار ایمنی داده ها
7. تامین کننده استقلال داده ای فیزیکی و منطقی
8. امکان تعریف شیء با اندازه های مختلف



معایب دید

1. ایجاد فزونکاری در سیستم برای انجام تبدیل خارجی/ادراکی و احیانا خارجی/خارجی
2. عدم امکان انجام عملیات ذخیره سازی در بسیاری از گونه های دید و در نتیجه ایجاد محدودیت برای کاربر



عملیات در دیدهای رابطه‌ای

بازیابی از دید رابطه‌ای

چون دید خود نوعی رابطه است ، پس برای بازیابی از دید هم يك عبارت جبري يا حسابي مي‌نویسیم



مثال بازیابی از یک دید با استفاده از دستورات SQL

```
CREATE VIEW V1
AS SELECT STID, STDEG
FROM STT
WHERE STPROG='Math'

SELECT STID
FROM V1
WHERE STDEG='bs';
```



عملیات در دیدهای رابطه‌ای

عملیات ذخیره‌سازی در دیدهای رابطه‌ای

برای انجام عملیات ذخیره‌سازی در دید از همان دستورات سه‌گانه
INSERT، UPDATE و DELETE استفاده می‌شود.



دیدها از نظر پذیرش عملیات ذخیره‌سازی

1. دیدهای پذیرا

2. دیدهای ناپذیرا



1. دیدهای گزینشی
2. دید گزینش – پرتوی دارای کلید رابطه مبنا
3. دید پیوندی CK-CK
4. دید پیوندی CK-FK
5. دید حاصل اجتماع ، اشتراك و تفاضل دو رابطه
- دیدهای پذیرا



دید گزینشی حاصل عملگر گزینش در يك رابطه
است.



در دید “گزینش-پرتوي داراي کليد رابطه مبنا” علاوه بر
گزینش تاپلهایي از رابطه مبنا ، عملگر پرتو نیز اعمال شده
است. این نوع دید را اصطلاحاً دید داراي کليد با تاپلهای
ناقص مي گوئیم.



دید پیوندی CK-CK ، حاصل پیوند دو رابطه روی
کلید کاندید مشترک آنها است و در عملیات ذخیره سازی
مشکلی ندارد



دید پیوندی CK-FK ، حاصل پیوند روی کلید کاندید یک رابطه و کلید خارجی رابطه دیگر است. این دید در حذف مشکل دارد ، زیرا با حذف یک تاپل از این دید ، در هر یک از دو رابطه مبنا ، یک تاپل حذف می شود و چنانچه کاربر بخواهد محتوای دید خود را نمایش دهد ، تاپلهای دیگر هم از دید او حذف می شوند که درخواست نکرده است.



دید حاصل اجتماع ، اشتراك و تفاضل دو رابطه در عملیات
ذخیره سازی مشکلي ندارد ، به شرط آنکه سیستم بتواند
تشخیص دهد که عمل درخواست شده ، در کدامیک از دو
رابطه مبنا انجام شود.



مشکلات مهمتر دیدهای پذیرا

- بروز عارضه جانبی در خود دید
- بروز عارضه جانبی در دیدهای دیگر
- نقض قاعده جامعیت
- بروز فزونکاری در سیستم
- تغییر ماهیت عمل درخواست شده
- تعدد تبدیلات و مشکل تصمیم‌گیری



1. دید پرتوي يا گزينشي فاقد کلید رابطه مبنا

2. دید پیوندي NK-NK

3. دید پیوندي FK-FK

4. دید حاوي صفت مجازي

5. دید حاصل تقسيم

دیدهاي ناپذيرا



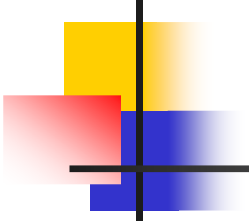
ویژگیهای دیدهای قابل بهنگام سازی

1. عبارت تعریف کننده محدوده دید ، يك عبارت معتبر SELECT باشد.
2. در کلاز FROM ، عبارت SELECT ، فقط يك جدول وجود داشته باشد.
3. جدول قید شده در کلاز FROM ، يك جدول مبنا يا يك دید قابل بهنگام سازی باشد.
4. در item-list عبارت SELECT ، ستونهای مورد نظر باید در جدول مبنا متناظر باشد.
5. در عبارت SELECT نباید کلاز GROUP BY و HAVING و گزینه DISTINCT وجود داشته باشد.
6. در کلاز WHERE عبارت SELECT نباید عبارتی حاوی کلاز FROM باشد ، به گونه ای که در آن به همان جدولی ارجاع داده شده باشد که در کلاز FROM قبلی به آن ارجاع شده است.



