
اصول طراحی پایگاه داده‌ها

تالیف:

مهندس رمضان عباس نژادورزی
مهندس علی‌رضا عظیمی – مهندس باقر رحیم‌پور کامی



فن‌آوری نوین

سرشناسه	: عباس نژاد ورزی، رمضان، ۱۳۴۸ -
عنوان و نام پدیدآور	: اصول طراحی پایگاه داده‌ها/تالیف رمضان عباس نژادورزی، باقر رحیمپور کامی، علی رضا عظیمی.
مشخصات نشر	: بابل: فن آوری نوین، ۱۳۹۰
مشخصات ظاهری	: ۲۵۶ص: مصور، جدول
شابک	: ۷۳۰۰۰ ریال: 978-600-9141-39-5
وضعیت فهرست نویسی	: فیپا
موضوع	: سرور اس. کیو. ال
موضوع	: پایگاه‌های اطلاعاتی -- طراحی
شناسه افزوده	: رحیمپور کامی، باقر، ۱۳۶۰ -
شناسه افزوده	: عظیمی، علیرضا، - ۱۳۵۹
رده بندی کنگره	: ۱۳۸۹ ۱۶۶ع۲ پ/۷۶/۹QA
رده بندی دیویی	: ۰۰۵/۷۴
شماره کتابشناسی ملی	: ۲۲۴۰۶۸۶



www.fanavarienovin.net

تلفن: ۰۱۱۱-۲۲۵۶۶۸۷

بابل، کدپستی ۴۷۱۶۷-۷۳۴۴۸

فن آوری نوین

اصول طراحی پایگاه داده‌ها

تألیف: مهندس رمضان عباس نژادورزی - مهندس باقر رحیمپور کامی - مهندس علی رضا عظیمی

نوبت چاپ: چاپ اول

سال چاپ: بهار ۱۳۹۰

شمارگان: ۱۰۰۰ جلد

قیمت: ۷۳۰۰ تومان

نام چاپخانه و صحافی: فرنگاررنگ

شابک: ۹۷۸ - ۶۰۰ - ۹۱۴۱۳ - ۹ - ۵

نشانی ناشر: بابل، چهارراه نواب، کاظم بیگی، جنب حسینیه منصور کاظم بیگی، طبقه همکف

طراح جلد: کانون آگهی و تبلیغات آبان (احمد فرجی)

تهران، خ اردیبهشت، نبش وحید نظری، پلاک ۱۴۲ تلفکس: ۶۶۴۰۰۱۴۴-۶۶۴۰۰۲۲۰

فهرست مطالب

۳۲ - ۱۰ - ۲ . ۱. دستیابی به داده‌های تثبیت نشده ۳۲	فصل اول: آشنایی با مفاهیم اولیه بانک اطلاعات
۳۲ - ۱۰ - ۳ . ۱. بازیابی ناسازگار..... ۳۲	۱ - ۱ . دلایل نیاز به بانک اطلاعات ۹
۳۲ - ۱۱ . ۱. استقلال داده‌ها..... ۳۲	۱ - ۲ . اجزای تشکیل دهنده بانک اطلاعاتی ۱۲
۳۴ - ۱۲ . ۱. پرسش‌های چهارگزینه‌ای ۳۴	۱ - ۲ - ۱ . سخت‌افزار ۱۲
۳۸ - ۱۳ . ۱. پاسخ پرسش‌های چهارگزینه‌ای ۳۸	۱ - ۲ - ۲ . داده‌ها..... ۱۲
فصل دوم: مدل‌سازی داده‌ها با استفاده از مدل ER	۱ - ۲ - ۳ . کاربردان ۱۴
۴۱ - ۲ - ۱ . مدل ER ۴۱	۱ - ۲ - ۴ . نرم‌افزار ۱۴
۴۲ - ۱ - ۱ . ۲. موجودیت ۴۲	۱ - ۳ . ۱. انواع بانک‌های اطلاعاتی ۱۵
۴۳ - ۱ - ۲ . ۲. صفات ۴۳	۱ - ۳ - ۱ . بانک‌های اطلاعاتی هرمی (سلسله مراتبی) ۱۵
۴۶ - ۱ - ۳ . ۲. نوع ارتباط ۴۶	۱ - ۳ - ۲ . بانک اطلاعاتی شبکه‌ای ۱۵
۵۱ - ۲ - ۲ . ۲. توسعه نمودار ER ۵۱	۱ - ۳ - ۳ . بانک اطلاعاتی رابطه‌ای ۱۶
۵۱ - ۲ - ۱ . ۲. تخصیص ۵۱	۱ - ۳ - ۴ . مدل بانک اطلاعاتی شی‌گرا ۱۷
۵۱ - ۲ - ۲ . ۲. تعمیم ۵۱	۱ - ۴ . انواع مختلف بانک‌های اطلاعاتی از لحاظ جغرافیایی ۱۸
۵۲ - ۲ - ۳ . ۲. ارث‌بری صفات ۵۲	۱ - ۵ . سطوح مختلف بانک اطلاعاتی ۲۰
۵۲ - ۲ - ۴ . ۲. تجمع ۵۲	۱ - ۵ - ۱ . سطح خارجی ۲۱
۵۴ - ۲ - ۳ . ۲. طراحی مدل مفهومی با مدل ER ۵۴	۱ - ۵ - ۲ . سطح ادراکی ۲۳
۵۶ - ۲ - ۴ . ۲. مسائل حل شده ۵۶	۱ - ۵ - ۳ . سطح داخلی ۲۳
۶۱ - ۲ - ۵ . ۲. پرسش‌های چهارگزینه‌ای ۶۱	۱ - ۶ . زبان میزبان ۲۴
۶۴ - ۲ - ۶ . ۲. پاسخ پرسش‌های چهارگزینه‌ای ۶۴	۱ - ۶ - ۱ . زبان تعریف داده ۲۴
فصل سوم: مدل رابطه‌ای و جبر رابطه‌ای	۱ - ۶ - ۲ . زبان دستکاری داده ۲۴
۶۵ - ۳ - ۱ . مفاهیم اولیه مدل رابطه‌ای ۶۵	۱ - ۶ - ۳ . زبان کنترل داده ۲۴
۶۵ - ۳ - ۱ - ۱ . دامنه ۶۵	۱ - ۷ . جامعیت ۲۵
۶۶ - ۳ - ۱ - ۲ . رابطه ۶۶	۱ - ۸ . امنیت ۲۵
۶۶ - ۳ - ۱ - ۳ . تاپل ۶۶	۱ - ۹ . وظایف سیستم مدیریت بانک اطلاعاتی ۲۵
۶۶ - ۳ - ۱ - ۴ . درجه رابطه ۶۶	۱ - ۱۰ . مشکلات همروندی یا همزمانی ۳۱
۶۶ - ۳ - ۱ - ۵ . کاردینالیته رابطه ۶۶	تراکنش‌ها ۳۱
۶۷ - ۳ - ۱ - ۶ . کلید ۶۷	۱ - ۱۰ - ۱ . تغییرات مفقود شده ۳۱
۶۸ - ۳ - ۲ . جبر رابطه‌ای ۶۸	
۷۳ - ۳ - ۱ . عملگر تصویر یا پرتو (II) ۷۳	
۷۳ - ۳ - ۲ . عملگر انتخاب یا گزینش (σ) ۷۳	

- ۱۲۷..... ۴- ۵- ۵- ۴. سطح نرمال 4NF
- ۱۳۰..... ۴- ۵- ۵- ۳. سطح نرمال پنجم 5NF
- ۱۳۱..... ۴- ۶. مراحل نرمال سازی جداول
- ۱۳۳..... ۴- ۷. مسائل حل شده
- ۱۳۷..... ۴- ۸. پرسش های چهارگزینه ای
- ۱۴۲..... ۴- ۹. پاسخ پرسش های چهارگزینه ای

فصل پنجم: دستورات SQL

- ۱- ۵- ۱. ورود به بانک اطلاعاتی SQL Server ... ۱۴۶
- ۲- ۵- ۱. تایپ و اجرای دستورات SQL ۱۴۷
- ۳- ۵- ۱. ایجاد بانک اطلاعاتی ۱۴۷
- ۱- ۳- ۵- ۱. تغییر خواص بانک اطلاعاتی موجود ۱۴۹
- ۲- ۳- ۵- ۱. حذف بانک اطلاعاتی موجود ۱۴۹
- ۴- ۵- ۱. ایجاد و تغییر ساختار جدول ۱۵۰
- ۱- ۴- ۵- ۱. ایجاد جدول ۱۵۳
- ۲- ۴- ۵- ۱. تغییر ساختار جدول با دستور SQL ۱۵۴
- ۳- ۴- ۵- ۱. حذف جدول با دستور SQL ۱۵۵
- ۵- ۵- ۱. دستورات SQL برای ورود، ویرایش و حذف داده ها ۱۵۶
- ۱- ۵- ۵- ۱. دستور INSERT ۱۵۶
- ۲- ۵- ۵- ۱. ویرایش رکوردهای جدول ۱۵۷
- ۳- ۵- ۵- ۱. حذف رکوردهای جدول ۱۵۸
- ۴- ۵- ۵- ۱. اضافه کردن رکوردها به جداول ۱۵۸
- ۵- ۵- ۱. پرس و جوی بازیابی اطلاعات ۱۵۹
- ۶- ۵- ۱. عملگرها در SQL ۱۵۹
- ۱- ۶- ۵- ۱. عملگرهای محاسباتی ۱۶۰
- ۲- ۶- ۵- ۱. عملگر انتساب ۱۶۰
- ۳- ۶- ۵- ۱. عملگرهای بیتی ۱۶۱
- ۴- ۶- ۵- ۱. عملگرهای رابطه ای ۱۶۱
- ۵- ۶- ۵- ۱. عملگرهای منطقی ۱۶۱
- ۶- ۶- ۵- ۱. عملگرهای یکانی ۱۶۲
- ۷- ۶- ۵- ۱. عملگرهای اتصال رشته ای ۱۶۲

