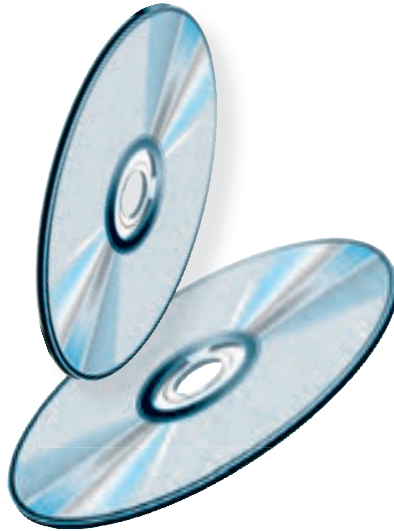


هدف کلی

آشنایی با مفاهیم بانک اطلاعاتی و کار با DBMS متداول



فصل ۱

مفاهیم بانک اطلاعاتی^۱

در این فصل به معرفی تعاریف و مفاهیم اولیه بانک‌های اطلاعاتی خواهیم پرداخت. اگرچه نرم‌افزارهای کاربردی به‌سهولت قابل استفاده هستند، اما بدون دانستن دقیق مفاهیم نمی‌توان از آن‌ها به‌طور مناسب استفاده کرد و هیچ‌گاه نمی‌توان بانک اطلاعاتی کارآمدی را ایجاد و پیاده‌سازی کرد.

هدف‌های رفتاری: پس از آموزش این فصل هنرجو می‌تواند:

- انواع سیستم‌های ذخیره و بازیابی را توضیح دهد.
- مفهوم فیلد، رکورد، فایل را توضیح دهد.
- بانک اطلاعاتی را تعریف کند.
- سیستم مدیریت بانک اطلاعاتی و معماری آن را شرح دهد.
- مفاهیم مدل رابطه‌ای را شرح دهد.
- سیستم بانک اطلاعاتی را طراحی کند.

۱-۱- کلیات

بانک‌های اطلاعاتی یکی از انواع سیستم‌های «ذخیره و بازیابی اطلاعات»^۲ است. سیستم ذخیره و بازیابی سیستمی است که به کاربر امکان می‌دهد تا داده‌ها و اطلاعات خود را ذخیره، بازیابی و پردازش نماید.

۱- در این کتاب پایگاه داده‌ها، بانک اطلاعاتی، بانک داده و بانک به یک معنی استفاده شده است.

نظر به تعریف عام بالا، موارد زیر قابل ذکر است :

۱- کاربر می تواند دارای قابلیت «برنامه سازی» یا فاقد این مهارت باشد.

۲- داده می تواند متن، تصویر، صوت و نظایر آن باشد.

سیستم ذخیره و بازیابی اطلاعات با گذشت زمان، از مفهوم ساده فایلینگ^۱ به صورت زیر توسعه

پیدا کرده است :

۱- سیستم فایلینگ Filing System (FS)

۲- سیستم مدیریت داده ها Data Management System (DMS)

۳- سیستم مدیریت پایگاه داده ها Database Management System (DBMS)

۴- سیستم مدیریت پایگاه دانش Knowledge Base Management System (KBMS)

پیش از تعریف سیستم مدیریت پایگاه داده بهتر است، برخی از مفاهیم، تعاریف مقدماتی و مفهوم

سیستم فایلینگ بررسی شود تا با دقت بیشتری بتوانیم مفهوم بانک اطلاعاتی را بشناسیم.

۱-۲- مفاهیم مقدماتی در ذخیره و بازیابی

۱-۲-۱- فیلد^۲ : یک قطعه داده است^۳.

هر فیلد دارای دو جزء می باشد :

۱- نام

۲- مقدار

مقدار، حکم داده^۴ را دارد و نام فیلد به همراه مقدار در حکم اطلاع^۵ خواهد بود. به عنوان

مثال :

$$\underbrace{\text{نام خانوادگی}} = \underbrace{\text{علوی}} \\ \text{نام فیلد} \quad \text{مقدار فیلد}$$

در واقع هر فیلد یک ویژگی یا صفت به حساب می آید. توجه کنید که از نام فیلد و مقدار فیلد به همراه هم اطلاع پیدا می کنیم که نام خانوادگی فرد مورد نظر علوی است. مثال جالب تر این که : مقدار عددی «۸۸۹۶۵۴۲۳» هیچ معنی خاصی ندارد. حال به این عبارت توجه کنید : «تلفن = ۸۸۹۶۵۴۲۳»،

۱- Filing

۲- Field

۳- فیلد واحد معنایی داده است. (Semantic Data Unit)

۴- Data

۵- Information

۶- Attribute

با درج نام فیلد در کنار مقدار عددی، شما اطلاع پیدا می‌کنید که آن عدد شماره تلفن است. اگر به‌جای تلفن، شماره شناسنامه قرار گیرد اطلاع کسب شده متفاوت خواهد بود.



یکی از خصوصیات مهم فیلد، نوع داده‌ای است که در آن قرار می‌گیرد. مثلاً
Real ، Integer یا Text و

۱-۲-۲ رکورد^۱: به مجموعه فیلدهای مرتبط، یک رکورد می‌گوییم.
به‌عنوان مثال، قالب رکورد مربوط به یک دانش‌آموز می‌تواند شامل نام فیلدهای زیر باشد:
نام، نام خانوادگی، سال تولد، مقطع تحصیلی، کلاس و ...
محتوای رکوردهای دانش‌آموزان می‌تواند برای دو دانش‌آموز هم‌کلاسی به‌صورت زیر باشد:
۱- مجید، شجاعی، ۱۳۷۰، اول دبیرستان، کلاس ۱ الف و ...
۲- حسن، کاملیان، ۱۳۷۱، اول دبیرستان، کلاس ۱ الف و ...
محتوای رکورد می‌تواند دارای طول ثابت یا متغیر باشد. در حالت رکورد با طول ثابت، تعداد، ترتیب و اندازه فیلدهای متناظر در تمام رکوردها یکسان در نظر گرفته می‌شود و در حالت رکورد با طول متغیر، چنین الزامی وجود ندارد.

۱-۲-۳ فایل^۲: مجموعه‌ای از رکوردها، تشکیل فایل را می‌دهند. به‌عنوان مثال: فایل تحصیلی دانش‌آموزان، محتوای فیلدهای رکوردهای مربوط به دانش‌آموزان را دارا می‌باشد.