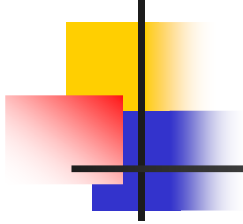
جلسه چهاردهم

طراحی پایگاه داده‌های رابطه‌ای



آنچه در این جلسه می خوانید:

- 1- مراحل طراحی پایگاه داده‌ها
- 2- نمودار مراحل اساسی طراحی پایگاه داده‌ها
- 3- روشهای طراحی منطقی پایگاه داده‌ها
- 4- روش طراحی بالا به پایین
- 5- روش سنتز رابطه‌ای
- 6- روش ترکیبی
- 7- خصوصیات طراحی خوب
- 8- چهارده روش تبدیل نمودار ER به رابطه‌ها



هدفهای کلی: آشنایی با طراحی پایگاه داده‌های رابطه‌ای

هدفهای رفتاری: دانشجو در پایان این جلسه می‌تواند:

- مراحل طراحی پایگاه داده‌ها را بیان کند.
- روشهای طراحی منطقی پایگاه را شرح دهد.
- روشهای تبدیل نمودار ER به رابطه را نام برده و توضیح دهد.
- خصوصیات یک طراحی خوب را بیان کند.

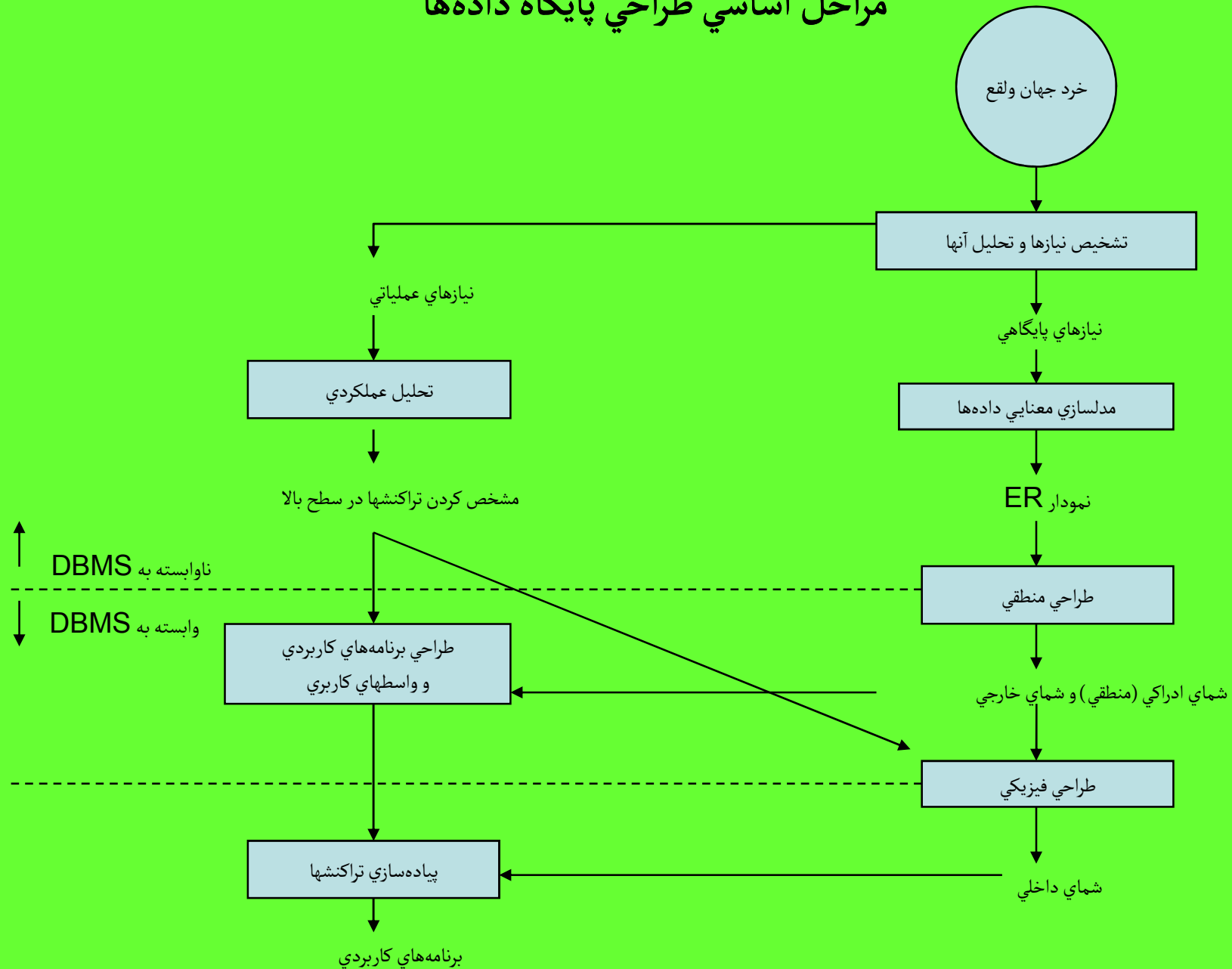


طراحی پایگاه داده‌های رابطه‌ای (روش بالا به پایین)