- ۳- ۲- ۳- ۳. عملگر اجتماع (U) ۷۴
- ۴- ۲- ۳- ۳. عملگر اشتراک (∩) ۷۶
- ۵- ۲- ۳- ۳. عملگر تفاضل (-) ۷۶
- ۶- ۲- ۳- ۳. عملگر ضرب دکارتی (x) ۷۸
- ۷- ۲- ۳- ۳. پیوند ۷۹
- ۸- ۲- ۳- ۳. توابع تجمیع ۸۸
- ۹- ۲- ۳- ۳. افزودن رکورد به جدول ۸۹
- ۱۰- ۲- ۳- ۳. حذف رکوردهای جدول ۸۹
- ۱۱- ۲- ۳- ۳. بروزرسانی اطلاعات جداول ۹۰
- ۱۲- ۲- ۳- ۳. عملگر تغییر نام ρ ۹۰
- ۱۳- ۲- ۳- ۳. عملگر جایگزینی (←) ۹۰
- ۱۴- ۲- ۳- ۳. عملگر پرتو توسعه یافته ۹۱
- ۱۵- ۲- ۳- ۳. عملگر تقسیم (÷) ۹۲
- ۱۶- ۲- ۳- ۳. بهینه سازی پرس و جو ۹۵
- ۳- ۳- ۳- ۳. حساب رابطه ای ۹۸
- ۱- ۳- ۳- ۳. حساب رابطه ای دامنه ای ۹۹
- ۴- ۳- ۳- ۳. مسائل حل شده ۱۰۱
- ۵- ۳- ۳- ۳. پرسش های چهارگزینه ای ۱۰۸
- ۶- ۳- ۳- ۳. پاسخ پرسش های چهارگزینه ای ۱۱۴

فصل چهارم: وابستگی های تابعی و نرمال سازی رابطه ها

- ۱- ۴- ۱. دلایل نرمال سازی رابطه ها ۱۱۷
- ۲- ۴- ۱. وابستگی تابعی ۱۱۸
- ۳- ۴- ۱. قوانین استنتاج وابستگی های تابعی ۱۲۰
- ۴- ۴- ۱. مجموعه وابستگی کاهش ناپذیر ۱۲۱
- ۵- ۴- ۱. سطوح نرمال سازی ۱۲۳
- ۱- ۴- ۵- ۱. سطح نرمال سازی 1NF ۱۲۳
- ۲- ۴- ۵- ۱. سطح نرمال 2NF ۱۲۴
- ۳- ۴- ۵- ۱. سطح نرمال 3NF ۱۲۵
- ۴- ۴- ۵- ۱. سطح نرمال BCNF ۱۲۶

۱۸۴.....	IF ... ELSE دستور	۵-۱۸-۱	۱۶۲.....	عملگرهای ویژه	۵-۶-۸
۱۸۴.....	IF تو در تو دستور	۵-۱۸-۲	۵-۷	بازیابی اطلاعات از جدول با دستور	
۱۸۵.....	CASE دستور	۵-۱۸-۳	۱۶۳.....	SELECT	
۱۸۵.....	رویه‌های ذخیره شده	۵-۱۹	۵-۸	مرتب سازی رکوردها (تاپل)	۱۶۶.....
۱۸۶.....	ایجاد رویه‌های ذخیره شده	۵-۱۹-۱	۵-۹	پرس وجوهای مرکب	۱۶۷.....
۱۸۷.....	اجرای رویه ذخیره شده	۵-۱۹-۲	۵-۹-۱	عملگر UNION	۱۶۷.....
۱۸۸.....	تغییر رویه ذخیره شده	۵-۱۹-۳	۵-۹-۲	عملگر UNION ALL	۱۶۷.....
۱۸۸.....	حذف رویه‌های ذخیره شده	۵-۱۹-۴	۵-۹-۳	عملگر INTERSECT	۱۶۸.....
۱۸۸.....	مجوز در SQL	۵-۲۰	۵-۹-۴	عملگر EXCEPT	۱۶۸.....
۱۸۹.....	GRANT دستور	۵-۲۰-۱	۵-۱۰	توابع تجمعی	۱۷۰.....
۱۹۰.....	REVOKE دستور	۵-۲۰-۲	۵-۱۱	گروه‌بندی اطلاعات	۱۷۰.....
۱۹۱.....	مسائل حل شده	۵-۲۱	۵-۱۲	تست مقدار تهی فیلد	۱۷۱.....
۲۰۰.....	پرسش‌های چهار گزینه‌ای	۵-۲۲	۵-۱۳	پرس وجوی فرعی	۱۷۲.....
۲۱۳.....	پاسخ پرسش‌های چهارگزینه‌ای	۵-۲۳	۵-۱۴	پیوند جداول (رابطه)	۱۷۵.....
۲۱۷.....	پیوست الف: مسائل تکمیلی		۵-۱۴-۱	پیوند ضربدری	۱۷۶.....
	پیوست ب: پرسش‌های چهارگزینه‌ای امتحان پیام نور		۵-۱۴-۲	پیوند متعادل	۱۷۶.....
۲۲۷.....			۵-۱۴-۳	پیوند نامتعادل	۱۷۷.....
۲۵۳.....	پیوست ج: محاسبات رابطه‌ای		۵-۱۴-۴	پیوند درونی	۱۷۸.....
۲۵۶.....	منابع:		۵-۱۴-۵	پیوند بیرونی	۱۷۹.....
			۵-۱۵	دید	۱۸۰.....
			۵-۱۵-۱	ایجاد دید	۱۸۱.....
			۵-۱۵-۲	تغییر دید	۱۸۲.....
			۵-۱۵-۳	حذف دید	۱۸۲.....
			۵-۱۶	متغیرها	۱۸۲.....
			۵-۱۷	توضیحات	۱۸۳.....
			۵-۱۸	ساختارهای تصمیم	۱۸۳.....

دیگر آثار مولف

انتشارات	نام کتاب	انتشارات	نام کتاب
علوم رایانه	آموزش گام به گام Crystal Report	فن آوری نوین	حل مسائل C (مرجع کامل)
علوم رایانه	آشنایی با شبکه GSM	فن آوری نوین	حل مسائل C++ (مرجع کامل)
علوم رایانه	آموزش گام به گام سیستم عامل لینوکس	فن آوری نوین	آموزش گام به گام برنامه نویسی بانک اطلاعات با C# (مرجع کامل)
علوم رایانه	خود آموز اکسس	فن آوری نوین	حل مسائل C# (مرجع کامل)
علوم رایانه	آموزش گام به گام Word	فن آوری نوین	حل مسائل پاسکال (مرجع کامل)
علوم رایانه	آموزش گام به گام اکسل	فن آوری نوین	آموزش گام به گام برنامه نویسی بانک اطلاعات با ویژوال بیسیک نت (مرجع کامل)
علوم رایانه	ICDL مهارت ۱: مفاهیم پایه اطلاعات	فن آوری نوین	آموزش گام به گام LINQ با C#
علوم رایانه	ICDL مهارت ۲: به کارگیری کامپیوتر و مدیریت	فن آوری نوین	تجارت الکترونیکی
علوم رایانه	ICDL مهارت ۳: واژه پردازی به کمک کامپیوتر	فن آوری نوین	امنیت شبکه
علوم رایانه	ICDL مهارت ۴: صفحات گسترده	علوم رایانه	برنامه نویسی با دلفی
علوم رایانه	ICDL مهارت ۵: پایگاه داده	علوم رایانه	آموزش گام به گام دلفی نت
علوم رایانه	ICDL مهارت ۶: ارائه مطلب	علوم رایانه	آموزش گام به گام C#.NET
علوم رایانه	ICDL مهارت ۷: اطلاعات و ارتباطات	علوم رایانه	آموزش گام به گام VisualC++.NET
علوم رایانه	آموزش گام به گام FLASH MX	علوم رایانه	آموزش گام به گام برنامه نویسی با ویژوال C++
علوم رایانه	درس و کنکور برنامه نویسی به زبان C	علوم رایانه	آموزش گام به گام J#.NET
علوم رایانه	برنامه سازی سیستم	علوم رایانه	آموزش گام به گام SQL Server
علوم رایانه	پرسش های چهار گزینه ای پاسکال	علوم رایانه	آموزش گام به گام SQL
علوم رایانه	آموزش گام به گام VC++	علوم رایانه	رهیافت و پرسش های چهار گزینه ای C
علوم رایانه	کارور رایانه ۱	علوم رایانه	رهیافت و پرسش های چهار گزینه ای C++
علوم رایانه	کارور رایانه ۲	علوم رایانه	تست و پرسش های چهار گزینه ای C
علوم رایانه	آموزش گام به گام برنامه ویژوال بیسیک نت	علوم رایانه	مبانی فناوری اطلاعات
علوم رایانه	برنامه نویسی به زبان اسمبلی	علوم رایانه	آموزش گام به گام برنامه ویژوال بیسیک

مقدمه

امروزه حجم زیادی از داده ذخیره، پردازش و بازیابی می‌شوند. برای جلوگیری از افزونگی داده (تکرار بی-مورد داده‌ها)، بی‌نظمی و ایجاد سازگاری بین گزارش‌ها از پایگاه داده (بانک اطلاعات) استفاده می‌شود. از آنجائی که در سیستم‌های امروزی (به ویژه سیستم‌های تجارت الکترونیک، داده کاوی و پشتیبانی تصمیم) داده‌ها و بانک اطلاعاتی جایگاه بسیار مهمی دارند، درس پایگاه داده از دروس تخصصی رشته‌های مهندسی کامپیوتر و فناوری اطلاعات در نظر گرفته شده است. به همین دلیل سعی گردیده، کتاب حاضر تدوین شود. تمام مطالب کتاب با توجه به سر فصل مصوب وزارت علوم و تحقیقات، به زبانی ساده و روان براساس کتب آقایان دیت، سیلورشاتس و ... بیان گردید.

کتاب حاضر دارای ۵ فصل و سه پیوست می‌باشد. هر فصل کتاب از سه بخش تشکیل شده است که عبارت‌اند از:

۱. تشریح مفاهیم مربوط به آن فصل به همراه مثال‌های متعدد.