۱-۲-۴ کلید^۳: به فیلد یا زیرمجموعه‌ای از فیلدها، که باعث یکتایی هریک از رکوردها شود، کلید گویند. به‌وسیله کلید می‌توان به یک رکورد مشخص دسترسی داشت. به‌عنوان مثال، به رکوردهای زیر توجه کنید:

ردیف	شماره دانش‌آموزی	نام	نام خانوادگی
۱	۱۰۰	علی	حسینی
۲	۱۱۰	حسن	احمدی
۳	۱۰۸	علی	علوی
۴	۹۴	رضا	علوی

فیلد نام نمی‌تواند کلید باشد زیرا دو مقدار مشابه «علی» در رکوردهای ردیف ۱ و ۳ وجود دارد. هم‌چنین فیلد نام خانوادگی نمی‌تواند کلید باشد زیرا مقادیر «علوی» در رکوردهای ردیف ۳ و ۴ وجود دارد. اما شماره دانش‌آموزی می‌تواند کلید باشد زیرا هیچ دو شماره دانش‌آموزی برابر وجود ندارد. با توجه به رکوردهای این جدول، فیلدهای نام و نام خانوادگی نیز با هم می‌توانند کلید باشند. اگرچه در یک محیط عملیاتی مدرسه واقعی، ممکن است دو دانش‌آموز نام و نام خانوادگی یکسان داشته باشند.



آیا ممکن است در یک سیستم ذخیره و بازیابی کلید وجود نداشته باشد؟

در انتخاب کلید موارد زیر قابل توجه است :

- ۱- کلید می‌تواند ساده یا مرکب باشد.
- ۲- ممکن است کلیدهای مختلفی اعم از ساده یا مرکب بتوان تشخیص داد (کلید کاندید) ولی یکی از آن‌ها را با توجه به ملاحظات به‌عنوان کلید اصلی^۱ در نظر می‌گیریم.
- ۳- در برخی موارد به‌جای استفاده از فیلدها و صفات مورد نیاز شناسایی شده، به‌صورت مجازی فیلدی با مقدار یکتا (صفت مجازی)، به‌عنوان کلید اضافه می‌شود. مانند شماره دانش‌آموزی. مجازی بودن یعنی یک فرد زمانی که در یک مدرسه ثبت‌نام می‌کند و دانش‌آموز آن مدرسه می‌شود، یک شماره دانش‌آموزی می‌گیرد و زمانی که فارغ‌التحصیل می‌شود دیگر این شماره را از دست می‌دهد.
- ۴- مقدار فیلدی که کلید در نظر گرفته می‌شود، تهی (خالی) نیست و طبق تعریف کلید، مقدار تکراری ندارد.

۱-۳- عملیات روی رکوردها

این عملیات عبارت است از :

Retrieve

۱- بازیابی رکوردها

Insert

۲- درج رکوردها

^۱ - Primary Key

Delete

۳- حذف رکوردها

Update

۴- به هنگام سازی رکوردها

Restructure

۵- تغییر ساختار

۱-۳-۱- بازیابی رکوردها (Retrieve) : منظور از بازیابی به دست آوردن رکورد یا

رکوردهایی است که دارای شرایط خاصی هستند. به عنوان مثال می‌خواهیم اطلاعات دانش‌آموزانی که معدل آنها بیش از ۱۶ است را به دست آوریم. به این کار بازیابی گفته می‌شود. ممکن است نتیجه بازیابی یک یا چند رکورد باشد.

به طور کلی برای دستیابی و بازیابی رکوردها دو روش وجود دارد :

۱- ترتیبی^۱

۲- مستقیم^۲

در روش ترتیبی، رکوردها از ابتدا یکی پس از دیگری بررسی می‌شوند تا «محتوای» مورد نظر بازیافت شود. مثلاً فرض کنید می‌خواهیم منزل شخصی به نام علی علوی را در یک ساختمان چند طبقه پیدا کنیم. برای یافتن آپارتمان مورد نظر باید مثلاً از طبقه اول، زنگ واحدها را پشت سر هم بزیم تا به منزل این شخص برسیم. این روش کند و زمان بر است.

در روش مستقیم با توجه به محتوا، «آدرس» رکورد به دست می‌آید و رکورد مورد نظر به طور مستقیم با آن آدرس بازیابی می‌شود. روش مستقیم به وسیله ساختارهای مختلف پیاده‌سازی می‌شود مثل شاخص^۲. در مثال بالا اگر به دفتر سرایدار آپارتمان مراجعه کنیم، وی می‌گوید منزل این شخص مثلاً طبقه چهارم، واحد ۲ است. ما می‌توانیم از آسانسور استفاده کنیم و سریع به منزل این شخص برسیم. حتی اگر آسانسور در دسترس نباشد از راه‌پله‌ها خود را به طبقه چهارم می‌رسانیم بدون آن که در طبقه‌ها وقت صرف جستجو کنیم^۲.

۲-۳-۱- درج رکوردها (Insert) : منظور، افزودن رکورد یا رکوردهایی است که تاکنون

در سیستم وجود نداشته است. درج می‌تواند در انتهای فایل یا محل منطقی آن رکورد صورت گیرد. محل منطقی، محلی است که نظم فایل را حفظ می‌کند.

به عنوان مثال فرض کنید، فایلی به صورت زیر وجود دارد. اگر بخواهیم دانش‌آموزی با معدل ۱۵ را اضافه کنیم و نظم فایل که به صورت صعودی مقادیر معدل مرتب است حفظ شود، باید این رکورد

۱- Sequential

۲- Direct

۳- Index

۴- اگرچه از نظر زمانی نسبت به روش ترتیبی سریع‌تر است اما مصرف حافظه در آن بیشتر خواهد بود.

بین رکورد اول و دوم قرار گیرد. یعنی رکوردی بین اول و دوم اضافه کنیم و اطلاعات دانش آموز جدید را در آن ثبت کنیم.

سایر فیلدها ...	فیلد معدل		سایر فیلدها ...	فیلد معدل
	۱۴	درج رکوردی با معدل ۱۵ در محل منطقی	رکورد اول	۱۴
	۱۷		رکورد دوم	۱۵
	۱۸		رکورد سوم	۱۷
	۲۰		رکورد چهارم	۱۸
رکورد اول			رکورد پنجم	۲۰

شکل ۲-۱- درج در محل منطقی (نظم صعودی فیلد معدل حفظ شده است)

۳-۳-۱- حذف رکوردها (Delete): در صورتی که به رکورد یا رکوردهایی نیاز نداشته باشیم، آن را از فایل مربوطه حذف می‌کنیم.