مراحل طراحی پایگاه داده‌ها

1. مطالعه و شناخت خرد جهان واقع
2. انجام عملیات مهندسی نیازها
3. مدلسازی معنایی داده‌ها
4. طراحی منطقی پایگاه داده‌ها
5. طراحی فیزیکی پایگاه داده‌ها
6. انجام تحلیل عملکردی: تعیین تراکنشها
7. طراحی برنامه‌های کاربردی و واسطه‌های کاربری

مراحل اساسي طراحي پایگاه داده‌ها





روشهای طراحی منطقی پایگاه داده‌ها

1. روش بالا به پایین
2. روش سنتز رابطه‌ای
3. روش ترکیبی



روش طراحی بالا به پایین

در این روش ابتدا مدلسازی معنایی داده‌ها انجام می‌شود.
سپس مدلسازی را به مجموعه‌ای از رابطه‌ها تبدیل می‌کنیم.
حاصل این فرایند ، معمولا مجموعه‌ای از رابطه‌های
خوش طرح است



روش سنتز رابطه‌ای

در این روش مجموعه صفات خردجهان واقع را مشخص می‌کنیم. سپس با تحلیل قواعد و محدودیتهای ناظر به صفات و تشخیص وابستگیهای بین آنها، صفات را متناسباً باهم سنتز می‌کنیم. هر گروه از صفات عنوان يك رابطه را به ما می‌دهد به گونه‌ای که حتی‌الامکان در مطلوبترین صورت نرمال باشد. هر رابطه بدست آمده نمایشگر بخشی از خردجهان واقع بوده، ضوابط يك طرح خوب و قابل دفاع را خواهد داشت.



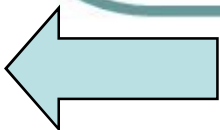
روش ترکیبی

در این روش ، ابتدا طراحی را با روش بالا به پایین انجام می دهیم و سپس با دخالت دادن برخی محدودیتهای جامعیتی که در مرحله مدلسازی معنایی داده ها منظور نشده اند و با تحلیل هر رابطه ، مرحله نرمالترسازی رابطه ها را انجام می دهیم تا به مجموعه ای از نرمالترین رابطه ها برسیم.



خصوصیات طراحی خوب

- 1- نمایش واضحی از خردجهان واقع باشد.
- 2- نمایش صحیحی از خردجهان واقع باشد.
- 3- نمایش جامعی از خردجهان واقع باشد.
- 4- تمام قواعد جامعیتی که قابل اعمال در هر مرحله از طراحی منطقی باشند ، در طراحی منظور شده باشند.
- 5- معنای هر یک از صفات از هر نوع موجودیت به درستی رعایت شده باشد.
- 6- کمترین میزان افزونگی را داشته باشد.



ادامه



خصوصیات طراحی خوب

- 7- کمترین میزان اختلاط اطلاعات را داشته باشد.
- 8- انعطاف پذیری داشته باشد.
- 9- کمترین دشواری در انجام عملیات ذخیره سازی
- 10- کمترین میزان هیچمقدار
- 11- هیچ اطلاع جعلی در اثر انجام عملیات در پایگاه داده ها پدید نیاید.
- 12- با در نظر گرفتن طراحی فیزیکی و تاثیر آن در طراحی منطقی ، بیشترین کارایی برای سیستم کاربردی پایگاه داده ها تامین شود.



روش تبدیل نمودار ER به رابطه‌ها

حالت اول

- تعداد نوع موجودیت: $n \geq 2$

- وضع موجودیتها: مستقل

- چندی ارتباط: $N:M$

در این حالت ، $n+1$ رابطه لازم است. n رابطه برای n موجودیت مستقل و یک رابطه برای نمایش ارتباط بین آنها. اگر ارتباط بین انواع موجودیتها ، صفت ساده یا مرکب چندمقداری نداشته باشد ، در این صورت کلید کاندید رابطه نمایشگر ارتباط بین n موجودیت ، از ترکیب کلیدهای کاندید n رابطه نمایشگر n موجودیت به دست می‌آید. سپس کلیدهای خارجی ، اجزاء تشکیل دهنده کلید کاندید رابطه نمایشگر ارتباط هستند.



روش تبدیل نمودار ER به رابطه‌ها

حالت دوم

- تعداد نوع موجودیت: دو
- وضع موجودیتها: مستقل
- چندی ارتباط: 1:N

در این حالت ، دو رابطه کفایت می‌کند: یک رابطه برای نمایش نوع موجودیت طرف یک ، یک رابطه برای نمایش نوع موجودیت طرف N و نیز ارتباط 1:N. در رابطه اخیر ، کلید کاندید رابطه اول ، به عنوان کلید خارجی رابطه دوم ، ارتباط مورد نظر را نشان می‌دهد و جزء تشکیل‌دهنده کلید کاندید رابطه نیست.



