



پایگاهداده‌ها

(رشته‌های مهندسی کامپیوتر، فناوری اطلاعات و ریاضی)

دکتر احمد فراهی

دکتر مصطفی حق جو

سرشناسه	: حق جو، مصطفی، ۱۳۳۴ -
عنوان و نام پدیدآور	: پایگاه داده‌ها / مصطفی حق جو، احمد فراهی.
مشخصات نشر	: تهران: دانشگاه پیام نور، ۱۳۹۷.
مشخصات ظاهری	: دوازده، ۲۵۲ ص.: مصور، جدول، نمودار.
فروخت	: دانشگاه پیام نور؛ ۱۴۲۴. گروه مهندسی فناوری اطلاعات؛ آ/۲۳.
شابک	: ۹۷۸-۹۶۴-۳۸۷-۴۴۴-۵
وضعیت فهرست نویسی	: فیپا
یادداشت	: کتابنامه: ص. [۲۵۱] - ۲۵۲.
یادداشت	: واژه‌نامه.
موضوع	: پایگاه‌های اطلاعاتی -- آموزش برنامه‌ای
موضوع	: Databases -- Programmed instruction:
شناسه افزوده	: فراهی، احمد، ۱۳۳۸ -
شناسه افزوده	: دانشگاه پیام نور.
رده بندی کنگره	: QA۷۶/۹۲۷ ۱۳۹۷
رده بندی دیوی	: ۰۰۵/۰۷۱۱
شماره کتابشناسی ملی	: ۰۱۷۰۷۱۱



پایگاه داده‌ها

دکتر مصطفی حق جو دکتر احمد فراهی

ویراستار علمی: دکتر اکبر فرهودی نژاد

تهیه و تولید: دفتر تدوین و تولید کتب و محتوای آموزشی

لیتوگرافی، چاپ و صحافی: مرکز چاپ و توزیع دانشگاه پیام نور

شمارگان: نسخه

نوبت و تاریخ چاپ: اول، ۱۳۹۷

شابک: ۹۷۸-۹۶۴-۳۸۷-۴۴۴-۵

ISBN: 978-964-387-444-5

فروش این کتاب فقط از طریق نمایندگی‌های دانشگاه پیام نور مجاز می‌باشد و فروش آن در سایر مراکز فروش کتاب موجب تعقیب قانونی فروشندۀ خواهد گردید.

(کلیه حقوق مادی و معنوی این اثر محفوظ و متعلق به دانشگاه پیام نور است، تولید، کپی یا خلاصه‌برداری، نشر و ترجمه، چاپ، افسـت، توزیع گسترده، تهیه CD و ... این اثر یا بخشی از آن اعم از متن، طرح پشت و روی جلد و ... بدون اخذ مجوز کتبی از معاونت فناوری و پژوهش دانشگاه خلاف مقررات و تعرض به حقوق مادی و معنوی دانشگاه است. متخلفین وفق قانون حمایت از مؤلفان و مصنفان و هنرمندان دارای مستولیت حقوقی و کیفری بوده، و تحت پیگرد قرار می‌گیرند.)

قیمت: ریال

امروزه کتاب‌خوانی و علم‌آموزی، نه تنها یک وظیفه‌ی ملی، که یک واجب دینی است.