۳-۳-۲- به‌هنگام‌سازی رکوردها (Update): اصلاح مقادیر فیلد یا فیلدهایی از رکورد، در اصطلاح به‌هنگام‌سازی نام دارد. به‌عنوان مثال تغییر نمرهٔ یک درس دانش‌آموز به‌هنگام‌سازی محسوب می‌شود.

۳-۳-۳- تغییر ساختار (Restructure): اگر به هر دلیل در ساختار رکوردها تغییر ایجاد شود مثلاً اندازهٔ فیلد تغییر کند یا فیلد جدیدی اضافه شود، تغییر ساختار صورت می‌گیرد.

۴-۱- سیستم فایلینگ (FS)

همان‌طور که گفتیم سیستم فایلینگ، اولین نوع سیستم ذخیره و بازیابی اطلاعات است. در این سیستم، نرم‌افزارهای کاربردی، مدیریت ذخیره و بازیابی داده‌ها را نیز برعهده داشتند. در این سیستم برای کاربردهای مختلف، فایل‌های جداگانه‌ای طراحی و پیاده‌سازی می‌شوند. در سیستم فایلینگ انجام عملیات به‌وسیلهٔ برنامه‌سازی و پیاده‌سازی الگوریتم‌ها انجام می‌شود. زمانی که سیستم بزرگ و پیچیده می‌شود، هر تغییر جزئی در خواسته‌ها، منجر به کدنویسی می‌گردد.

به علت وجود فایل‌های مختلف، کنترل رعایت استاندارد در زمان پیاده‌سازی سخت است. به‌عنوان مثال فیلد نام‌خانوادگی در دو فایل ممکن است اندازه‌های مختلف داشته باشد. در سیستم فایلینگ، ممکن است «ناسازگاری داده‌ها» به‌وجود آید. در نظر بگیرید شخصی در زیر سیستم اول

خود را «علی علوی تهرانی» و در زیر سیستم دوم «علی علوی» معرفی کند. بدین ترتیب مثلاً در مقایسه دو زیرسیستم، این دو مقدار یکسان نیستند و این به معنی ناسازگاری داده است. هم چنین احتمال بروز افزونگی^۱ نیز وجود دارد. یعنی محتویات بعضی از فیلدها به صورت متعدد و تکراری ذخیره شوند. به طور مثال، مشخصات سکونت یک فرد هم در سیستم فایلینگ اول و هم در سیستم دوم ثبت گردد. نتیجه افزونگی، اتلاف حافظه و مشکلات در انجام عملیات مبنایی است.



نکته
افزونگی، می تواند ناشی از طراحی نامناسب باشد که در سایر سیستم های ذخیره و بازیابی نظیر بانک اطلاعاتی نیز وجود دارد و منحصر به سیستم فایلینگ نیست.

برنامه سازی در این روش معمولاً با استفاده از یک زبان سطح بالا صورت می گیرد و معمولاً فایل های سیستم قابل استفاده به وسیله سایر سیستم های فایلینگ نیستند. سیستم فایلینگ هنوز کاربرد دارد و به طور کلی معایب روش فایلینگ را می توان به صورت زیر خلاصه کرد:

- ۱- احتمال بروز افزونگی (تکرار در ذخیره سازی)
- ۲- حجم زیاد کدنویسی
- ۳- احتمال بالای ناسازگاری داده ها
- ۴- سخت بودن اعمال استاندارد در سیستم ها

۵-۱- پایگاه داده ها

سیستم مدیریت پایگاه داده، یکی از سیستم های توسعه یافته ذخیره و بازیابی اطلاعات است. اگرچه این مفهوم بسیار شناخته شده و رایج است، اما واقعیت آن است که تعریف کاملاً واحدی در متون معتبر برای آن ذکر نشده است.

در نگرش بانک اطلاعاتی، طراحی به صورت یکپارچه انجام می گیرد. از آن جایی که یک محیط عملیاتی می تواند دارای زیر محیط های مختلفی باشد، این نگرش باعث می شود برخی معایب روش فایلینگ نظیر ناسازگاری داده ها و اعمال نشدن استانداردها رفع گردد، مثلاً برای فیلد نام در تمام محیط، مقدار

یکسانی از حافظه اختصاص می‌یابد. حال آن‌که در روش فایلینگ به علت مجزا بودن فایل‌ها به سختی و با صرف هزینه بالا می‌توان نگرش بانک اطلاعاتی داشت.

یکپارچه‌سازی و مدیریت متمرکز باعث جلوگیری از تکرار در ذخیره‌سازی (افزونگی) خواهد شد و در نتیجه از حافظه، استفاده بهتری صورت می‌گیرد. در مجموع می‌توان پایگاه داده را به صورت زیر تعریف کرد:

«مجموعه‌ای از داده‌ها به صورت یکپارچه با حداقل افزونگی، تحت کنترل یک سیستم متمرکز و در چارچوب یک مدل داده‌ای»

با ترکیب دست‌آوردهای شبکه‌ای، پایگاه داده می‌تواند به وسیله چند کاربر و همزمان استفاده شود.



تعریف‌های مختلف پایگاه داده را بررسی کنید.

۶-۱- سیستم مدیریت پایگاه داده

اصلی‌ترین تفاوت روش پایگاه داده باروش‌های قبلی مثل سیستم فایلینگ، وجود حصارى به نام «سیستم مدیریت بانک اطلاعاتی» است. هرگونه دستیابی به داده‌ها باید از طریق آن انجام شود.

در DBMS، بانک اطلاعاتی و تمامی فایل‌های آن فقط و فقط در اختیار این نرم‌افزار قدرتمند هستند. کاربران درخواست خود را به این نرم‌افزار ارسال می‌کنند و در صورت تأیید، کار خواسته شده انجام می‌گیرد.

۱-۶-۱ معماری پایگاه داده: سؤال این است که چگونه می‌توان در یک سیستم بزرگ و پیچیده بانک اطلاعاتی، بین کاربران و داده‌های بسیار زیاد ذخیره شده روی رسانه ارتباط برقرار کرد و

چگونه سیستم مدیریت پایگاه داده درخواست‌های کاربران برای عملیات پایگاه داده‌ای (مانند: بازیافت، درج، به‌هنگام‌سازی، حذف و غیره) را اداره و کنترل می‌کند.