روش تبدیل نمودار ER به رابطه‌ها

حالت سوم

- تعداد نوع موجودیت: دو
- وضع موجودیتها: مستقل
- چندی ارتباط: 1:1

در این حالت ، دو رابطه لازم است: يك رابطه براي نمایش یکی از دو نوع موجودیت و رابطه‌اي دیگر براي نمایش نوع موجودیت دیگر و ارتباط بین دو نوع موجودیت.



روش تبدیل نمودار ER به رابطه‌ها

حالت چهارم

- تعداد نوع موجودیت: يك
- وضع موجوديتها: مستقل
- چندي ارتباط: $N:M$

در این حالت ، ارتباط با خود داریم. دو رابطه لازم است: يك رابطه براي نمایش خود نوع موجودیت و ديگري براي نمایش ارتباط ، اعم از اینکه مشارکت الزامي باشد يا نباشد.



روش تبدیل نمودار ER به رابطه‌ها

حالت پنجم

- تعداد نوع موجودیت: يك
- وضع موجوديتها: مستقل
- چندي ارتباط: 1:N

در این حالت ، هم “ارتباط با خود” داریم و يك رابطه كفايت مي كند.



روش تبدیل نمودار ER به رابطه‌ها

حالت ششم

- تعداد نوع موجودیت: يك
- وضع موجوديتها: مستقل
- چندي ارتباط: 1:1

در این حالت يك رابطه کفایت می‌کند، به شرط آنکه مشارکت در ارتباط الزامی باشد. البته می‌توان با دو رابطه هم طراحی کرد.



روش تبدیل نمودار ER به رابطه‌ها

حالت هفتم

نمایش موجودیت ضعیف

موجودیت ضعیف دارای شناسه یکتا نیست ، بلکه صفت ممیزه دارد. برای نمایش این نوع موجودیت در طراحی پایگاه رابطه‌ای ، يك رابطه طراحی می‌کنیم که در عنوان آن ، صفات موجودیت ضعیف و کلید کاندید موجودیت قوی که با آن ارتباط دارد (موجودیت شناسا) ، وجود دارند.



روش تبدیل نمودار ER به رابطه‌ها

حالت هشتم

وجود صفت چندمقداری

اگر MVA ، یک صفت (ساده یا مرکب) چندمقداری، EID شناسه موجودیت E و ...
و سایر صفات تک‌مقداری موجودیت E باشند، در این صورت برای نمایش این نوع موجودیت،
دو رابطه لازم است:

$$R_1\left(\frac{EID}{C.K.}, A_1, A_2, \dots, A_i\right)$$

$$R_2\left(\frac{EID, MVA}{C.K.}\right)$$



روش تبدیل نمودار ER به رابطه‌ها

حالت نهم

ارتباط IS-A یا “گونه‌ای است از ...”

در این حالت يك روش این است که موجودیت سطح بالاتر با يك رابطه نشان داده می‌شود و هر نوع موجودیت سطح پایین‌تر با صفات خاص خودش همراه با شناسه اصلی موجودیت سطح بالاتر ، در رابطه دیگری نمایش داده می‌شود.



روش تبدیل نمودار ER به رابطه‌ها

حالت دهم

وجود دسته (طبقه) در ارتباط IS-A

در این ارتباط وقتی زیرنوع دسته (طبقه) داریم و صفات شناسه زبرنوعها ، متفاوت باشند ، يك طرز طراحی این است که این زیرنوع را با يك رابطه نشان دهیم. کلید این رابطه ، يك کلید جایگزین (شناسه ساختگی) است و بعلاوه يك صفت دیگر در آن در نظر می گیریم نشان دهنده نوع آن (از نوع کدام زبرنوع است). هر زبرنوع را هم با يك رابطه نشان می دهیم و صفت کلید زیرنوع را به هريك از رابطه هاي نشان دهنده زبرنوعها اضافه می کنیم.



روش تبدیل نمودار ER به رابطه‌ها

حالت یازدهم

ارتباط IS – A PART – OF یا “جزئی است از ...”

در این حالت ، برای موجودیت کل (شامل) يك رابطه و برای هريك از موجودیتهای جزء (مشمول) نیز يك رابطه طراحی می‌شود. در رابطه نشان‌دهنده موجودیت جزء ، کلید کاندید رابطه نشان‌دهنده موجودیت کل ، آورده می‌شود و بقیه صفات این رابطه ، خاص خود موجودیت جزء هستند.



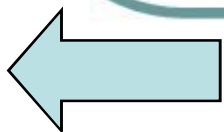
روش تبدیل نمودار ER به رابطه‌ها

حالت دوازدهم

بیش از يك ارتباط بين دو نوع موجودیت

فرض می‌کنیم هر دو نوع موجودیت مستقل هستند. تعداد رابطه‌ها، بستگی به چندی هر ارتباط دارد. به صورت زیر عمل می‌کنیم:

- هر ارتباط با چندی $N:M$ با يك رابطه نمایش داده می‌شود.
- نوع موجودیت سوي N و همه ارتباطهای با چندی $1:N$ ، اگر مشارکت دو نوع موجودیت در همه ارتباطها الزامی باشد، را می‌توان با يك رابطه نمایش داد.