۲. بیان مسائل اضافی و حل آن‌ها.

۳. بیان تست‌های ارشد سال‌های قبل دانشگاه دولتی و آزاد و حل تشریحی آن‌ها

در فصل اول، مفاهیم اولیه بانک اطلاعات، مزایا، معایب بانک اطلاعات و غیره بیان گردید. در فصل دوم، مدل ER و چگونگی پیاده‌سازی آن بحث و بررسی گردید، در فصل سوم، مفاهیم رابطه‌ای، جبر رابطه و حساب دامنه‌ای شرح داده شده است. در فصل چهارم، وابستگی‌های تابعی و نرمال‌سازی بیان گردیده است و در فصل پنجم، دستورات SQL به طور کامل بیان شده، در محیط SQL تست گردیده و خروجی آن‌ها در کتاب آمده است. در پیوست الف، مسائل تکمیلی با حل آن آمده است. در پیوست ب، چند دوره سوالات تست دانشگاه پیام‌نور با جواب تشریحی آورده شده است و در پیوست ج، جبر تاپلی و دامنه‌ای بحث گردید. کتابی که در پیش رو دارید، با بهره‌گیری از سال‌ها تجربه در امر تدریس (در درس پایگاه داده) و تالیف کتب کامپیوتری تدوین گردیده است.

امیدواریم این اثر نیز مورد توجه اساتید و دانشجویان عزیز واقع شود.

در پایان از تمامی خوانندگان عزیز (اساتید و دانشجویان) تقاضا داریم، هرگونه اشکال، ابهام در متن کتاب، پیشنهادات و انتقادات‌شان را به آدرس پست الکترونیک fanavarienovin@yahoo.com ارسال نمایند.

بابل، زمستان ۱۳۸۹

مولفین

www.fanavarienovin.net

آشنایی با مفاهیم اولیه بانک اطلاعات

سیستم مدیریت بانک اطلاعات، مکانیزم نگهداری رکوردها^۱ است. یعنی، بانک اطلاعات مخزنی برای نگهداری داده‌ها است که کاربران می‌توانند اعمال زیر را در آن انجام دهند:

۱. افزودن جداول خاص به بانک اطلاعات

۲. افزودن رکوردهایی به جداول بانک اطلاعات

۳. تغییر ساختار جداول

۴. حذف جدول

۵. تغییر رکوردهای بانک اطلاعات

۶. حذف رکوردهای بانک اطلاعات

۷. اجرای پرس‌وجو^۲ بر روی جداول

به عبارت ساده‌تر، سیستم مدیریت بانک اطلاعات (DBMS)^۳، سیستم کامپیوتری است که هدف آن ذخیره و بازیابی داده‌ها می‌باشد. بانک اطلاعات داده را پردازش نموده به اطلاعات تبدیل کرده، آن‌ها را ذخیره و بازیابی می‌نماید.

۱-۱. دلایل نیاز به بانک اطلاعات

قبل از این که به دلایل نیاز به بانک اطلاعاتی بپردازیم، مشکلات سیستم‌های سنتی ذخیره و بازیابی اطلاعات را می‌آموزیم.

در سیستم‌های سنتی، داده‌ها به صورت فایل‌ها بر روی دیسک سخت نگهداری می‌گردند و برنامه‌هایی نوشته می‌شوند تا داده‌ها را به فایل اضافه کنند، داده‌های موجود را ویرایش نمایند، داده‌های موجود را حذف کنند و مهم‌تر از همه این اعمال، بازیابی داده‌های موجود را انجام دهند. روش‌های پردازش فایل سنتی دارای مشکلات زیر است:

۱. **افزایش افزونگی داده**، چون در سیستم ذخیره و بازیابی اطلاعات از طریق فایل‌های سنتی، داده به مرور جمع‌آوری می‌شوند. بنابراین، ممکن است داده‌ها در فایل‌ها پخش شوند. در نتیجه، داده‌ها در فایل‌های مختلف تکرار می‌گردند. برخی از این تکرارها می‌توانند حذف شوند (به تکرار بی‌مورد داده‌ها **افزونگی داده** می‌گویند).

¹.Records

².Query

³. Database Management System

یکی از ویژگی‌های بانک اطلاعات به اشتراک گذاری و مجتمع کردن داده‌ها است. به اشتراک گذاری و مجتمع کردن داده موجب کاهش افزونگی داده می‌گردد.

۲. **افزایش بی‌نظمی و ناسازگاری داده‌ها**، از آنجایی که داده‌ها در فایل‌های مجزا ذخیره می‌گردند و ممکن است داده‌ها در چند فایل تکرار شوند، تغییر داده‌های تکراری شاید موجب بی‌نظمی و ناسازگاری گردد. به عنوان مثال، اطلاعات آدرس دانشجو در فایل‌های عمومی، حسابداری، پرونده پزشکی و غیره تکرار خواهند شد. با تغییر آدرس یک دانشجو، ممکن است تغییر آدرس در بعضی از این فایل‌ها اعمال شود، و برخی دیگر اعمال نگردد. این امر موجب بی‌نظمی و ناسازگاری در فایل‌ها خواهد شد. بانک اطلاعاتی با به اشتراک گذاری داده‌ها بی‌نظمی و ناسازگاری را کاهش می‌دهد. به عنوان مثال، اگر بخواهید آدرس دانشجو را تغییر دهید، کافی است این تغییر را در یک نقطه اعمال کنید. چون، آدرس در بانک اطلاعات در یک نقطه نگهداری می‌شود.

۳. **پیچیدگی زیاد فایل‌ها**، برای مدیریت فایل‌ها باید برنامه‌هایی نوشته شوند تا اعمالی از قبیل، ویرایش، حذف و بازیابی رکوردها را انجام دهند. نوشتن چنین برنامه‌هایی با توجه به نوع دستیابی به فایل پیچیده خواهد شد. به عنوان مثال، فرض کنید اطلاعات در فایل ترتیبی ذخیره شده است و بخواهید رکورد خاصی را حذف کنید. برای این منظور، باید فرآیند زیر را انجام دهید:

۱. فایلی که رکوردی از آن را می‌خواهید حذف کنید، باز نمایید.

۲. فایل جدیدی ایجاد کنید.

۳. تمام رکوردها به جزء رکوردی که باید حذف شود، از ابتدای فایل اول خوانده، در فایل دوم بنویسید.

۴. فایل اول را حذف کنید.

۵. فایل جدید ایجاد شده را به نام فایل اول تغییر نام دهید.

این فرآیندها برای به روزرسانی و اضافه کردن داده‌ها در فایل نیز وجود دارد. بانک اطلاعاتی، **نرم‌افزار سیستم مدیریت بانک اطلاعات (DBMS)** را ارائه کرده است. این نرم‌افزار مدیریت بر داده را آسان نموده است.

🚩 **افزایش مشکلات جامعیت داده**، محیط‌های پردازش فایل سنتی، اجازه تعریف قیود جامعیت را نمی‌دهند. به عنوان مثال، تست نمی‌کند نمره خارج از بازه 0 تا 20 نباشد یا بررسی نمی‌کند مانده حسابی منفی نباشد. برای تعریف هر یک از این قیود باید کد برنامه تغییر یابد. اگر قید جامعیت جدیدی بخواهید تعریف کنید باید این قید را به کد برنامه اضافه نمایید. انجام این کار مشکل می‌باشد. به ویژه، وقتی قیود جامعیت باید طوری اعمال گردد تا بروی چندین داده در فایل‌های مختلف اعمال شود، مشکل چند برابر خواهد شد. در نرم‌افزار DBMS مدیر بانک اطلاعاتی به راحتی می‌تواند قیود جامعیت را تعریف نماید.

آشنایی با مفاهیم اولیه بانک اطلاعاتی ۱۱

وجود مشکلات امنیتی، هر لحظه ممکن است سیستم‌های کامپیوتری خراب شوند. از طرف دیگر، برخی از کارها به هم وابسته‌اند. به عنوان مثال، هنگام انتقال پول از حساب A به حساب B اعمال زیر باید انجام شود:

۱. برداشت پول از حساب A

۲. واریز پول به حساب B

فرض کنید پس از برداشت پول از حساب A برنامه دچار خطا گردد. اکنون، پول از حساب A برداشت شده است و به حساب B واریز نگردید. این خرابی موجب ناسازگاری در داده می‌شود. بدیهی است که برای سازگاری پایگاه داده باید هم برداشت از حساب A و هم واریز به B انجام گردد یا هیچ کدام انجام نگردند. در نگهداری داده به صورت سنتی، تضمین یکپارچگی کار بسیار مشکلی است. اما، سیستم‌های مدیریت بانک اطلاعات از تراکنش پشتیبانی می‌کنند. یکی از خواص تراکنش یکپارچگی است. در ادامه با مفهوم تراکنش و خواص آن آشنا خواهید شد.

وجود مشکلات دستیابی همزمان، در برخی از محیط‌ها به ویژه اینترنت روزانه چندین میلیون خریدار به داده‌های فروشگاه‌های اینترنتی مراجعه می‌نمایند. در این مراجعات، ممکن است همزمانی برای دستیابی به اطلاعات رخ دهد. این همزمانی موجب ناسازگاری داده خواهد شد. به عنوان مثال، فرض کنید دو کاربر بخواهند به ترتیب مبلغ 5000000 و 2000000 ریال را از حساب A که دارای مانده 20000000 ریال است، برداشت کنند. در سیستم پردازش فایل سنتی، پس از برداشت‌ها از حساب A مانده حساب یکی از مقادیر 15000000 یا 18000000 ریال خواهد شد (بستگی دارد کدام یک ابتدا به کارش خاتمه دهد). در صورتی که پس از اجرای این برداشت‌ها مبلغ واقعی مانده حساب باید 13000000 ریال باشد. سیستم مدیریت بانک اطلاعات برای جلوگیری چنین ناسازگاری از پی‌درپی‌پذیری تراکنش یا قفل‌گذاری استفاده می‌کند.