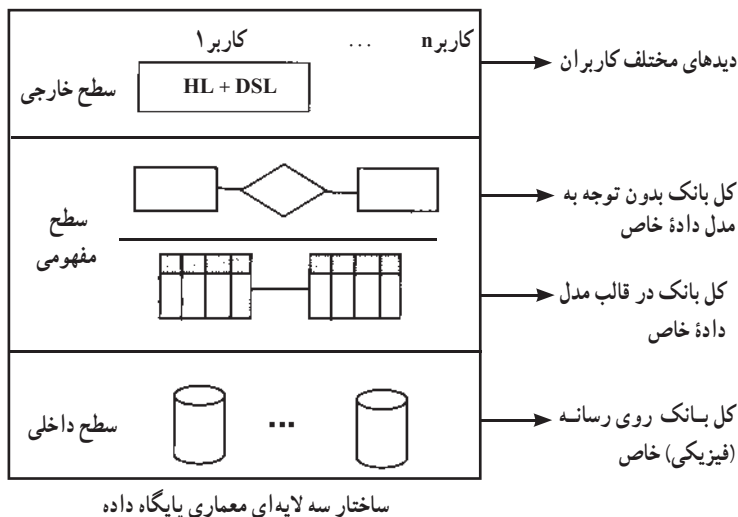
پاسخ این سؤال در معماری سه لایه‌ای است که به وسیله کمیته ANSI/SPARC^۱ عرضه شده است. در ایجاد این معماری سطوح به گونه‌ای از هم مجزا شده‌اند که برای رسیدن به هدف با هم به خوبی مراد و تعامل کنند و هم به اندازه کافی مجزا و مستقل باشند که تغییر در یک سطح به سطوح دیگر انتقال پیدا نکند. این سه لایه عبارت است از:

۱- سطح خارجی^۲ ۲- سطح مفهومی^۳ ۳- سطح داخلی^۴

● **سطح خارجی:** همان دید کاربری است. بالاترین سطح انتزاع است. به این ترتیب که فقط بخشی از پایگاه داده که به مسایل کاربر و یا برنامه کاربردی مربوط است را شامل می‌شود.

● **سطح مفهومی:** ساختار پایگاه داده است. تمام موجودیت‌ها^۵ و ارتباطات بین آن‌ها را شامل می‌شود.

● **سطح داخلی:** جزئیات ذخیره‌سازی است. پایین‌ترین سطح انتزاع است و به روش‌های فیزیکی ذخیره و بازیابی نزدیک است. چگونگی ذخیره‌سازی را نشان می‌دهد. ساختارهای داده را توضیح می‌دهد و هم چنین روش‌های دسترسی را نشان می‌دهد. سطح داخلی، سطح فایلینگ است.



۱- ANSI Standard Planning and Requirements Committee

۲- External Level

۳- Conceptual Level

۴- Internal Level

۵- Entity, مانند دانش آموز در محیط عملیاتی مدرسه



نکته
HL^۱: زبان برنامه‌نویسی غیربانکی است. مثل Visual Basic و...
DSL^۲: زبان برنامه‌نویسی بانکی یا به عبارتی زبان فرعی داده‌ای است. مثل SQL و...

۲-۶-۱ مدل داده‌ای: در تعریف پایگاه داده از «مدل داده‌ای» نام برده‌ایم. سیستم مدیریت بانک اطلاعاتی پنهان از دید کاربر، هرگونه دستیابی به داده‌ها را انجام می‌دهد. به این منظور در معماری پایگاه، مدل داده‌ای استفاده شده است. مدل‌های داده‌ای براساس ساختمان داده‌های مختلفی در بانک‌های اطلاعاتی استفاده می‌شوند که سه نوع متداول آن عبارت است از:

۱- رابطه‌ای Relational Data Structure

۲- سلسله مراتبی Hierarchical Data Structure^۲

۳- شبکه‌ای Network Data Structure^۳

در مورد مدل‌های داده‌ای دیگر پژوهش کنید.



همان‌طور که دیدیم، سیستم مدیریت پایگاه داده‌ها، نرم‌افزاری است پیچیده که واسط بین کاربران و محیط فیزیکی ذخیره داده‌ها است. از جمله سیستم‌های مدیریت پایگاه اطلاعات می‌توان از: Oracle، My Sql، SQL Server، Access و... نام برد.

برخی از این سیستم‌های نرم‌افزاری از دیگری پیچیده‌تر، کامل‌تر و بنابراین گران‌تر هستند و برای ذخیره‌سازی داده‌های بسیار زیاد استفاده می‌شوند. به برخی از آن‌ها نیز غیر از مدیریت داده‌ها، بخش‌های دیگری اضافه شده است که واسط کاربری را آرایه می‌کنند. مثلاً در Oracle، بخش Development 2000 جدای از بخش مدیریت بانک عرضه می‌شود که طراح به کمک آن می‌تواند فرم‌ها و گزارش‌ها را ایجاد کند و آسان‌تر با محیط بانک ارتباط برقرار کند. در نرم‌افزار اکسس هم بخش‌هایی چون فرم‌ها، گزارش‌ها، ماکروها و... برای این منظور ایجاد شده است که در بخش‌های بعدی بیشتر با آن آشنا می‌شویم.

۱- Host Language

۲- Data Sub Language

۳ و ۴- درباره مدل‌های سلسله مراتبی و شبکه‌ای در مقاطع بالاتر بیشتر خواهید آموخت.

۱-۷-۱- مفاهیم پایه پایگاه داده رابطه‌ای

۱-۷-۱- محیط عملیاتی^۱ و تعریف موجودیت: محیط عملیاتی، محیطی است که می‌خواهیم یک سیستم ذخیره و بازیابی برای آن ایجاد کنیم. مانند: مدرسه یا دانشگاه، بانک، تعمیرگاه، فروشگاه و...