ادامه



روش تبدیل نمودار ER به رابطه‌ها

حالت دوازدهم

بیش از یک ارتباط بین دو نوع موجودیت

- اگر مشارکت دو نوع موجودیت در یک ارتباط $1:N$ ، الزامی نباشد، بهتر است برای چنین ارتباطی یک رابطه جداگانه طراحی شود که صفات آن، شناسه دو نوع موجودیت و در صورت وجود، صفات خود ارتباط هستند.
- در صورت وجود بیش از یک ارتباط $1:1$ و الزامی بودن مشارکتها، هر دو نوع موجودیت و همه این ارتباطها را می‌توان با یک رابطه نمایش داد.
- اگر مشارکت در یک ارتباط $1:1$ الزامی نباشد، بهتر است این ارتباط را با یک رابطه جداگانه نمایش داد.



روش تبدیل نمودار ER به رابطه‌ها

حالت سیزدهم

ارتباط سلسله‌مراتبی (پدر - فرزندی)

در تبدیل این نوع ارتباط بین موجودیتها به طراحی رابطه‌ای یک روش چنین است:

1- برای نوع موجودیت ریشه ، یک رابطه در نظر می‌گیریم.

2- کلید کاندید این رابطه همان شناسه ریشه است.

3- برای هر نوع موجودیت فرزند یک رابطه در نظر می‌گیریم و کلید کاندید هر نوع

موجودیت فرزند ، ترکیب شناسه آن نوع موجودیت با کلید کاندید نوع موجودیت

پدرش است.



روش تبدیل نمودار ER به رابطه‌ها

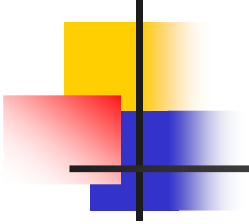
حالت چهاردهم

تجمع (ارتباط با ارتباط)

در این حالت ، ابتدا نمودار ER جزئی درون مستطیل نشان دهنده نوع موجودیت حاصل از عمل تجمع را به رابطه‌های لازم تبدیل می‌کنیم. سپس با توجه به چندی ارتباط بین نوع موجودیت مرتبط با ارتباط و چگونگی مشارکت در ارتباط با ارتباط ، طراحی را کامل می‌کنیم.

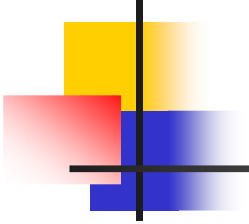
جلسه پانزدهم

نرمالترسازي رابطه‌ها



آنچه در این جلسه می خوانید:

- 1- آنومالی در عملیات ذخیره سازی
- 2- صورتهای نرمال
- 3- وابستگی تابعی
- 4- وابستگی تابعی نامهم (بدیهی)
- 5- قواعد استنتاج آرمسترانگ
- 6- مجموعه کاهش ناپذیر وابستگیهای تابعی
- 7- وابستگی تابعی تام (کامل)
- 8- رابطه 1NF
- 9- رابطه 2NF



آنچه در این جلسه می خوانید:

10- رابطه 3NF

11- صورت نرمال بایس-کاد

12- وابستگی تابعی چندمقداری

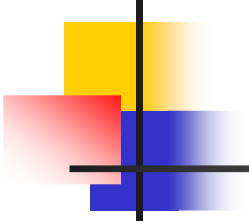
13- رابطه 4NF

14- وابستگی تابعی چندمقداری ادغام شده

15- رابطه 5NF

16- مزایا و معایب نرمالترسازی

17- انواع افزونگی در پایگاه داده ها



هدفهاي كلي: آشنائي با نرمالترسازي رابطه‌ها

هدفهاي رفتاري: دانشجو در پايان اين جلسه مي‌تواند:

- صورتهاي مختلف نرمال سازي را نام برده و هريك را توصيف كند.
- مزاي و معايب روش نرمالترسازي را شرح دهد.



طراحی پایگاه داده‌های رابطه‌ای روش سنتز: نرمال‌سازی رابطه‌ها

يك رابطه ، هرچند نرمال ، بازهم ممکن است در عملیات
ذخیره‌سازی (درج ، حذف و بهنگام‌سازی) مشکلاتی داشته باشد.