وجود مشکلات تفکیک وجداسازی داده‌ها، چون داده‌های فایل‌ها به مرور زمان جمع می‌گردند و از طرف دیگر، برنامه نویسان مختلفی برنامه‌های کار با فایل‌ها را نوشته‌اند، ممکن است فایل‌های ایجاد شده دارای ساختار متفاوتی باشند. نوشتن برنامه‌های جدید که بتواند داده را از ساختارهای متفاوت تفکیک نماید، کار بسیار پیچیده‌ای است. در سیستم مدیریت بانک اطلاعات، داده‌ها با یک ساختار ذخیره می‌گردند. بنابراین، تفکیک داده‌ها آسان‌تر خواهد شد.

وجود مشکلات امنیتی، همه کاربران نباید به تمام داده دسترسی داشته باشند (هر کاربر باید به بخشی از داده دسترسی داشته باشد). به عنوان مثال، در یک سیستم دانشگاه، استاد باید فقط به اطلاعات دانشجویان خودش برای همان درس دسترسی داشته باشد، واحد مالی باید به اطلاعات مالی و آموزش باید به اطلاعات آموزشی دسترسی داشته باشد. ایجاد این دسترسی‌ها در سیستم‌های پردازش فایل سنتی دشوار است. در نرم‌افزار DBMS با تعریف سطح خارجی این دسترسی‌ها را به سادگی می‌توان تعریف و اعمال نمود. از طرف

دیگر، داده‌ها می‌توانند در مقابل دستیابی غیر قانونی و غیر مجاز حفظ شوند. زیرا، اطلاعات در یک نقطه نگهداری می‌گردند.

🔗 **فشرده سازی اطلاعات**، چون داده‌های بانک اطلاعاتی دارای ساختار هستند، نیازی به نگهداری فایل‌های متنی حجیم نیست و از ورود داده‌های بدون ساختار جلوگیری می‌کند. این امر موجب فشرده‌سازی اطلاعات می‌گردد.

🔗 **دسترسی سریع‌تر**، در هر زمان اطلاعات دقیق و به هنگام شده بازیابی می‌شوند. زیرا، اطلاعات به صورت مجتمع و یک‌پارچه نگهداری می‌گردند.

🔗 **نیاز به بودجه کمتر**، خیلی از یکنواختی‌ها در نگهداری فایل‌ها به روش دستی و سنتی که به فضای زیادی نیاز دارند، حذف خواهند شد و دیگر نیاز به ساختمان‌های بزرگ برای کارمندان جهت نگهداری و پردازش اطلاعات نمی‌باشد.

۲-۱. اجزای تشکیل دهنده بانک اطلاعاتی

هر بانک اطلاعاتی از چهار جزء تشکیل می‌شود که عبارت‌اند از:

۱. سخت‌افزار ۲. داده ۳. کاربران ۴. نرم‌افزار

۱-۲-۱. سخت‌افزار

سخت‌افزار مورد نیاز بانک اطلاعاتی را می‌توان به بخش‌های زیر تقسیم کرد:

۱. **سخت‌افزار پردازش‌گر مرکزی و حافظه اصلی**، این سخت‌افزارها برای اجرای نرم‌افزار بانک اطلاعاتی مورد استفاده قرار می‌گیرند.

۲. **محیط ذخیره‌سازی داده**، دیسک‌های جانبی از قبیل دیسک سخت، نوارها و غیره و دستگاه‌های I/O نظیر درایوها، کنترل‌گرها و ابزارهای دیگر برای ذخیره‌سازی و بازیابی داده‌ها به کار می‌روند.

۳. **سخت‌افزارهای ارتباطی**، دستگاه‌های نظیر هاب‌ها، سوئیچ‌ها، کابل‌ها، فیبرنوری و غیره هستند که برای انتقال داده به کار می‌روند.

بحث درباره اجزاء سخت‌افزاری از مقوله این کتاب خارج است. زیرا، حوزه‌های سخت‌افزاری و ارتباطی وسیع و گسترده هستند. برای هر یک از این حوزه‌ها نیاز به کتاب جداگانه‌ای می‌باشد.

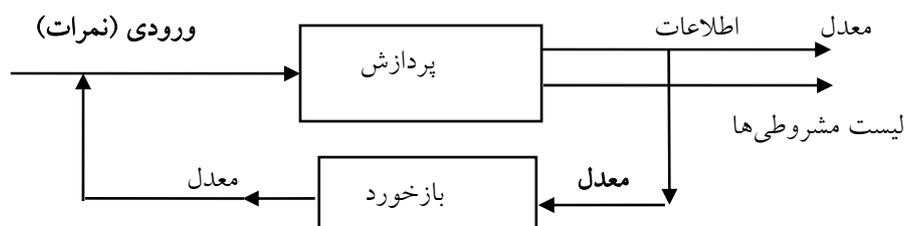
۲-۲-۱. داده‌ها

یکی از اجزا بسیار مهم بانک اطلاعاتی داده است. هر سیستم از سه بخش تشکیل می‌گردد (شکل ۱-۱). این سه بخش عبارت‌اند از:

۱. **داده**، هر چیزی که وارد سیستم می‌گردد، مانند شکل ۱-۱، نمرات و معدل ورودی سیستم هستند.

¹.Data

۲. اطلاعات^۱، هر چیزی که روی آن پردازش انجام شده و از سیستم خارج می‌شود. به عنوان مثال، در شکل ۱ - ۱ معدل و لیست مشروطی‌ها اطلاعات هستند.
۳. پردازش^۲، فرآیندی که داده را به اطلاعات تبدیل می‌کند، پردازش نام دارد.



شکل ۱ - ۱ بخش‌های یک سیستم.

همان‌طور، که می‌دانید سیستم‌های کامپیوتری می‌توانند تک کاربره^۳ یا چند کاربره^۴ باشند. در سیستم تک کاربره، در یک لحظه فقط یک کاربر می‌تواند از سیستم استفاده کند. اما، در سیستم‌های چند کاربره، در یک زمان چند کاربر می‌توانند از سیستم استفاده کنند. در سیستم مدیریت بانک اطلاعاتی از رویکرد چند کاربره استفاده می‌شود. لذا، در یک لحظه چند کاربر می‌توانند به داده‌ها دسترسی داشته باشند. بنابراین، داده باید مجتمع باشد و به اشتراک گذاشته شود. مجتمع کردن و به اشتراک گذاری داده مزایای زیر را دارد:

۱. بی‌نظمی و ناسازگاری را کاهش می‌دهد، چون داده‌ها در یک نقطه جمع‌آوری می‌شوند (پخش نمی‌شوند)، تغییر یک فیلد در یک مکان موجب تغییر این داده در کلیه بخش‌های بانک اطلاعات خواهد شد. بنابراین، موجب کاهش بی‌نظمی و ناسازگاری داده خواهد شد.
۲. افزونگی داده را کاهش می‌دهد، از آنجائی که داده‌ها به اشتراک گذاشته می‌شوند، نیازی به تکرار بی‌مورد داده‌ها در جداول مختلف نمی‌باشد. بنابراین، افزونگی نیز کاهش می‌یابد.

اما، وقتی که داده‌ها را به اشتراک می‌گذاریم باید مسائل و مشکلات زیر را حل کنیم:

۱. امنیت، وقتی داده‌ها به اشتراک گذاشته می‌شوند باید تعیین گردد چه کاربری با چه مجوزهایی به کدام یک از داده‌ها دسترسی داشته باشد. برای این منظور می‌توان سطوح دستیابی را برای کاربران تعریف نمود.

۲. کنترل همروندی (همزمانی)، همان‌طور که بیان گردید، همزمانی دسترسی به داده‌ها ممکن است ناسازگاری در داده‌ها ایجاد کند. به عنوان مثال، دو کاربر همزمان بخواهند مانده حساب کاربری را تغییر دهند، مشخص نیست کدام یک از تغییرات باید اعمال گردد. DBMS برای رفع این مشکل تراکنش‌ها را پی‌درپی انجام می‌دهد (تا یکی تمام نگردید، دیگری را شروع نمی‌کند) یا از قفل‌گذاری استفاده می‌کند. یعنی، داده‌هایی که به آن‌ها می‌خواهد دستیابی داشته باشد، قفل کرده تا زمانی که کار تراکنش با آن داده

¹.Information

².Process

³.Single User

⁴.Multi User

خاتمه نیافت، قفل را از دست نمی‌دهد. بنابراین، با این دو روش می‌توان همزمانی و همروندی دستیابی به داده‌ها را کنترل کرد.

۳-۲-۱. کاربران

کاربران^۱، افرادی هستند که از بانک اطلاعاتی استفاده می‌کنند. کاربران بانک اطلاعاتی سه دسته هستند که عبارت‌اند از:

۱. کاربران برنامه‌نویس کاربردی

این کاربران، با یکی از زبان‌های برنامه‌نویسی از قبیل ++C، ویژوال بیسیک، C# و غیره برنامه‌های کاربردی بانک اطلاعاتی را می‌نویسند. این کاربران فرم‌های ورود، ویرایش، حذف و جستجوی داده را برای کاربران نهایی^۲ فراهم می‌کنند.

بعضی از کاربران برنامه‌نویس کاربردی از ابزارهای توسعه سریع کاربرد^۳ استفاده می‌کنند. از طریق این ابزارها، برنامه‌نویس می‌تواند بدون نوشتن برنامه زیاد فرم‌ها و گزارش‌های کاربران نهایی را تولید نماید.

۲. کاربران نهایی

کاربران غیرحرفه‌ای هستند که از طریق برنامه‌های نوشته شده توسط برنامه‌نویسان کاربردی، از سیستم استفاده می‌کنند و داده‌های بانک اطلاعاتی را دستکاری می‌نمایند. به عنوان مثال، تحویل‌دار بانکی می‌خواهد مبلغی پول را از حساب A به حساب B انتقال دهد. برای این منظور، از یک فرم از قبل طراحی شده استفاده می‌کند. کاربران نهایی ممکن است گزارش‌های ایجاد شده توسط بانک اطلاعات را بخوانند. کاربران نهایی علاوه بر ورود اطلاعات از طریق فرم‌های ورود و تهیه گزارش می‌توانند دستورات DELETE، UPDATE، INSERT و SELECT را اجرا کنند. با این دستورات در فصل ۵ آشنا خواهید شد.