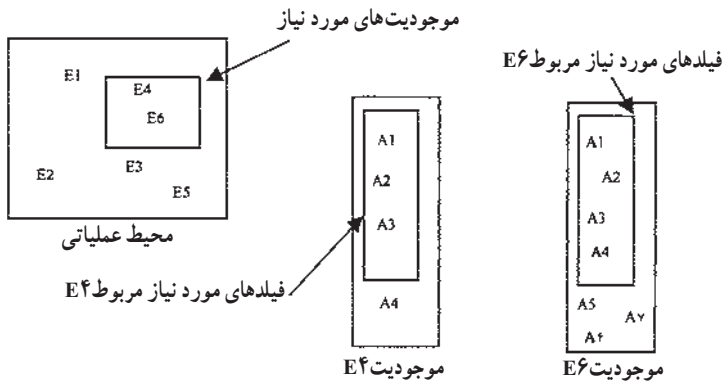
در هر محیط عملیاتی تعدادی موجودیت وجود دارد. موجودیت، هر مفهوم یا شیء در محیط عملیاتی است. مثلاً دانش‌آموز، معلم، کارمند مدرسه، کلاس و درس موجودیت‌های محیط عملیاتی مدرسه هستند. تمام موجودیت‌های یک محیط عملیاتی مهم نیستند بلکه موجودیت‌هایی برای ما مهم هستند که می‌خواهیم در مورد آن‌ها اطلاع داشته باشیم. به‌عنوان یک مثال ساده در محیط عملیاتی مدرسه موجودیت‌های دانش‌آموز، معلم و درس برای سیستم فرضی ثبت‌نام مهم و قابل توجه هستند. در هر محیط عملیاتی مجموعه‌ای از موجودیت‌های مناسب و مرتبط براساس مرز و محدوده سیستم مورد طراحی، انتخاب می‌شوند.

در محیط‌های عملیاتی بانک، خانواده و باشگاه ورزشی چه موجودیت‌هایی قابل تشخیص است؟

تمرین ۱-۱:

پرسش: در چه مواقعی از فیلدهای خود موجودیت برای تعریف کلید استفاده نمی‌شود؟

۱-۷-۲- ویژگی‌های هر موجودیت: صفات و ویژگی‌های هر موجودیت در واقع همان فیلدها هستند. در مورد انتخاب فیلدهای یک موجودیت نیز مانند انتخاب موجودیت‌های محیط عملیاتی فقط صفات مورد نیاز و مرتبط به‌عنوان فیلد در نظر گرفته می‌شوند و تمام صفات یک موجودیت در نظر گرفته نمی‌شوند. به‌عنوان مثال، در محیط عملیاتی مدرسه در مورد اطلاعات مربوط به دانش‌آموز ویژگی‌هایی نظیر قد، وزن، رنگ چشم در نظر گرفته نمی‌شود چرا که در محیط عملیاتی مورد نظر به آن‌ها نیاز نیست، ولی ممکن است در محیط عملیاتی دیگر مناسب و مفید باشند.



شکل ۱-۱- زیر مجموعه موجودیت‌ها و فیلدهای انتخاب شده برحسب نیاز



در نظر گرفتن محدوده محیط عملیاتی قبل از تشخیص موجودیت و ویژگی‌ها چه فایده‌هایی دارد؟

در سه محیط عملیاتی تمرین ۱-۱ موجودیت‌های مرتبط به همراه فیلدهای مورد نیاز را شناسایی کنید.

تمرین ۱-۲:



محیط عملیاتی مدرسه را در نظر بگیرید. موجودیت‌ها و ویژگی‌های هر موجودیت را تعریف کنید.

موجودیت‌ها: دانش آموز، معلم و درس است.

ویژگی‌های موجودیت دانش آموز: شماره دانش آموزی، نام، نام خانوادگی، تلفن، آدرس.

ویژگی‌های موجودیت معلم: نام، نام خانوادگی، تلفن، آدرس.

ویژگی‌های موجودیت درس: شماره درس، نام درس، تعداد واحد، ساعت، کلاس و نام کلاس.

* دقت کنید نام معلم ویژگی درس نیست زیرا تداخل اطلاعاتی رخ می‌دهد، بعدها خواهیم دید چرا مشخصه معلم به درس اضافه می‌شود.

مثال ۱-۱

۳-۷-۱- ارتباط^۱: پیش از این عنوان شد که در محیط عملیاتی، موجودیت‌هایی وجود دارند. معمولاً بین موجودیت‌ها ارتباط یا ارتباط‌هایی وجود دارد که قابل شناسایی است. در محیط عملیاتی مدرسه، بین موجودیت‌های درس، دانش‌آموز و معلم ارتباط‌های زیر قابل تعریف است:

- ۱- معلم درس ارائه می‌کند.
 - ۲- دانش‌آموز هر ترم چندین درس ثبت‌نام می‌کند.
 - ۳- دانش‌آموز برای هر یک از درس‌های ثبت‌نام شده نمره می‌گیرد.
همان‌طور که ملاحظه می‌شود:
- در هر ارتباط موجودیت‌هایی شرکت دارند.
 - هر ارتباط دارای یک مفهوم یا عملکرد است. مثلاً در ارتباط اول «ارائه کردن» عملکرد می‌باشد.
 - هر ارتباط دارای «ماهیتی» خاص است. مثلاً در ارتباط دوم ماهیت ارتباط می‌تواند یک به چند باشد. یعنی یک دانش‌آموز چند درس اخذ می‌کند.
 - برای ایجاد ارتباط، از کلید خارجی استفاده می‌شود. کلید خارجی فیلدی است مشترک بین دو رابطه که در یک رابطه اصلی است و در رابطه‌ای که کلید خارجی محسوب می‌شود، به تنهایی کلید اصلی نیست.
 - ارتباط مانند موجودیت می‌تواند ویژگی‌هایی داشته باشد.



آیا بین سه موجودیت نیز می‌توان ارتباط برقرار کرد؟ مثال بزنید.

۴-۷-۱- انواع ماهیت در ارتباط^۲: تناظر بین موجودیت‌ها، ماهیت ارتباط است و به یکی از سه نوع زیر وجود دارد:

- ۱- یک به یک $1:1$
- ۲- یک به چند $1:N$
- ۳- چند به چند $M:N$

در ارتباط یک به یک، یک نمونه از موجودیت اول فقط با یک نمونه از موجودیت دیگر ارتباط دارد. مثلاً، هر مدرسه یک مدیر دارد. در ارتباط یک به چند، یک نمونه موجودیت با چندین نمونه از موجودیت دیگر مرتبط است. مثلاً یک معلم چند درس ارائه می‌کند. در حالت سوم، یک نمونه از موجودیت اول با چندین نمونه از موجودیت دوم مرتبط است و برعکس. مثلاً یک دانش‌آموز چند درس اخذ می‌کند و یک درس به وسیله چند دانش‌آموز اخذ می‌شود.

مثالی از حالت چند به چند ارائه کنید.

تمرین ۳-۱:



حالتی را مثال بزنید که موجودیت با خودش مرتبط باشد.

در محیط عملیاتی مدرسه، ماهیت ارتباطها به صورت زیر است:

۱- یک معلم چند درس ارائه می‌کند و هر درس به وسیله یک معلم ارائه می‌شود.