آنومالي در عملیات ذخیره‌سازی به هریک از سه وضع زیر گفته می‌شود:

- عدم امکان انجام يك عمل (که منطقاً باید قابل انجام باشد)
- بروز پیامد بد پس از انجام يك عمل
- بروز فزونکاری در سیستم در انجام يك عمل



صورتهاي نرمال

- صورت نخست نرمال (1NF)
- صورت دوم نرمال (2NF)
- صورت سوم نرمال (3NF)
- صورت نرمال بایس-کاد (BCNF)
- صورت چهارم نرمال (4NF)
- صورت پنجم نرمال (5NF)
- صورت نرمال میدان-کلیدی (DKNF)
- صورت نرمال تحدید-اجتماع (RUNF)



وابستگی تابعی

تعریف- فرض کنید که R یک متغیر رابطه‌ای و A و B دو زیرمجموعه دلخواه از عنوان R باشند. می‌گوییم B با A وابستگی تابعی دارد و چنین نمایش می‌دهیم:

اگر و فقط اگر در هر میدان ممکن $A \rightarrow B$ از متغیر رابطه‌ای R ، به هر مقدار A فقط یک مقدار B متناظر باشد.



وابستگی تابعی نامهم (بدیهی)

اگر در $R(A, B, C, \dots)$ داشته باشیم: $B \subseteq A$ در این صورت:

یک وابستگی تابعی نامهم است. به بیان دیگر اگر B زیرمجموعه‌ای از A باشد،
در این صورت $A \rightarrow B$ یک وابستگی نامهم است.



قواعد استنتاج آرمسترانگ

فرض: A, B, C و D زیرمجموعه‌هایی از صفات رابطه R باشند. قواعد زیر برقرارند:

1. قاعده انعکاس: اگر $A \rightarrow B$ آنگاه $B \rightarrow A$
2. قاعده تعدی (تراگذری): اگر $A \rightarrow B$ و $B \rightarrow C$ آنگاه $A \rightarrow C$
3. قاعده افزایش: اگر $A \rightarrow B$ آنگاه $(A, C) \rightarrow (B, C)$
4. قاعده تجزیه: اگر $A \rightarrow (B, C)$ و $A \rightarrow C$ آنگاه $A \rightarrow B$
5. قاعده اجتماع: اگر $A \rightarrow B$ و $A \rightarrow C$ آنگاه $A \rightarrow (B, C)$
6. قاعده ترکیب: اگر $A \rightarrow B$ و $C \rightarrow D$ آنگاه $(A, C) \rightarrow (B, D)$
7. قاعده شبه تعدی: اگر $A \rightarrow B$ و $(C, B) \rightarrow D$ آنگاه $(A, C) \rightarrow D$
8. قاعده یگانگی عمومی: اگر $A \rightarrow B$ و $C \rightarrow D$ آنگاه $A \cup (C - B) \rightarrow (B, D)$



مجموعه کاهش ناپذیر وابستگیهای تابعی

مجموعه‌ای از وابستگیهای تابعی R ، به نام F را کاهش ناپذیر گوییم اگر:

1. در F وابستگی تابعی افزونه نباشد.
2. در سمت راست هر FD از F صفت ساده وجود داشته باشد.
3. هیچ صفتی در سمت چپ FD های F افزونه نباشد.



وابستگی تابعی تام (کامل)

اگر X و Y دو زیرمجموعه از مجموعه عنوان رابطه R باشند، می‌گوییم Y با X وابستگی تابعی تام دارد و چنین نشان می‌دهیم:

$$X \Rightarrow Y$$

اگر و فقط اگر Y با X وابستگی داشته باشد. ولی با هیچ زیرمجموعه‌ای از X وابستگی تابعی نداشته باشد.



رابطه 1NF

تعریف- رابطه‌ای 1NF است اگر هر صفت خاصه آن در هر تاپل ،
تک‌مقداری باشد ، به بیان دیگر ، صفت چندمقداری نداشته باشد.



رابطه 2NF

تعریف- رابطه‌ای 2NF است اگر اولاً 1NF باشد و ثانياً تمام صفات غیرکلید با کلید اصلی وابستگی تابعی تام داشته باشند. به عبارت دیگر هر صفت غیرکلید با کلید اصلی بطور کاهش‌ناپذیر وابسته باشد.



رابطه 3NF

تعریف- رابطه‌ای 3NF است اگر 2NF باشد و هر صفت غیرکلید با کلید اصلی ، وابستگی تابعی بی واسطه داشته باشد.



صورت نرمال بایس-کاد

تعریف اول: رابطه‌ای **BCNF** است اگر در آن هر دترمینان ،
کلید کاندید باشد. این تعریف ساده‌شده و غیر صوری است.



صورت نرمال بایس-کاد

تعریف دوم: رابطه $R(H)$ ، با مجموعه وابستگی‌های تابعی F ، در **BCNF** است اگر
برای هر وابستگی تابعی در F به صورت $A \rightarrow B$ که در آن A حداقل یکی از دو حالت زیر برقرار باشد: $B \subseteq R(H)$
a: یک وابستگی نامهم باشد یعنی
 $A:b$ سوپرکلید رابطه R باشد.
 $B \subseteq A$



صورت نرمال بایس-کاد

تعریف سوم: رابطه R در $BCNF$ است هرگاه يك وابستگی تابعي به صورت
در مجموعه FD هاي R وجود داشته باشد ، A سوپرکلید R باشد.