مقام معظم رهبری

در عصر حاضر یکی از شاخصه‌های ارزیابی رشد، توسعه و پیشرفت فرهنگی هر کشوری میزان تولید کتاب، مطالعه و کتاب‌خوانی مردم آن مرز و بوم است. ایران اسلامی نیز از دیرباز تاکنون با داشتن تمدنی چندهزارساله و مراکز متعدد علمی، فرهنگی، کتابخانه‌های معتبر، علما و دانشمندان بزرگ با آثار ارزشمند تاریخی، سرآمد دولت‌ها و ملت‌های دیگر بوده و در عرصه‌ی فرهنگ و تمدن جهانی بهسان خورشیدی تابناک همچنان می‌درخشد و با فرزندان نیکنہاد خویش هنرنمایی می‌کند. چه کسی است که در دنیا با دانشمندان فرزانه و نام‌آور ایرانی همچون ابوعلی سینا، ابوالیحان بیرونی، فارابی، خوارزمی و ... همچنین شاعران بر جسته‌ای نظیر فردوسی، سعدی، مولوی، حافظ و ... آشنا نباشد و در مقابل عظمت آنها سر تعظیم فرود نیاورد. تمامی این افتخارات ارزشمند، برگرفته از میزان عشق و علاقه فراوان ملت ما به فraigیری علم و دانش از طریق خواندن و مطالعه منابع و کتاب‌های گوناگون است. به شکرانه‌ی الهی، تاریخ و گذشته ما، همیشه درخشنan و پربار است. ولی اکنون در این زمینه در چه جایگاهی قرار داریم؟ آمار و ارقام ارائه شده از سوی مجتمع و سازمان‌های فرهنگی در مورد سرانه‌ی مطالعه‌ی هر ایرانی، برایمان چندان امیدوارکننده نمی‌باشد و رهبر معظم انقلاب اسلامی نیز از این وضعیت بارها اظهار گله و ناخشنودی نموده‌اند.

کتاب، دروازه‌ای به سوی گستره‌ی دانش و معرفت است و کتاب خوب، یکی از بهترین ابزارهای کمال بشری است. همه‌ی دستاوردهای بشر در سراسر عمر جهان، تا آنجا که قابل کتابت بوده است، در میان دستنوشته‌هایی است که انسان‌ها پدید آورده و می‌آورند. در این مجموعه‌ی بی‌نظیر، تعالیم الهی، درس‌های پیامبران به بشر، و همچنین علوم مختلفی است که سعادت بشر بدون آگاهی از آنها امکان‌پذیر نیست. کسی که با دنیای زیبا و زندگی بخش کتاب ارتباط ندارد بی‌شک از مهم‌ترین دستاورده انسانی و نیز از بیشترین معارف الهی و بشری محروم است. با این دیدگاه، به روشنی می‌توان ارزش و مفهوم رمزی عمیق در این حقیقت تاریخی را دریافت که اولین خطاب خداوند متعال به پیامبر گرامی اسلام(ص) این است که «بخوان!» و در اولین

سوره‌ای که بر آن فرستاده‌ی عظیم الشأن خداوند، فرود آمده، نام «قلم» به تجلیل یاد شده‌است: «إِقْرَأْ وَ رَبُّكَ الْأَكْرَمُ. الَّذِي عَلَّمَ بِالْقَلْمَنْ» در اهمیت عنصر کتاب برای تکامل جامعه‌ی انسانی، همین بس که تمامی ادیان آسمانی و رجال بزرگ تاریخ بشری، از طریق کتاب جاودانه مانده‌اند.

دانشگاه پیام‌نور با گستره‌ی جغرافیایی ایران‌شمول خود با هدف آموزش برای همه، همه‌جا و همه‌وقت، به‌عنوان دانشگاهی کتاب‌محور در نظام آموزش عالی کشورمان، افتخار دارد جایگاه اندیشه‌سازی و خردورزی بخش عظیمی از جوانان جویای علم این مرز و بوم باشد. تلاش فراوانی در ایام طولانی فعالیت این دانشگاه انجام پذیرفته تا با بهره‌گیری از تجربه‌های گرانقدر استادان و صاحب‌نظران برجسته کشورمان، کتاب‌ها و منابع آموزشی درسی شاخص و خودآموز تولید شود. در آینده هم، این مهم با هدف ارتقای سطح علمی، روزآمدی و توجه بیشتر به نیازهای مخاطبان دانشگاه پیام‌نور با جدیت ادامه خواهد داشت. به‌طور قطع استفاده از نظرات استادان، صاحب‌نظران و دانشجویان محترم، ما را در انجام این وظیفه‌ی مهم و خطیر یاری‌رسان خواهد بود. پیش‌اپیش از تمامی عزیزانی که با نقد، تصحیح و پیشنهادهای خود ما را در انجام این وظیفه‌ی خطیر یاری می‌رسانند، سپاسگزاری می‌نماییم. لازم است از تمامی اندیشمندانی که تاکنون دانشگاه پیام‌نور را منزلگه اندیشه‌سازی خود دانسته و ما را در تولید کتاب و محتوای آموزشی درسی یاری نموده‌اند، صمیمانه قدردانی گردد. موفقیت و بهروزی تمامی دانشجویان و دانش‌پژوهان عزیز آرزوی همیشگی ما است.