(1: n)

۲- دانش‌آموز (در هر ترم) چندین درس ثبت‌نام می‌کند و یک درس به وسیله چندین

دانش‌آموز انتخاب می‌شود. (m: n)

۳- یک دانش‌آموز برای هر درس ثبت‌نام شده، یک نمره می‌گیرد. یک نمره به

یک درس داده می‌شود. (1: n)

مثال ۱-۲

۵-۷-۱- نمودار ارتباط - موجودیت/ER: برای نشان دادن ارتباط بین موجودیتها

و بیان عملکرد ارتباط، از نمودار ER استفاده می‌شود. به عبارت دیگر، یک مدل‌سازی معنایی از داده‌هاست.

اجزای نمودار ER عبارت است از:

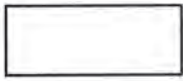
۱- موجودیتها

۲- عملکرد ارتباط

۳- ماهیت ارتباط

۴- ویژگی‌های موجودیت

برای ترسیم نمودار، از شکل‌هایی به صورت زیر استفاده می‌شود^۱:



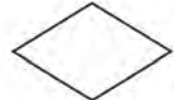
موجودیت



ویژگی موجودیت



ارتباط بین موجودیت‌ها



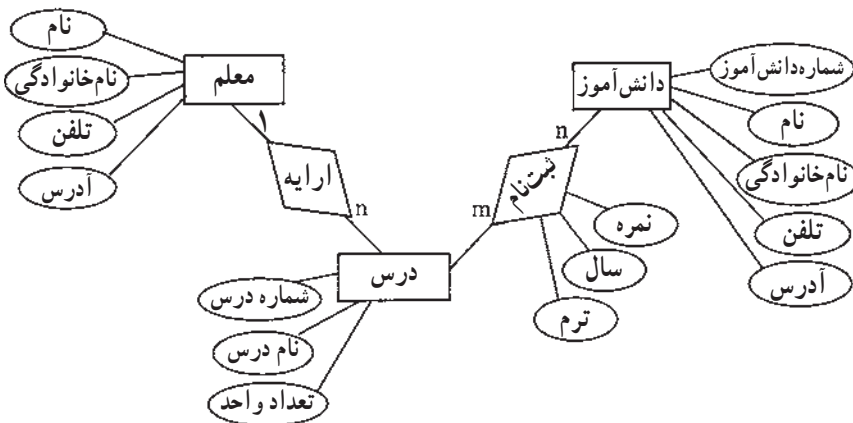
عملکرد

موجودیت‌ها از جنس «اسم» هستند مانند دانش‌آموز و درس. عملکرد ارتباط از جنس «فعل» یا «مصدر» است مانند اخذ کردن. ماهیت ارتباط تناظر بین موجودیت‌ها را نشان می‌دهد، ویژگی‌های موجودیت همان صفات قابل شناسایی هر موجودیت است.



ماهیت ارتباط روی خطوط متصل بین موجودیت‌ها در دو طرف شکل لوزی ذکر می‌گردد.

نمودار ER محیط عملیاتی مدرسه به صورت زیر است:



مثال ۱-۳

۱- شکل‌ها به صورت قراردادی است و در منابع مختلف متفاوت است.

دقت: یک عملکرد خود می تواند مشخصه‌هایی داشته باشد. در عملکرد ثبت نام مشخصه‌های سال، ترم و نمره درس وجود دارد.

۶-۷-۱- رابطه / جدول : مدل رابطه‌ای بر مبنای یک مفهوم ریاضی به نام «رابطه»^۱ تعریف شده است. استحکام نظریه مجموعه‌ها و سهولت پیاده‌سازی آن باعث رواج این مدل داده‌ای شده است. بر این اساس هر بانک اطلاعاتی از تعدادی رابطه یا همان جدول تشکیل می‌شود.

هر جدول می‌تواند نشان‌دهنده یک موجودیت و یا ارتباط بین موجودیت‌ها باشد. ER مثال ۱-۱ را دوباره ملاحظه نمایید. طبق مدل داده‌ای رابطه‌ای جدول‌های زیر را می‌توان در نظر گرفت :

- جدول دانش‌آموز (Student)
- جدول درس (Course)
- جدول معلم (Teacher)

Course		
شماره درس	نام درس	تعداد واحد

Student				
شماره دانش‌آموزی	نام	نام خانوادگی	تلفن	نشانی

Teacher			
نام	نام خانوادگی	تلفن	آدرس

هر جدول با نام خود و مجموعه نام ستون‌هایش مشخص می‌شود. دقت کنید بهتر است در نرم‌افزار اکسس نام جدول و نام ستون‌ها به صورت لاتین باشد ولی در این جا فارسی آورده شده است.

۱- رابطه، زیر مجموعه‌ای از حاصل ضرب دکارتی مجموعه‌ها است. مفاهیم مرتبط در این نظریه به ترتیب عضو، مجموعه حاصل ضرب دکارتی، رابطه و تابع است.

محاسن مدل داده‌ای رابطه‌ای :

- ۱- از نظر نمایش، ساده و قابل فهم است و تنها از یک مفهوم اصلی به نام رابطه یا جدول استفاده می‌کند.
- ۲- انواع ماهیت‌های ارتباط را پشتیبانی می‌کند.
- ۳- مبنای ریاضی دارد.

هنگامی که طراح، موجودیت‌های مورد نیاز و ویژگی‌های آن‌ها را به همراه ارتباط بین موجودیت‌ها شناسایی کرد، کافی است نام و نوع آن‌ها را در حالت جدولی به پایگاه داده معرفی نماید.

۷-۷-۱- تبدیل ER به جدول

- هر موجودیت مستقل به یک جدول تبدیل می‌شود.
- صفات موجودیت‌ها به فیلدهای جدول مربوطه تبدیل می‌شوند.
- موجودیت‌هایی که ارتباط ۱: ۱ دارند به یک جدول تبدیل می‌شوند. [اگرچه ممکن است داخل اطلاعاتی رخ دهد]. در مواردی نیز مانند $n: 1$ یکی از جدول‌ها به دیگری منتقل می‌شود.
- در ارتباط $n: 1$ ، کلید اصلی جدول طرف ۱ به جدول طرف n اضافه می‌شود.
- ارتباط $n: m$ ، تبدیل به یک جدول می‌شود که کلید اصلی دو موجودیت را شامل می‌شود.

۸-۱- قدم‌های طراحی بانک اطلاعاتی

در طراحی هر سیستم اطلاعاتی، صرف‌نظر از روش و متدولوژی طراحی باید ابتدا محیط سیستم شناخته و مطالعه شود، سپس نیازسنجی صورت می‌گیرد یعنی تعیین نیازهای اطلاعاتی - پردازشی، تشخیص محدودیت‌ها و قواعد حاکم بر محیط.