صورت نرمال بایس-کاد

تعریف چهارم: رابطه‌ای **BCNF** است اگر و فقط اگر سمت چپ هر **FD** مهم (مطرح) و کاهش ناپذیر، کلید کاندید رابطه باشد.



وابستگی تابعی چندمقداری

تعریف اول- در رابطه $R(X, Y, Z)$ با صفات ساده یا مرکب X ، Y و Z می‌گوییم که Y با X وابستگی تابعی چندمقداری دارد و چنین نمایش می‌دهیم:

اگر به یک مقدار X ، مجموعه‌ای از مقادیر Y متناظر باشند.



وابستگی تابعی چندمقداری

تعریف دوم- در رابطه R ، صفت Y با X وابستگی تابعی چندمقداری دارد اگر فقط اگر مجموعه مقادیر Y متناظر با یک مقدار از جفت (X, Z) در R فقط به مقدار X بستگی داشته باشد و وابسته به مقدار Z نباشد.



رابطه 4NF

تعریف- رابطه‌ای 4NF است اگر BCNF باشد و در آن وابستگی
تابعی چند مقداری مهم وجود نداشته باشد.



وابستگی تابعی چندمقداری ادغام شده

در رابطه $R(X, Y, Z, U, V, \dots)$ ، پدیده وابستگی تابعی چندمقداری ادغام شده وجود دارد اگر در پرتوی از این رابطه ، پدیده وابستگی تابعی چندمقداری وجود داشته باشد. توجه داشته باشید که این پرتو باید روی سه صفت (ساده یا مرکب) باشد.



رابطه 5NF

تعریف- رابطه‌ای 5NF است اگر تمام وابستگی‌های پیوندی آن ناشی از کلیدهای کاندید آن باشد. در نتیجه:

اگر بتوانیم یک وابستگی پیوندی در رابطه R پیدا کنیم که در همه پرتوهای کلید کاندید R وجود نداشته باشد، رابطه 5NF نیست.



مزایای نرمالترسازی

- ارائه یک طراحی بهتر و واضح تر با کمترین اختلاط اطلاعات
- کاهش بعضی انواع افزونگی
- کاهش بعضی آنومالیها
- تسهیل اعمال بعضی قواعد جامعیت



انواع افزونگی در پایگاه داده‌ها:

1. افزونگی طبیعی

2. افزونگی تکنیکی ناشی از وجود کلید خارجی یا هرصفت مشترک

دیگر در دو یا بیش از دو رابطه

3. افزونگی ناشی از وابستگی بین صفات

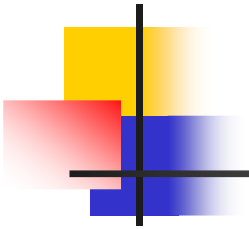


معایب روش نرمالترسازي

- 1- بروز فزونکاري در سیستم در عمل بازیابی
- 2- ایجاد نوعي افزونگي از نوع افزونگي در سطح ادراکي
- 3- زمانگیر بودن فرآیند نرمالترسازي به ویژه اگر محیط عملیاتی بزرگ و تعداد رابطه‌ها زیاد باشد.
- 4- تصمیم‌گیری دشوار در اثر تعدد تجزیه‌ها
- 5- ممکن وابستگی بین مجموعه صفات يك خردجهان ، وابستگی‌هاي به جز وابستگی تابعي و پیوندي باشد و بنابراین سبب مطرح شدن قواعد جامعیت دیگر و نیز ضوابط دیگری برای تجزیه رابطه‌ها شود و ...

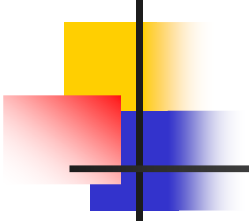
جلسه شانزدهم

طراحی فیزیکی پایگاه داده‌ها



آنچه در این جلسه می خوانید:

- 1- تحلیل عوامل
- 2- تحلیل پرسشها
- 3- تحلیل تراکنشها
- 4- تصمیمات در طراحی فیزیکی
- 5- اهداف تنظیم پایگاه دادهها
- 6- اطلاعات لازم در تنظیم پایگاه دادهها

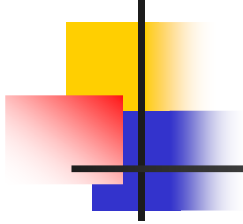


آنچه در این جلسه می خوانید:

7- مسائل مطرح در تنظیم

8- نکات قابل توجه در بازبینی طراحی منطقی پایگاه داده ها پس از عملیاتی شدن آن

9- کارهای لازم برای اصلاح شاخصهای ایجادشده در مرحله تنظیم پایگاه داده ها



هدفهاي كلي: آشنائي با طراحي فيزيكي پايگاه داده‌ها

هدفهاي رفتاري: دانشجو در پايان اين جلسه مي‌تواند:

- عواملی که در طراحی فیزیکی پایگاه داده‌ها موثر هستند را بیان کند.
- پرسشهایی را که در تحلیل پرسشها و تراکنش مطرح می‌شوند ، بشناسد.
- اطلاعات لازم در تنظیم پایگاه داده‌ها را بیان کند.