۳. کاربر مدیر بانک اطلاعاتی

مدیر بانک اطلاعاتی (DBA)^۴، مسئولیت‌هایی از قبیل کنترل متمرکز، طراحی، تصمیم‌گیری و غیره را بر روی بانک اطلاعات بر عهده دارد. مدیر بانک اطلاعاتی، در بانک‌های اطلاعاتی بزرگ ممکن است این مسئولیت را به شخص یا تیم دیگری واگذار کند. در ادامه وظایف مدیر بانک اطلاعات را می‌بینید.

۴-۲-۱. نرم‌افزار

نرم‌افزار بانک اطلاعاتی همان سیستم مدیریت بانک اطلاعاتی (DBMS) است. این نرم‌افزار واسط کاربر، برنامه‌های کاربردی و بانک اطلاعاتی می‌باشد. از جمله وظایف DBMS دریافت، ترجمه، اجرا، بهینه‌سازی دستورات و غیره می‌باشد. در ادامه وظایف بیشتر DBMS را می‌بینید.

¹.Users ².End users ³.Rapid Application Development ⁴.Database Administrator

۳-۱. انواع بانک‌های اطلاعاتی

برای سازماندهی داده‌ها و روابط بین آن‌ها روش‌های مختلفی وجود دارد. برخی از این روش‌ها عبارت‌اند از:

۱. بانک‌های اطلاعاتی هرمی (سلسله مراتبی)
۲. بانک‌های اطلاعاتی شبکه‌ای
۳. بانک‌های اطلاعاتی رابطه‌ای
۴. بانک‌های اطلاعاتی شی‌گرا (فرا رابطه‌ای).

۱-۳-۱. بانک‌های اطلاعاتی هرمی (سلسله مراتبی)

در بانک‌های اطلاعاتی هرمی، اطلاعات به صورت درختی در اختیار کاربران قرار می‌گیرد. داده‌ها در بخش‌هایی به نام **قطعه^۱** ذخیره می‌شوند. قطعه بالای سلسله، **ریشه^۲** نام دارد. ریشه به صورت رابطه پدر-فرزندی به قطعات زیرین وصل می‌شود. قطعه پدر می‌تواند چندین فرزند داشته باشد، ولی هر فرزند فقط یک پدر دارد. به عنوان مثال، چگونگی رزرو صندلی در شرکت هواپیمایی در مدل هرمی در شکل ۲-۱ آمده است.

برخی از ویژگی‌های این مدل را در زیر می‌بینید:

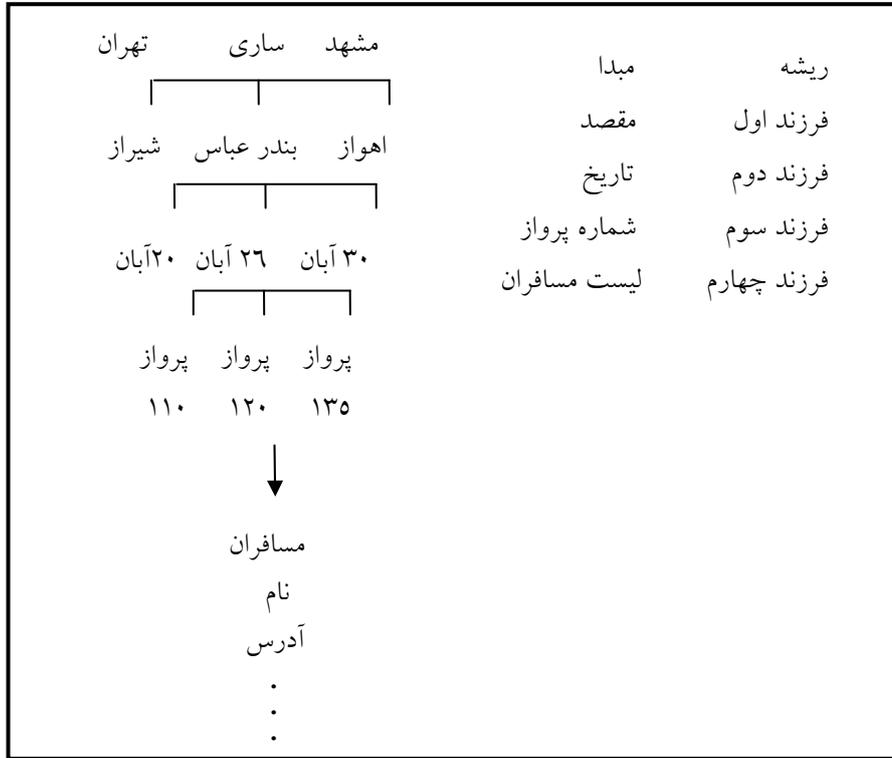
۱. در این مدل با حذف گره پدر، کلیه فرزندان آن نیز حذف خواهند شد.
۲. عملگرهای بازیابی اطلاعات در هرمی به سادگی عملگرهای مدل رابطه‌ای نیستند. برای بازیابی اطلاعات در روش هرمی، از پیمایش اشاره‌گرها استفاده می‌شود.
۳. پیمایش و بازیابی گره‌های سطح پایین‌تر لزوماً باید از نقطه ورود (ریشه)، در سطح بالاتر شروع شود.

۲-۳-۱. بانک اطلاعاتی شبکه‌ای

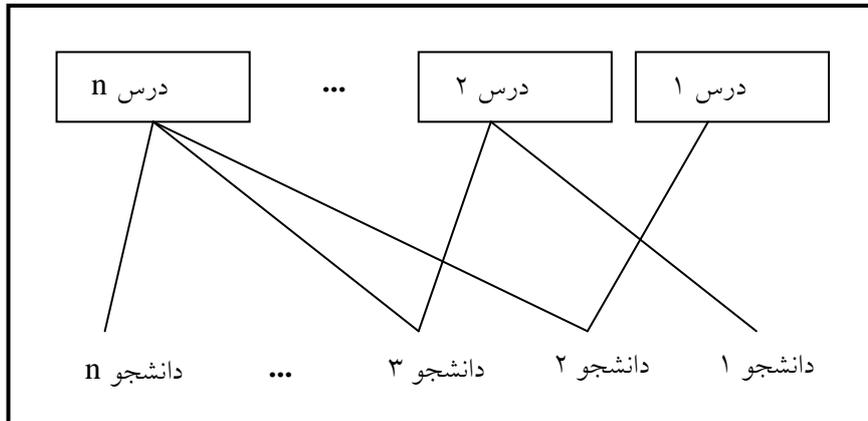
بانک اطلاعاتی شبکه‌ای، حالت تغییر یافته مدل بانک اطلاعاتی هرمی است. این دو مدل به سادگی به یکدیگر قابل تبدیل هستند. در مدل هرمی می‌توان یک عامل را به چند عامل پیوند داد، ولی در مدل شبکه‌ای می‌توان چندین عامل را به چندین عامل پیوند داد. شکل ۳-۱ مدل شبکه‌ای ثبت نام دانشجویان را نشان می‌دهد. پیاده‌سازی این مدل از طریق گراف انجام می‌شود. در این جا می‌توان نام هر دانشجو را در ترم و درسی که انتخاب می‌کند، جداگانه ثبت کرد. ولی، این کار موجب تکرار داده‌ها شده و از سرعت بازیابی اطلاعات می‌کاهد.

^۱.Segment

^۲. Root



شکل ۱-۲ مدل هرمی (سلسله مراتبی) بانک اطلاعاتی.



شکل ۱-۳ انتخاب واحد دانشجویان در مدل شبکه‌ای بانک اطلاعاتی.

۳-۳-۱. بانک اطلاعاتی رابطه‌ای

در سیستم بانک اطلاعاتی رابطه‌ای، هر یک از فایل‌های بانک اطلاعاتی را یک جدول می‌نامند. به عبارت دیگر، بانک اطلاعاتی مجموعه‌ای از جدول‌ها است و هر جدول حاوی داده‌های مرتبط به هم است. به عنوان

آشنایی با مفاهیم اولیه بانک اطلاعاتی ۱۷

مثال، شکل ۴ - ۱ نمونه‌ای از جدول را نشان می‌دهد که حاوی اطلاعات مربوط به دانشگاه است. اگر نگاهی به شکل ۴ - ۱ داشته باشید، می‌بینید که جدول موجود در این شکل دارای ۳ ستون و ۳ سطر است. هر ستون را یک فیلد^۱ و هر سطر را یک رکورد^۲ گویند. به عبارت دیگر، به هر یک از اطلاعات کامل هر دانشگاه، یک رکورد گویند. بنابراین، هر رکورد بانک اطلاعاتی حاوی چند فیلد است که هر فیلد از نوع خاصی است. به عنوان مثال، فیلد نام دانشگاه از نوع رشته‌ای، فیلد کد دانشگاه از نوع عددی و فیلد کد رئیس دانشگاه نیز عددی است. هر یک از جدول‌های موجود در بانک اطلاعاتی می‌توانند با یکدیگر ارتباط داشته باشند. نکته دیگری که در مورد جدول‌ها قابل توجه است، فیلد کلید اولیه^۳ است. این فیلد، مشخصه هر رکورد است و در هیچ دو رکوردی از جدول، یکسان نیست. به عنوان مثال، اگر فیلد شماره دانشگاه را در شکل ۴ - ۱ به عنوان فیلد کلید اولیه در نظر بگیریم، برای هر دانشگاه منحصر به فرد است (این فیلد مقدار تکراری نخواهد داشت).

برخی از ویژگی‌ها بانک اطلاعاتی در زیر آمده است:

۱. داده‌ها و ارتباط بین آن‌ها با سطرها نشان داده می‌شوند.
۲. عملگرهای بازیابی اطلاعات نسبتاً ساده هستند.
۳. از مبنای تئوریک ریاضی برخوردار است.
۴. پرس‌وجو از یک جدول می‌تواند مستقل از جدول دیگر باشد.