دانشگاه پیام‌نور

فهرست مطالب

یازده	پیشگفتار
۱	فصل اول: شناخت پایگاه‌داده‌ها و محیط پایگاهی
۱	۱ هدف کلی
۱	۲ هدف‌های یادگیری
۲	۳ مقدمه
۲	۴ ۱-۱ نسل‌های مختلف پایگاه‌داده
۲	۵ ۱-۱-۱ سیستم پروندهای
۳	۶ ۲-۱-۱ سیستم مدیریت داده‌ها
۳	۷ ۳-۱-۱ سیستم مدیریت پایگاه‌داده‌ها (DBMS)
۴	۸ ۴-۱-۱ سایر سیستم‌ها
۴	۹ ۲-۱ مدیریت پایگاه‌داده‌ها
۷	۱۰ ۱-۲-۱ تعریف پایگاه‌داده
۸	۱۱ ۲-۲-۱ انواع DBMS
۹	۱۲ ۳-۲-۱ اجزای DBMS
۱۰	۱۳ ۳-۱ ویژگی‌ها و امتیازهای پایگاه‌داده‌ها
۱۰	۱۴ ۱-۳-۱ پرسش
۱۲	۱۵ ۲-۳-۱ تراکنش
۱۵	۱۶ ۳-۳-۱ ذخیره و بازیابی بهینه داده‌ها
۱۸	۱۷ ۴-۱ مدل‌های پایگاه‌داده
۱۸	۱۸ ۱-۴-۱ مدل‌های ابتدایی (پیش‌رابطه‌ای)
۲۰	۱۹ ۲-۴-۱ مدل رابطه‌ای
۲۴	۲۰ ۳-۴-۱ مدل‌های جدید (فرا رابطه‌ای)
۲۴	۲۱ ۵-۱ معماřی پایگاه‌داده‌ها

۲۴	۱-۵-۱ معماری چندلایه‌ای
۲۷	۲-۵-۱ اجزای لایه‌ها
۳۰	۱-۶ طراحی پایگاهداده‌ها
۳۱	۱-۶-۱ مراحل طراحی پایگاهداده‌ها
۳۲	۲-۶-۱ ابزار طراحی پایگاهداده‌ها
۳۲	۷-۱ جداول نمونه
۳۳	خلاصه فصل اول
۳۴	خودآزمایی تشریحی فصل اول
۴۱	تمرین‌های تشریحی فصل اول
۴۳	فصل دوم: مدل‌سازی معنایی
۴۳	هدف کلی
۴۳	هدف‌های یادگیری
۴۴	مقدمه
۴۵	۱-۲ روش‌های مدل‌سازی معنایی
۴۶	۲-۲ مدل‌سازی با روش ER
۴۷	۱-۲-۲ نوع پدیده یا موجودیت
۴۸	۲-۲-۲ صفت و انواع آن
۵۲	۳-۲-۲ نوع ارتباط
۵۳	۳-۲ نمودار ER
۵۳	۱-۳-۲ صفت در نوع ارتباط
۵۶	۲-۳-۲ وضعیت مشارکت در ارتباط
۵۶	۳-۳-۲ چندی یا ماهیت ارتباط (کاردینالیتی)
۵۹	۴-۳-۲ حد ارتباط
۶۰	۵-۳-۲ نوع پدیده ضعیف
۶۱	۶-۳-۲ صفت ممیزه
۶۲	۷-۳-۲ نوع ارتباط به مثابه نوع پدیده (ارتباط پدیده‌ای)
۶۲	۴-۲ مشکلات ER
۶۳	۱-۴-۲ دام حلقه‌ای
۶۴	۲-۴-۲ دام چندشاخه (چتری)
۶۵	۳-۴-۲ دام گسل (شکاف)
۶۷	۴-۴-۲ برخی تصمیم‌گیری‌ها در مدل‌سازی به روش ER
۶۷	۵-۲ مدل‌سازی با روش EER
۶۷	۱-۵-۲ تجزیه و ترکیب
۶۹	۲-۵-۲ تخصیص
۷۲	۳-۵-۲ تجمیع
۷۴	۶-۲ آشنایی با روش مدل‌سازی UML