طراحی فیزیکی پایگاه داده‌ها

طراحی فیزیکی به عواملی از جمله حجم و ماهیت داده‌ها، تعداد کاربران و از آنجا تعداد و ماهیت تراکنشها و نهایتاً تعداد و ماهیت برنامه‌های کاربردی و فرکانس اجرای هر برنامه بستگی دارد.



تحلیل عوامل

تحلیل پرسشها

در این تحلیل باید به پرسشهای زیر پاسخ داده شود:

- چه رابطه‌هایی در پرسش دخیل هستند
- نوع رابطه‌ها چیست
- صفات موجود در مسند گزینش
- صفات موجود در مسند پیوند
- صفات مورد نظر در پرسش
- فرکانس اجرای هر پرسش
- اسلوب اجرای پرسش: اندرکنشی یا کامپایل شده



تحلیل عوامل

تحلیل تراکنشها

در این تحلیل باید به پرسشهای زیر پاسخ داده شود:

- رابطه‌های دخیل در تراکنش و نوع هر رابطه
- نوع هر عمل در تراکنش
- صفات موجود در مسند گزینش
- صفاتی که مقدارشان تغییر می‌کند



تصمیمات در طراحی فیزیکی

- روی چه فایل‌هایی شاخص ایجاد شود یا تابع درهم‌ساز اعمال شود.
- روی چه صفاتی از فایل ، شاخص ایجاد شود.
- شاخص ساده باشد یا مرکب.
- شاخص خوشه‌ساز باشد یا نباشد.



اهداف تنظيم پایگاه داده‌ها

- تسريع اجرای برنامه‌هاي کاربردي
- کاهش زمان پاسخدهي پرسشها
- بهبود توان عملیاتي سیستم مدیریت پایگاه داده‌ها
- اجتناب از خرید سخت افزار بیشتر
- کاهش اندازه پیکربندي سخت افزاري
- رضایت بیشتر کاربران سیستم



اطلاعات لازم در تنظیم پایگاه داده‌ها

- اطلاعات در مورد خود DBMS
- اطلاعات اولیه مورد استفاده در طراحی منطقی و فیزیکی پایگاه
- آمار جمع‌آوری شده توسط DBMS و DBA در مورد رفتار سیستم مانند:
 - کاردینالیتی هر رابطه
 - تعداد مقادیر متمایز برای هر صفت و درصد هیچ مقدار برای هر صفت
 - فرکانس عرضه هر پرسش به سیستم و فرکانس اجرای آن
 - فرکانس اجرای هر تراکنش
 - اندازه صفحات
 - زمان اجرای پرسشها و تراکنشها
 - ...



مسائل مطرح در تنظیم

- چگونگی اجتناب از بروز تنگنا در قفل گذاری روی داده ها به منظور افزایش همروندی اجرای تراکنشها
- چگونگی کاهش سربار ناشی از ایجاد فایل های ثبت تراکنشها
- چگونگی بهینه سازی اندازه بافرها
- چگونگی بهینه سازی زمان بندی اجرای تراکنشها
- چگونگی بهینه سازی تخصیص منابع



در بازبینی طراحی منطقی پایگاه داده‌ها پس از عملیاتی شدن آن ، باید به نکات زیر توجه شود:

- آیا رابطه‌هایی باید تجزیه شوند ؟
- آیا سطح نرمالیتی رابطه‌هایی باید کاهش یابد ؟
- آیا مجموعه دیگری از رابطه‌ها ، با همان سطح نرمالیتی ، باید طراحی شوند ؟
- آیا صفاتی با پذیرش میزانی از افزونگی ، باید در رابطه‌هایی تکرار شوند ؟
- آیا در طراحی دیدهای خارجی باید تغییراتی داده شود ؟



کارهای لازم برای اصلاح شاخصهای ایجادشده در مرحله تنظیم پایگاه داده‌ها

- ایجاد شاخصهای جدید
- حذف برخی از شاخصها
- کاهش تعداد شاخصها روی صفاتی که مقادیرشان مرتب تغییر می‌کند.
- اصلاح شاخصهایی که در کنترل همروندی تراکنشها تاثیر دارند
- بازسازی شاخصها می‌تواند در کارایی سیستم تاثیر مثبت داشته باشد
- در بعضی از **RDBMS**ها بهینه‌ساز، اگر در پرسش، عبارت محاسباتی، مقایسه یا هیچ مقدار و محاسبه تفاوت مقادیر وجود داشته باشد، از شاخص استفاده نمی‌کند.