روش‌های مختلفی برای طراحی وجود دارد. با بیان ساده می‌توان گفت که در طراحی بانک داده مدل‌سازی معنایی داده صورت می‌گیرد یعنی ابتدا طراحی مفهومی انجام می‌شود و نمودار ER رسم می‌گردد و در طراحی سیستم اطلاعاتی، معمولاً ابتدا تحلیل فرآیندی صورت می‌گیرد و پردازش‌ها و گردش اطلاعات شناسایی می‌شوند. با مشخص شدن پردازش‌ها، داده‌هایی که از طریق فرم‌ها رد و بدل می‌شوند و تغییراتی که روی آن‌ها انجام می‌شود، مشخص می‌گردند و به این ترتیب ذخایر داده‌ای شکل می‌گیرند و برنامه کاربردی طراحی می‌شود یعنی واسط‌های کاربری مثل فرم‌ها و گزارش‌ها و کنترل‌های

مورد نیاز تعیین می‌شود. در این روش‌ها، پس از انجام طراحی تفصیلی، جدول‌ها، فیلدها و کلید اصلی آن‌ها از نمودار ER و یا ذخایر داده‌ای تعیین می‌شوند. روش تبدیل موجودیت‌ها و رابطه‌ها (نمودار ER) را به جدول‌ها، فیلدها و کلید قبلاً توضیح داده‌ایم.

برای مطالعه بیشتر

در هر روش و متدولوژی طراحی، مراحل انجام طراحی، اصول و مفروضات مورد استفاده، ابزار و نکات کاربردی و محصولات و نتایج حاصل در هر مرحله و همچنین ارتباط بین مراحل، چگونگی عبور از هر مرحله، روش کنترل مستندات حاصل در هر مرحله، و در برخی متدولوژی‌ها ممیزی، کنترل و حساب‌رسی کیفیت مراحل و محصولات هر مرحله مشخص است. به‌طور کلی سه روش وجود دارد:

فرایندگرا (Process Oriented)، داده‌گرا (Data Oriented) و شیء‌گرا (Object Oriented). روش فرایندگرا مانند SSADM، Urdon، Jackson و غیره. داده‌گرا مانند روش Information Engineering (IE) و روش شیء‌گرا مانند RUP (UML).

کنجکاو

یک روش و متدولوژی طراحی نام برده و خصوصیات آن را به‌طور خلاصه بیان کنید.

بحث شناخت سیستم و طراحی فرم‌ها و گزارش‌ها به مبحث تجزیه و تحلیل سیستم‌های اطلاعاتی مرتبط است که خود محدوده وسیعی دارد و از بحث ما خارج است.

با این سرآغاز طراحی یک بانک داده را به قدم‌های زیر ساده می‌کنیم:

- ۱- ابتدا هدف از طراحی پایگاه داده را مشخص کنید.

مشخص کنید چه نوع اطلاعاتی باید از بانک دریافت شود. با افرادی که از بانک استفاده خواهند کرد صحبت کنید و در مورد فرم‌ها و گزارش‌های مورد نیاز، اطلاعات جمع‌آوری کنید.

- ۲- نمودار ER را طراحی کنید.

موجودیت‌ها، ویژگی‌ها و ارتباط بین آن‌ها را شناسایی کنید و نمودار ارتباط - موجودیت‌ها

(ER) را ترسیم کنید.

برای هر موجودیت کلید اصلی را تعیین یا ایجاد کنید.



بمخاطر بسپارید که یک جدول نباید اطلاعات افزونه داشته باشد یعنی هر جدول باید اطلاعات یک موضوع را نگهداری کند و از اختلاط اطلاعاتی پرهیز شود.

۳- جدول‌ها و فیلدهای آن‌ها را مشخص کنید.

نمودار ER ترسیم شده را به جدول‌ها، فیلدهای مربوطه تبدیل کنید. بعدها خواهیم دید که جدول ایجاد شده از نظر سطوح نرمال بررسی شده و ممکن است خود به جدول‌های جدید تجزیه شوند.

۴- فرم‌های مورد نیاز را تعریف کنید.

فرم‌های ورود، ویرایش و حذف داده را رسم کنید و از وجود فیلدهای فرم در جدول‌های بانک مطمئن شوید.

۵- گزارش‌های مورد نیاز را تعریف کنید.

گزارش‌های مورد نیاز کاربران بانک را معین کرده، قالب نمایش را تعیین کنید. وجود داده‌های لازم و مورد نیاز نمایش گزارش در جدول‌های بانک را بررسی نمایید.

۶- طراحی خود را اصلاح و بازنگری کنید.

یافته‌های خود را با افرادی که از بانک استفاده می‌کنند مطرح کنید و بار دیگر نیازها را بررسی و طراحی خود را اصلاح و بازنگری نمایید.



تفاوت‌های طرح خود را با طراحی مثال بعد بررسی و تحلیل کنید.



دقت کنید که ممکن است طراحی شما با جدول‌های زیر متفاوت باشد. این مسأله نشان‌دهنده غلط بودن طراحی نیست. بلکه برحسب مفروضات، سطح نرمال بودن جدول‌ها و محدوده سیستم کاملاً طبیعی است.

۱- با نرمال‌سازی در فصل ۷ آشنا خواهید شد.

محیط عملیاتی مدرسه را طراحی کنید.

نمودار ER رسم شده است.

۱- نمودار ER به جدول‌های زیر تبدیل می‌شود:

شکل [] نشان‌دهنده کلید اصلی و شکل () نشان‌دهنده کلید خارجی اضافه شده از جدول دارای ارتباط است.

جدول معلم Tbl-Teacher

Teacher Code	Teacher FName	Teacher LName	Teacher Tel	TeacherAddress
[کد معلم]	نام	نام خانوادگی	تلفن	آدرس

جدول دانش‌آموز Tbl-Student

Student Code	Student FName	Student LName	Student Tel	StudentAddress	Year
[شماره دانش‌آموزی]	نام	نام خانوادگی	تلفن	آدرس	سال

* در جدول درس، فیلد کد معلم را که کلید اصلی در جدول معلم است به عنوان کلید خارجی اضافه می‌کنیم.