۷۶	۱-۶-۲ اجزای UML
۷۶	۲-۶-۲ نحوه نمایش مفاهیم
۷۸	۷-۲ آشنایی با روش مدل‌سازی NIAM
۸۱	خلاصه فصل دوم
۸۲	خودآزمایی تشریحی فصل دوم
۸۵	تمرین‌های تشریحی فصل دوم
۸۷	فصل سوم: شناخت مدل رابطه‌ای و زبان بیانی SQL
۸۷	هدف کلی
۸۷	هدف‌های یادگیری
۸۸	مقدمه
۸۸	۱-۳ مدل داده رابطه‌ای
۹۱	۲-۳ قواعد جامعیت
۹۳	۳-۳ انواع کلید
۹۵	۴-۳ انواع جامعیت در مدل رابطه‌ای
۹۵	۵-۳ زبان روالی SQL
۹۶	۱-۵-۳ جداول
۹۷	۲-۵-۳ انواع داده استاندارد
۹۸	۳-۵-۳ نوع داده کاربر-تعريف
۹۸	۴-۵-۳ تعریف پایگاهداده
۹۹	۵-۵-۳ تعریف جدول
۱۰۲	۶-۵-۳ تغییر جدول موجود
۱۰۳	۷-۵-۳ بارگذاری پایگاهداده
۱۰۸	۸-۵-۳ استخراج اطلاعات
۱۳۴	خلاصه فصل سوم
۱۳۴	تمرین‌های تشریحی فصل سوم
۱۳۷	فصل چهارم: جبر و حساب رابطه‌ای
۱۳۷	هدف‌های کلی
۱۳۷	هدف‌های یادگیری
۱۳۸	مقدمه
۱۳۸	۱-۴ جبر رابطه‌ای
۱۳۸	۱-۱-۴ انواع عملگرها
۱۴۸	۲-۱-۴ بهینه‌سازی پرسش
۱۵۴	۳-۱-۴ بهروز درآوردن داده‌ها
۱۵۶	۲-۴ حساب رابطه‌ای دامنه‌ای
۱۵۶	۱-۲-۴ معرفی