شماره دانشگاه	نام دانشگاه	کد رئیس دانشگاه
۱	جهاد دانشگاهی مازندران	۱
۲	علوم و فنون مازندران	۳
۳	دانشگاه مازندران	۸

شکل ۴ - ۱ نمونه‌ای از جدول (رابطه) در مدل رابطه‌ای.

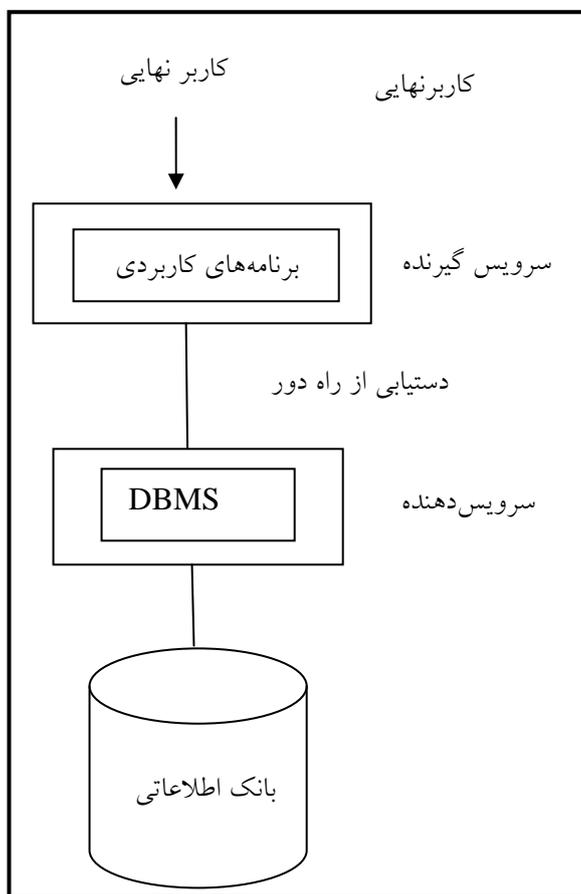
۴ - ۳ - ۱. مدل بانک اطلاعاتی شی‌گرا

امروزه برخی از کاربردهای بانک اطلاعاتی نظیر طراحی به کمک رایانه و سیستم‌های بانک اطلاعاتی جغرافیایی را نمی‌توان از طریق مدل بانک اطلاعاتی رابطه‌ای ذخیره نمود یا ذخیره آن‌ها مشکلاتی را ایجاد خواهد کرد. برای رفع این مشکلات از مدل بانک اطلاعاتی شی‌گرا استفاده می‌شود. از طرف دیگر، برخی از زبان‌های برنامه‌نویسی مدل شی‌گرایی را پشتیبانی می‌کنند. بنابراین، اشیا را باید بتوان در بانک اطلاعاتی ذخیره نمود. هر شیء از خواص^۴ (ویژگی‌های شیء)، رویدادها^۵ و متدها^۶ (اعمالی که بر روی شیء می‌تواند انجام شود) تشکیل می‌شود. بانک اطلاعاتی شی‌گرا، می‌تواند اشیا با تمام ویژگی‌ها، رویدادها و متدهای آن را نگهداری کند. در حالی که، در بانک اطلاعاتی رابطه‌ای امکان ذخیره‌سازی اشیا وجود ندارد.

¹.Field ².Record ³.Primary Key Field ⁴.Properties ⁵.Events
⁶.Methods

۴-۱. انواع مختلف بانک‌های اطلاعاتی از لحاظ جغرافیایی

هر بانک اطلاعاتی از دو بخش سرویس‌دهنده^۱ و سرویس‌گیرنده^۲ تشکیل می‌شود (شکل ۵-۱). سرویس‌دهنده، همان DBMS است که دستکاری داده، تعریف داده، مدیریت تراکنش‌ها، اجرای قوانین جامعیت و امنیت داده، جلوگیری از ناسازگاری داده، نگهداری دیکشنری داده و غیره از وظایف آن است. ولی، سرویس‌گیرنده، برنامه‌های کاربردی مختلفی هستند که در بالای سرویس‌دهنده (DBMS) قرار دارند. این برنامه‌ها، دستورات کاربران نهایی را اجرا می‌کنند. برنامه‌های کاربردی می‌توانند واسط‌های منویی و فرمی داشته باشند. بانک‌های اطلاعاتی از لحاظ جغرافیایی به دو دسته متمرکز و نامتمرکز (توزیع شده) تقسیم می‌گردند. در بانک اطلاعاتی متمرکز، یک سرویس‌دهنده وجود دارد. تمام سرویس‌گیرنده‌ها به این سرویس‌دهنده متصل هستند. این معماری را در شکل ۵-۱ می‌بینید.



شکل ۵-۱ بانک اطلاعاتی با معماری

سرویس‌گیرنده / سرویس‌دهنده.

این سرویس‌گیرنده‌ها پرس‌وجوی خودشان را به سرویس‌دهنده می‌فرستند تا توسط سرویس‌دهنده پردازش و نتایج را از سرویس‌دهنده می‌گیرند. ولی، در بانک اطلاعاتی توزیع شده چند سرویس‌دهنده در نقاط مختلف وجود دارند. سرویس‌گیرنده‌ها می‌توانند به یک یا چند سرویس‌دهنده دسترسی داشته باشند. شکل ۶-۱ این دستیابی را نشان می‌دهد که به دو روش زیر می‌باشند:

۱. هر سرویس‌گیرنده می‌تواند به یک یا چند سرویس‌دهنده دسترسی داشته باشد. ولی، در هر زمان فقط به یکی از آنها دسترسی دارد (یعنی، پرس‌وجوها را باید فقط به یک سرویس‌دهنده ارسال کند). به عبارت دیگر، یک پرس‌وجو نمی‌تواند توسط چند سرویس‌دهنده پاسخ داده شود

¹.Server ².Client

آشنایی با مفاهیم اولیه بانک اطلاعاتی ۱۹

و با هم ترکیب گردند. چون، ممکن است سرویس دهنده‌های مختلف داده‌های متفاوتی را نگهداری کنند. پس، سرویس گیرنده (کاربر) باید بداند هر سرویس دهنده چه داده‌ای را ذخیره کرده است.

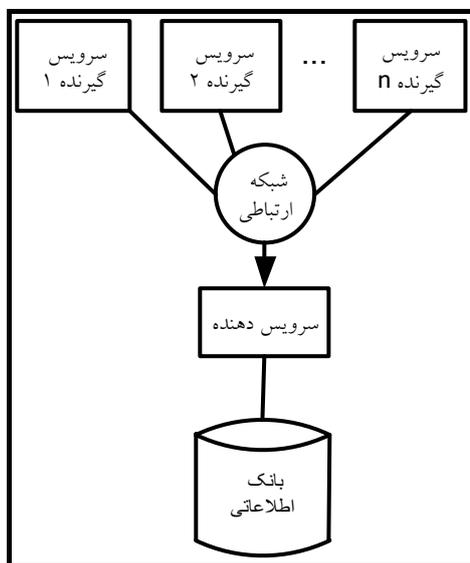
۲. سرویس گیرنده ممکن است بتواند همزمان به چند سرویس دهنده دسترسی داشته باشد (یک پرس و جوی بانک اطلاعاتی می‌تواند داده‌هایی را از چند سرویس دهنده ترکیب کند). در این صورت، نیازی نیست سرویس گیرنده بداند چه داده‌ایی در چه سرویس دهنده‌ای ذخیره شده است. به عنوان نمونه‌ای از بانک‌های اطلاعاتی متمرکز و توزیع شده می‌توان بانک اطلاعات مشترکین کل کشور بر روی یک سرویس دهنده نگهداری می‌شود. صورت متمرکز نگهداری شود، اطلاعات مشترکین کل کشور بر روی یک سرویس دهنده نگهداری می‌شود. اما، اگر این اطلاعات به صورت توزیع شده نگهداری شوند (هم اکنون به صورت توزیع شده نگهداری می‌گردد)، اطلاعات هر استان بر روی یک سرویس دهنده قرار می‌گیرد و کاربران هر استان فقط به اطلاعات سرویس دهنده خودشان دسترسی دارند. این دو نوع ذخیره‌سازی را می‌توان از جنبه‌های زیر با یکدیگر مقایسه کرد:

۱. سرعت پاسخ‌گویی به پرس و جوها

۲. برقراری امنیت

۱. سرعت پاسخ‌گویی به پرس و جوها

سرعت پاسخ‌گویی به پرس و جوها در بانک‌های اطلاعاتی متمرکز و توزیع شده به نوع پرس و جو بستگی دارد. در بانک اطلاعاتی دو نوع پرس و جو وجود دارد که عبارت‌اند از:



شکل ۶- ۱ یک سرویس دهنده و چند سرویس گیرنده (متمرکز).

✚ **پرس وجوهای محلی^۱**، پرس وجوهایی که مربوط به منطقه خاص باشند، پرس وجوی محلی نام دارند. نمونه‌ای از این پرس وجو می‌توان، بازیابی اطلاعات مشترکین مازندارنی که نام آن‌ها "علی" است را نام برد. سرعت پاسخ‌گویی به چنین پرس وجوها در بانک اطلاعات توزیع شده نسبت به متمرکز بیشتر است. زیرا، چنانچه کاربر به سرویس دهنده مازندارن متصل گردد، سریع‌تر می‌تواند این اطلاعات را بازیابی کند.

✚ **پرس وجوهای سراسری^۲**، پرس وجوهایی که مربوط به چندین منطقه باشند، پرس وجوهای سراسری نام دارند. به عنوان نمونه‌ای از این پرس وجو می‌توان بازیابی اطلاعات مشترکین ایرانی با نام "علی" را نام برد. برای پاسخ‌گویی به چنین پرس وجوهای بهتر است اطلاعات به صورت متمرکز نگهداری شوند. زیرا، چنانچه اطلاعات به صورت نامتمرکز (توزیع شده) نگهداری شوند، این پرس وجو باید برای سرویس دهنده‌های مختلف ارسال گردد و نتیجه هر یک از این سرویس دهنده‌ها دریافت و با یکدیگر ترکیب شوند تا نتیجه چنین پرس وجویی ایجاد گردد. در حالی که، در بانک اطلاعات متمرکز، این پرس وجو فقط به یک سرویس دهنده ارسال می‌گردد و نیازی به ترکیب نتیجه پرس وجوها ندارد.