جدول درس Tbl-Course

Course Code	Course FName	Course Unit	Teacher Code
[شماره درس]	نام درس	تعداد واحد	(کد معلم)

* ارتباط ۱: n بین درس و معلم، کلید جدول معلم (کد معلم) به جدول درس اضافه می‌شود. دقت کنید که کد معلم در جدول درس بخشی از کلید اصلی نیست و کلید خارجی به حساب می‌آید.

جدول ثبت نام Tbl-Register

Year	Term	Course Code	Student Code	CourseGrade
سال	ترم	شماره درس	شماره دانش آموز	نمره درس

* ارتباط $n:m$ بین دانش آموز و درس به جدول ثبت نام تبدیل می شود و کلید اصلی جدول درس (شماره درس) و کلید اصلی جدول دانش آموز (شماره دانش آموز) به آن اضافه می شود.

چون ارتباط $n:m$ مفروض خود دارای ویژگی هایی است پس این دو کلید به تنهایی کلید اصلی آن جدول نیستند و ترکیب آن دو با فیلد سال و ترم جدول کلید اصلی شده است.

با توجه به اصول و مفروضات طراحی که در بخش نرمال سازی خواهیم دید، جدول ثبت نام به دو جدول زیر تبدیل می شود:

جدول ثبت نام Tbl-Register

Register Year	Term	Student Code	Register Code
سال	ترم	شماره دانش آموز	سریال ثبت نام

* ما شماره سریال ثبت نام را به عنوان کلید اصلی اضافه می کنیم. می توانستیم سال و ترم را هم کلید بگیریم ولی بهتر است از کلید ترکیبی به علت پیچیدگی هایی که ایجاد می کنند استفاده نکنیم.

جدول دروس ثبت نام شده Tbl-Register

Register Code	Course Code	Grade
سریال ثبت نام	شماره درس	نمره درس

* دقت کنید که در جدول دروس ثبت نام شده شماره درس بخشی از کلید اصلی است.

۲- فرم‌های مورد نیاز :

فرم تعریف معلم

کد معلم :
 نام :
 نام خانوادگی :
 تلفن :
 آدرس :

فرم تعریف دانش‌آموزان

شماره دانش‌آموز :
 نام :
 نام خانوادگی :
 تلفن :
 آدرس :
 سال :

فرم تعریف درس

شماره درس :
 نام درس :
 تعداد واحد :
 کد معلم :

فرم ثبت نام

سال : ترم : نام دانش‌آموز :

شماره درس	نام درس	تعداد واحد

فرم ورود نمره

سال : ترم : نام دانش‌آموز :

شماره درس	نام درس	تعداد واحد	نمره نهایی

۳- گزارش‌های مورد نیاز :

گزارش ثبت‌نام

سال : _____
ترم : _____
نام دانش‌آموز : _____ شماره دانش‌آموز : _____
دروس ثبت‌نام شده

شماره درس	نام درس	تعداد واحد	نام معلم

گزارش لیست معلمان مدرسه

کد معلم	نام معلم	تلفن	آدرس

گزارش لیست حضور و غیاب

سال : _____
ترم : _____
نام معلم : _____ درس : _____
تعداد واحد : _____ ساعت تشکیل کلاس : _____

شماره دانش‌آموز	نام دانش‌آموز	حضور و غیاب																
		1	2	3	4	5	6	7	8	9	10	11	12					

گزارش کارنامه دانش‌آموزان

نام و نام خانوادگی : _____
تاریخ گزارش : _____
رشته : _____ سال : _____ ترم : _____

شماره درس	نام درس	تعداد واحد	نمره

جمع واحد : _____
معدل : _____

خلاصه فصل

سیستم ذخیره و بازیابی از مفهوم ساده فایلینگ توسعه پیدا کرده است. فیلد کوچک‌ترین واحد ذخیره داده است. رکورد مجموعه فیلدهای مرتبط است و مجموعه رکوردها فایل را تشکیل می‌دهند. این مفاهیم پایه‌ای در هر سیستم اطلاعاتی دیگر نیز مطرح‌اند. فیلد یا زیرمجموعه‌ای از فیلدها که باعث یکتایی رکورد شوند را کلید گویند. در سیستم‌های ذخیره و بازیابی، عملیات مشخصی روی رکوردها انجام می‌گیرد که عبارت است از بازیابی، درج، حذف، به‌هنگام‌سازی و تغییر ساختار. این عملیات را در محیط Access بیشتر بررسی می‌کنیم.

برای جلوگیری از درگیر شدن کاربر با مسایل برنامه‌نویسی و انجام سریع‌تر و دقیق‌تر تغییرات و در مجموع برای برقرار کردن ارتباط بین حجم بالای داده‌های ذخیره شده روی رسانه و کاربران معماری سه لایه‌ای مطرح شده است: سطح خارجی، سطح مفهومی، سطح داخلی.

نرم‌افزاری که این ارتباط را برقرار، مدیریت و اداره می‌کند، سیستم مدیریت بانک اطلاعاتی نام دارد که برای مدیریت دستیابی به داده‌ها و پنهان کردن آن از دید کاربران از یک مدل مفهومی به نام مدل داده‌ای استفاده می‌کند. مدل رابطه‌ای از جمله مدل‌های داده‌ای مهم است که بر پایه مفهوم ریاضی مجموعه‌ها بنا شده است و بیشتر سیستم‌های مدیریت بانک اطلاعاتی رایج از آن پشتیبانی می‌کنند.

در این فصل مفاهیم پایه مدل داده رابطه‌ای مانند: رابطه/جدول، ارتباط، ماهیت ارتباط، نمودار ارتباط - موجودیت (ER) و روش تبدیل این نمودار به جدول‌های پایگاه داده مطرح شده است.

۱- دلایل متغیر شدن طول رکورد چیست؟

۲- معایب روش فایلینگ چیست؟

۳- کدام یک از اجزای محیط ذخیره و بازیابی بیشتر اهمیت دارد؟

چرا؟

۴- هدف از ایجاد سیستم ذخیره و بازیابی چیست؟

۵- کلید را تعریف کنید و انواع آن را با ذکر مثال توضیح دهید.

۶- معماری پایگاه داده را توضیح دهید.

۷- مدل داده‌ای را تعریف کنید و توضیح دهید.

۸- ماهیت ارتباط را تعریف کنید و در یک محیط عملیاتی مثال

بزنید.

۹- مراحل طراحی پایگاه داده را توضیح دهید.

۱۰- لزوم طراحی قبل از پیاده‌سازی و شروع کار با نرم‌افزار Access

را بیان کنید و مزایا و معایب احتمالی آن را بررسی کنید.