۱۵۷	۲-۲-۴ انواع دستورها
۱۵۹	۳-۲-۴ خطر حلقه بی پایان
۱۵۹	۴-۲-۴ مقایسه با جبر رابطه‌ای و SQL
۱۶۰	خلاصه فصل چهارم
۱۶۱	خودآزمایی‌های تشریحی فصل چهارم
۱۶۲	تمرین‌های تشریحی فصل چهارم
۱۶۵	فصل پنجم: نرمال‌سازی
۱۶۵	هدف کلی
۱۶۵	هدف‌های یادگیری
۱۶۶	مقدمه
۱۶۶	۱-۵ مشکلات جداول غیرنرمال
۱۶۸	۲-۵ وابستگی و نرمال‌سازی
۱۶۹	۳-۵ وابستگی تابعی
۱۷۰	۱-۳-۵ تعاریف
۱۷۲	۲-۳-۵ یافتن مجموعه پوششی و مجموعه بهینه وابستگی
۱۷۴	۳-۳-۵ کلیدهای کاندید
۱۷۵	۴-۳-۵ نمودار وابستگی تابعی
۱۷۷	۴-۵ تجزیه جداول به فرم‌های نرمال در وابستگی تابعی
۱۷۷	۱-۴-۵ فرم نرمال اول
۱۷۸	۲-۴-۵ فرم نرمال دوم
۱۷۹	۳-۴-۵ صحت تجزیه جدول
۱۸۰	۴-۴-۵ فرم نرمال سوم
۱۸۲	۵-۴-۵ فرم نرمال BCNF
۱۸۶	۵-۵ وابستگی چندمقداری و فرم نرمال چهارم
۱۸۸	۶-۵ وابستگی پیوندی و فرم نرمال پنجم
۱۹۰	خلاصه فصل پنجم
۱۹۰	خودآزمایی تشریحی فصل پنجم
۱۹۲	تمرین‌های تشریحی فصل پنجم
۱۹۵	فصل ششم: آشنایی با مدل شی - رابطه‌ای و SQL3
۱۹۵	هدف کلی
۱۹۵	هدف‌های یادگیری
۱۹۵	مقدمه
۱۹۶	۱-۶ کاستی‌ها و امتیازهای مدل رابطه‌ای
۲۰۱	۲-۶ شی‌گرایی در پایگاهداده‌ها
۲۰۳	۱-۲-۶ مفاهیم بنیادین مدل شی‌گرا

۲۰۶	۲-۲-۶ مدل شی - رابطه‌ای
۲۰۹	۳-۶ معرفی SQL3
۲۰۹	۱-۳-۶ ویژگی‌های غیرشی‌گرایی
۲۱۱	۲-۳-۶ ویژگی‌های شی‌گرایی
۲۱۲	۴-۶ مقایسه
۲۱۳	۶-۵ ارتباط پایگاهداده با زبان‌های برنامه‌سازی
۲۱۴	خلاصه فصل ششم
۲۱۵	تمرین‌های تشریحی فصل ششم
۲۱۷	فصل هفتم: آشنایی با مفاهیم پیشرفته
۲۱۷	هدف کلی
۲۱۷	هدف‌های یادگیری
۲۱۸	مقدمه
۲۱۸	۱-۷ شاخص
۲۱۹	۲-۷ کلانداده و رایانش ابری
۲۲۲	۳-۷ NoSQL
۲۲۴	۴-۷ داده‌کاوی
۲۲۷	۵-۷ سیستم مدیریت جریان‌داده‌ها
۲۳۰	۶-۷ پایگاهداده توزیعی(نامتمرکز)
۲۳۴	۷-۷ پایگاهداده سیار (موبایل)
۲۳۶	خلاصه فصل هفتم
۲۳۷	تمرین‌های تشریحی فصل هفتم
۲۳۹	واژه‌نامه فارسی - انگلیسی
۲۴۵	واژه‌نامه انگلیسی - فارسی
۲۵۱	منابع

پیشگفتار

چارلی چاپلین در ملاقات با آلبرت اشتاین: می‌دانی چرا این همه مردم از من استقبال می‌کنند؟ زیرا همه می‌فهمند من چه می‌گویم ولی هیچ‌کدام نمی‌فهمند تو چه می‌گویی!