۲. برقراری امنیت

امنیت بانک‌های اطلاعاتی توزیع شده و متمرکز را از دو جنبه می‌توان بررسی کرد که عبارت‌اند از:

✚ **در هنگام تخریب اطلاعات**، همان‌طور که می‌دانید، در بانک‌های اطلاعاتی متمرکز کلیه اطلاعات بر روی یک سرویس دهنده قرار دارند (کلیه اطلاعات مشترکین مخابرات ایران بر روی یک سرویس دهنده قرار داشته باشد). بنابراین، با از بین رفتن اطلاعات این سرویس دهنده، کلیه اطلاعات از دست خواهند رفت و هیچ کاربری نمی‌تواند به سیستم دسترسی داشته باشد. ولی، در سیستم توزیع شده، با از دست رفتن اطلاعات یک سرویس دهنده، فقط اطلاعات همان سرویس دهنده از بین خواهد رفت و کاربران همین سرویس دهنده نمی‌توانند به اطلاعات خودشان دسترسی داشته باشند. اما، بقیه کاربران می‌توانند به کارشان ادامه دهند.

✚ **در هنگام دسترسی غیرمجاز**، از آنجائی که در بانک اطلاعاتی متمرکز، اطلاعات بر روی یک سرویس دهنده قرار دارند، می‌توان این سرویس دهنده را به فایروال قوی مجهز نمود تا از ورود کاربران غیر مجاز جلوگیری کرد. اما، در بانک اطلاعاتی توزیع شده، چون اطلاعات بر روی سرویس دهنده‌های مختلفی قرار می‌گیرند، از لحاظ امنیت دسترسی غیرمجاز مشکل‌تر خواهد شد.

۵-۱. سطوح مختلف بانک اطلاعاتی

همان‌طور که دیدید، کاربران بانک اطلاعاتی سه نوع هستند که عبارت‌اند از:

۱. کاربران نهایی
۲. کاربران برنامه‌نویسان کاربردی
۳. کاربران مدیر.

اکثر کاربران بانک اطلاعاتی، کاربران نهایی هستند. این کاربران اطلاعات زیادی درباره کامپیوتر ندارند. برای این که این کاربران به سادگی و بدون نیاز به آموزش زیاد بتوانند از بانک اطلاعاتی استفاده کنند، سه سطح^۳

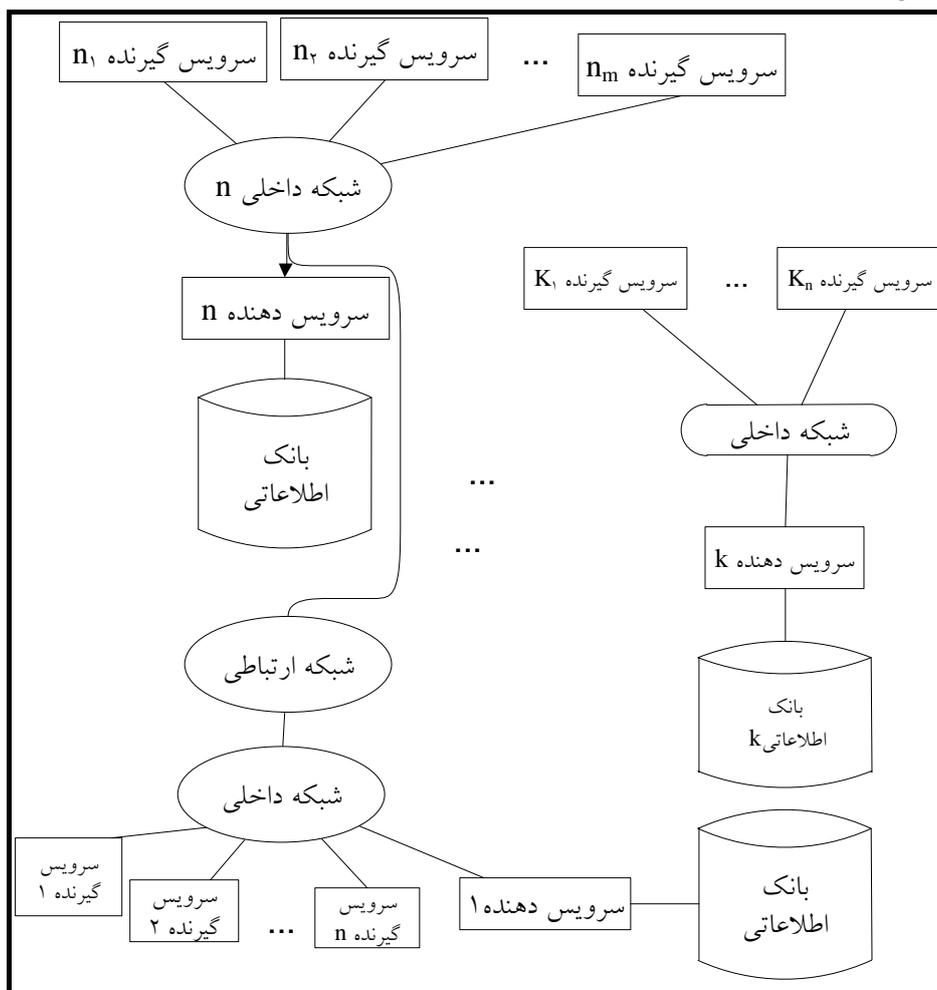
^۱.Local Queries

^۲.Global Queries

^۳.براساس مدل ANSI سه سطح خارجی، ادراکی و فیزیکی برای بانک اطلاعات پیشنهاد شده است.

آشنایی با مفاهیم اولیه بانک اطلاعاتی ۲۱

برای بانک اطلاعاتی در نظر گرفته شده است (در شکل ۸-۱ این سطوح را می بینید). این عمل موجب می شود تا پیچیدگی در سطوح بانک اطلاعاتی گنجانده شده و از دید کاربران مخفی گردد. سطوح مختلف بانک اطلاعاتی عبارتند از:



شکل ۷-۱ بانک اطلاعاتی توزیع شده با چند سرویس دهنده.

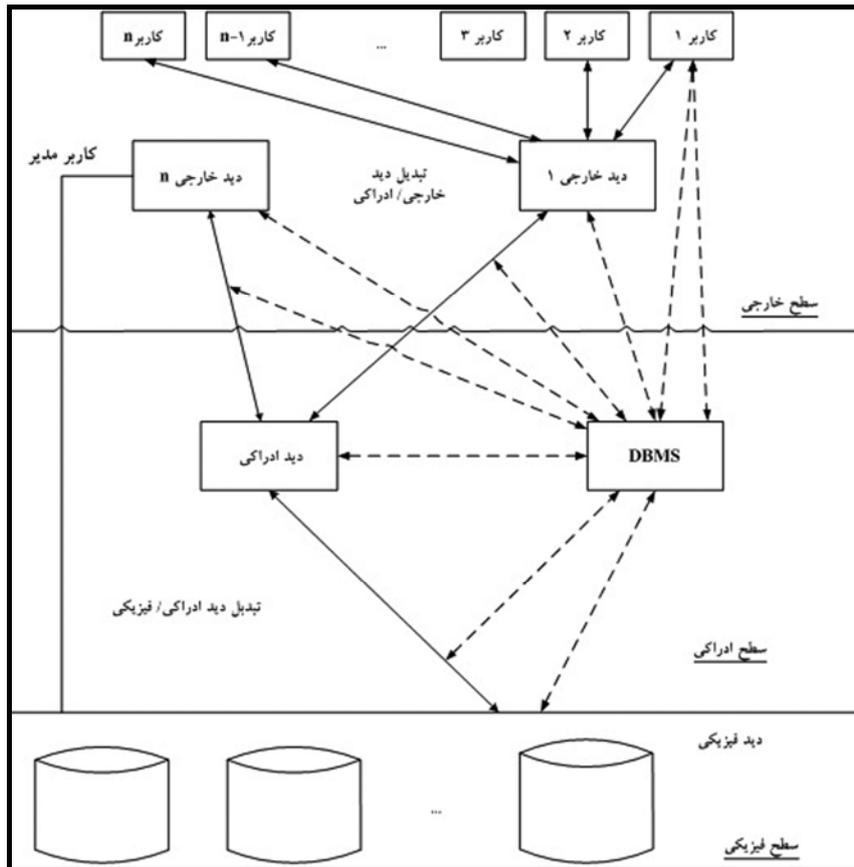
۱. سطح خارجی، دید هر یک از کاربران (به چه داده‌ای و با چه زبانی دسترسی داشته باشند)، نسبت به بانک اطلاعاتی را سطح خارجی گویند.
۲. سطح ادراکی، اجتماع دیدهای کاربران سطح خارجی، سطح ادراکی نام دارد.
۳. سطح فیزیکی، چگونگی ذخیره و بازیابی داده‌های بانک اطلاعاتی سطح فیزیکی نام دارد.

۱-۵-۱. سطح خارجی

- برای این که با سطوح مختلف بانک اطلاعاتی آشنا شوید، به مثال زیر توجه کنید:
۱. کاربر ۱، با زبان ++C به فیلدهای شماره دانشجویی (stNo) و نمره (score) دانشجو دسترسی دارد.

۲۲ فصل اول

۲. کاربر ۲، با زبان پاسکال به فیلدهای نام (Fname) و نام خانوادگی (Lname) دانشجو دسترسی دارد.
۳. کاربر ۳، با زبان ویژوال بیسیک به فیلدهای نام (Fname)، نام خانوادگی (Lname) و نمره (score) دانشجو دسترسی دارد.



شکل ۸ - ۱ سطوح مختلف بانک اطلاعات.

کاربر ۱، برای دسترسی به بانک اطلاعات دستورات زیر را تعریف می‌کند:

```
struct Student
{
    char stNo [9];
    float score;
};
```

کاربر ۲، دستورات زیر را می‌نویسد تا به اطلاعات بانک اطلاعات خودش دستیابی داشته باشد:

```
Student = Record
    Fname : String [15];
    Lname : String [20];
End;
```

اما، کاربر ۳، برای این که به داده‌های خودش دسترسی یابد، دستورات زیر را وارد می‌کند:

```
Struct Student
```

```
Dim Fname As String
Dim Lname As String
Dim Score As Single
End Struct
```

همان گونه که در این تعاریف دیده‌اید، هر یک از کاربران به بخشی از بانک اطلاعات با زبان خاصی دسترسی دارند. بنابراین، نیاز به آموزش زبان جدیدی برای استفاده از بانک اطلاعات ندارند.

۲-۵-۱. سطح ادراکی

سطح مفهومی، اجتماع دید کاربران سطح خارجی است. در سطح مفهومی به داده‌ها به همان شکلی که هستند، نگاه می‌شود، نه به شکلی که کاربران می‌بینند. به عنوان مثال، سطح مفهومی بانک اطلاعات بیان شده به شکل زیر است:

```
Student
Student_Number Character (8),
Fname           Character (5),
Lname           Character (20),
score           Decimal (5, 2),
```

سطح ادراکی کاملاً رابطه‌ای است و اشیایی که در این سطح قابل مشاهده هستند، جداول رابطه‌ای می‌باشند و عملگرهای این سطح نیز عملگرهای رابطه‌ای پرتو^۱، انتخاب^۲ و غیره هستند. در فصل ۳ این عملگرها را می‌آموزیم.

کاربر این سطح، مدیر بانک اطلاعاتی است. در این سطح مسائلی مانند امنیت و جامعیت مطرح است.

۳-۵-۱. سطح داخلی

سطح داخلی به چگونگی ذخیره‌سازی و بازیابی رکوردها می‌پردازد. یعنی، در این سطح دقیقاً تعیین می‌شود، رکوردها به چه ترتیبی ذخیره می‌گردند و ترتیب ذخیره‌سازی فیلدها چگونه است. در این سطح دقیقاً تعیین می‌شود هر فیلد چند بایت از حافظه را اشغال می‌کند و از چه مکانی از حافظه شروع می‌گردد. به عنوان مثال، بانک اطلاعاتی دانشجویان در سطح فیزیکی به شکل زیر بیان می‌گردد:

```
STORED-Student BYTES = 53
PERFIX BYTES = 6, OFFSET = 0
stNo   BYTES = 8, OFFSET = 6, INDEX = ON
Fname  BYTES = 15, OFFSET = 14
Lname  BYTES = 20, OFFSET = 29
Score  BYTES = 4, OFFSET = 49
```

همان‌طور که در این تعریف می‌بینید، شروع هر فیلد دقیقاً مشخص شده است. یعنی، با گزینه OFFSET. شروع فیلد تعیین شده است. OFFSET، فاصله فیلد تا ابتدای رکورد را تعیین می‌کند. اندازه هر فیلد مشخص شده است. به عنوان مثال، فیلد score، ۴ بایت از حافظه را اشغال می‌کند و ترتیب قرار گرفتن رکوردها نیز در این تعریف تعیین شده است (رکوردها به ترتیب شماره دانشجویی مرتب هستند. زیرا، INDEX (مرتب‌سازی) مربوط به فیلد شماره دانشجویی فعال (ON) شده است.

¹.Projection ².Selection

۶-۱. زبان میزبان

برخی از اعمال از قبیل تعریف متغیرهای محلی، اعمال محاسباتی، منطقی، تصمیم‌گیری و غیره که در بانک اطلاعات وجود ندارند، توسط **زبان میزبان**^۱ انجام می‌شوند. زبان میزبان می‌تواند یکی از زبان‌های سطح بالا مانند C، C#، دلفی، ویژوال بیسیک و غیره باشد. زیرا، زبان داده‌ای خاص مانند SQL می‌تواند مهمان آن‌ها شود. زبان فرعی داده (DSL)^۲ همان زبانی است که مهمان زبان میزبان است. زبان فرعی داده (DSL) سه نوع است که عبارت‌اند از:

۱-۶-۱. زبان تعریف داده

زبان تعریف داده (DDL)^۳، برای تعریف ساختار داده‌ای از قبیل ساختار فیلدها، و چگونگی ذخیره‌سازی آن‌ها به کار می‌رود. این زبان برای تعریف و دستکاری ساختار جداول و اشیا دیگر به کار می‌رود. اطلاعات جداول و ساختار آن‌ها در فایل فرهنگ داده^۴ ذخیره می‌گردد. از جمله دستورات این زبان می‌توان دستورات ALTER، DROP و CREATE در زبان SQL را نام برد.

۲-۶-۱. زبان دستکاری داده

زبان دستکاری داده (DML)^۵، برای دستکاری داده‌های بانک اطلاعاتی به کار می‌رود. یعنی، این زبان می‌تواند اعمال حذف داده موجود، ویرایش داده موجود، اضافه نمودن داده جدید و بازیابی داده‌های موجود را انجام دهد. دستورات زبان DML دو نوع‌اند که عبارت‌اند از:

🔲 DML رویه‌ای^۶، زبانی است که در آن کاربر تعیین می‌کند به چه داده‌هایی نیاز است و چگونه می‌تواند آن‌ها را بیان و دستکاری نماید.

🔲 DML غیر رویه‌ای^۷، زبانی است که کاربر بدون مشخص کردن چگونگی دستیابی به داده، تعیین می‌کند به چه داده‌هایی نیاز می‌باشد. پرس‌وجو^۸ که برای بازیابی داده به کار می‌رود، بخشی از DML است. این بخش **زبان پرس‌وجو** نام دارد.

دستوراتی مانند INSERT (افزودن رکورد جدید)، UPDATE (ویرایش رکوردها)، DELETE (حذف رکوردها) و SELECT (بازیابی رکوردها)، دستورات DML زبان SQL هستند.

۳-۶-۱. زبان کنترل داده

زبان کنترل داده (DCL)^۹، برای حفظ یکپارچگی، صحت، امنیت داده‌ها، ارتباط بین داده‌ها و جداول به کار می‌رود. نمونه‌ای از این دستورات می‌توان تعریف محدودیت‌ها و مجوزها (GRANT و REVOKE) در زبان SQL را نام برد.

^۱.Host Language

^۲. Data Sub Language

^۳.Data Definition Language

^۴.Data Dictionary

^۵.Data Manipulation Language

^۶.Procedural DML

^۷.Non Procedural DML

^۸.Query

^۹.Data Control Language

۷-۱. جامعیت

جامعیت^۱، اطمینان از این که داده‌های موجود در بانک اطلاعاتی صحیح هستند. یعنی، جامعیت از ورود داده‌های نادرست و انجام پردازش‌های غلط جلوگیری می‌کند.

برای این منظور مدیر داده قیود^۲ جامعیت را تعریف می‌نماید و مدیر بانک اطلاعات آن را پیاده‌سازی می‌کند. برخی از این قیودها عبارت‌اند از:

۱. موجودی حساب بانکی نمی‌تواند منفی شود. بنابراین، شخص نمی‌تواند بیش از موجودی حساب خودش برداشت نماید.
۲. نمره دانشجو نمی‌تواند خارج از بازه ۰ تا ۲۰ باشد. بنابراین، امکان ورود نمرات خارج از بازه ۰ تا ۲۰ توسط کاربر نمی‌باشد.
۳. ساعات کار کرد هفتگی یک کارمند نمی‌تواند بیش از ۷۲ ساعت باشد، پس در هنگام ورود اطلاعات کار کرد هفتگی باید جلوی کاربر را از ورود عددی بیش از ۷۲ ساعت بگیرد.

۸-۱. امنیت

امنیت^۳، به معنی محافظت داده است. محافظت داده را از دو بعد می‌توان بررسی کرد که عبارت‌اند از:
🚩 **محافظت داده در مقابل تخریب**، خطراتی از قبیل آتش‌سوزی، سیل، زلزله و خرابی هارد دیسک موجب تخریب اطلاعات می‌شوند. برای جلوگیری از تخریب اطلاعات از سیاست‌های مختلف **پشتیبان‌گیری**^۴ استفاده می‌کنیم.

🚩 **محافظت داده از دستیابی غیر مجاز**، همان‌طور که بیان گردیده، داده‌های بانک اطلاعاتی به اشتراک گذاشته می‌شوند. بنابراین، باید از دسترسی افراد غیرمجاز به داده‌ها جلوگیری کرد. برای این منظور می‌توان از کنترل دستیابی استفاده نمود. متداول‌ترین روش کنترل دستیابی استفاده از کلمه عبور^۵ برای کاربران است. هیچ‌گاه نمی‌توان صددرد امنیت را تضمین کرد (به خصوص بانک اطلاعات توزیع شده باشد).

۹-۱. وظایف سیستم مدیریت بانک اطلاعاتی

سیستم مدیریت بانک اطلاعاتی (DBMS)، بین بانک اطلاعاتی فیزیکی و کاربران سیستم قرار دارد و اعمالی از قبیل اضافه، حذف جداول، اضافه کردن رکورد، ویرایش رکوردها، حذف رکوردها و بازیابی رکوردها را به عهده دارد. سیستم مدیریت بانک اطلاعاتی، نرم‌افزاری است که تمام دستیابی‌ها به بانک اطلاعاتی را برعهده دارد. عملیاتی که در DBMS انجام می‌شود، در شکل ۹-۱ آمده است. هر DBMS وظایف زیر را به عهده دارد:

۱. تعریف داده‌ها

DBMS باید برای هر یک از زبان‌های تعریف داده پردازش‌گر DDL (اجزای کامپایلر DDL) داشته باشد. DBMS همچنین باید بتواند تعاریف DDL را درک کند. به عنوان مثال، می‌داند که رکوردهای خارجی

^۱.Integrity

^۲.Constraints

^۳.Security

^۴.Backup

^۵.Password