در سال ۱۳۷۷ کتاب یک‌جلدی بانک اطلاعات علمی - کاربردی نوشته اینجانب، مصطفی حق‌جو توسط انتشارات دانشگاه علم و صنعت ایران چاپ و منتشر شد. این کتاب که بعداً به دو جلد برای دوره‌های کارشناسی و کارشناسی ارشد تبدیل شد، هنوز نیز پرطرفدار و بارها به روز شده‌است. از آن زمان تاکنون علم و تکنولوژی پیشرفت زیادی کرده و نیاز به کتاب جدید پایگاهداده‌ها احساس می‌شود. شگفت‌آور نیست که مفاهیم اصلی کمابیش همان است که بود! زیرا پایگاهداده از مفاهیم بنیادین مهندسی نرم‌افزار است و ریشه‌های آن تغییر نکرده ولی صد البته هر روز تناورتر و پربارتر شده‌است. از سوی دیگر دانشگاه بزرگ پیام‌نور، که به وسعت ایران بزرگ گستردۀ است، نیازهای متفاوتی دارد. در این دانشگاه، دانشجویان به خوداتکایی عادت دارند.

کتاب‌های درسی این دانشگاه باید کاملاً خودخوان و همراه با مسائل حل شده و خودآزمایی‌های زیاد باشد. لازم به یادآوری است که امکاناتی چون تست‌های چهارگزینه‌ای را دانشگاه در اختیار دانشجویان قرار می‌دهد.

کتاب حاضر که از مبانی تا پیشرفت‌های اخیر را در پایگاهداده‌ها، در حد دورۀ کارشناسی می‌پوشاند، کتابی جدید و مبتنی بر نیازهای کنونی است. در این دوران که پایگاهداده‌ها به خوبی جای خود را در برنامه‌سازی کاربردی در زمینه‌های مختلف باز کرده‌است، اهمیت این مطلب دوچندان است. در این کتاب معماری و طراحی

پایگاهداده و پیادهسازی آن در مدل‌های رابطه‌ای و شئ – رابطه‌ای و نیز آشنایی با مفاهیم جدید در پایگاهداده‌ها گنجانده شده است.

از خوانندگان گرامی، به‌ویژه اساتید معظم، متخصصان و کارشناسان ارجمند و دانشجویان عزیز کامپیوتر و فناوری اطلاعات تقاضای نقد راه‌گشا داریم.

از کسانی که در تدوین این کتاب همکاری کرده‌اند صمیمانه سپاسگزاری می‌شود:

سپاس بیکران از همکار ارجمند جناب آقای دکتر اکبر فرهودی‌نژاد و دستیارانشان خانم‌ها شیما واحد و هاجر شایق برای ویراستاری کل کتاب،

سپاس ویژه از دانشجویان مهندسی کامپیوتر و فناوری اطلاعات:

سرکار خانم فاطمه بیژه محمدزاده برای تایپ و تدوین کل کتاب،

سرکار خانم محبوبه طغیانی برای پیشنهاد برخی از مسائل کاربردی،

آقای محمدرضا قربانی برای اجرای پرسش‌های SQL،

آقای ابوذر جباری برای همکاری در تدوین و تایپ فصل چهارم.

آقای علیرضا احمدیان برای همکاری در پایگاهدانش.

با احترام:

دکتر احمد فراهی

دکتر مصطفی حق‌جو سانیجی

استادیار مهندسی نرم‌افزار

دانشیار مهندسی نرم‌افزار

دانشگاه پیام‌نور – سازمان مرکزی

دانشگاه پیام‌نور – مرکز بین‌المللی کیش

فصل اول

شناخت پایگاهداده‌ها و محیط پایگاهی

هدف کلی

شناخت محیط پایگاهی، مدل‌ها و معماری پایگاهداده‌ها و مراحل طراحی آن

هدف‌های یادگیری

۱. خوانندگان گرامی قادر به بازشناسایی جایگاه پایگاهداده‌ها خواهند بود. داده‌پردازی در جهان مدرن بدون استفاده صحیح و علمی از پایگاهداده‌ها و شناخت درست از آن امکان‌پذیر نخواهد بود.
۲. اتوماسیون و گرفتن کنترل امور از دست انسان‌ها و دادن آن به نرم‌افزارهای توانمندی که سرعت و دقّت و خستگی‌ناپذیری غیرقابل مقایسه دارند، به‌ویژه نرم‌افزار مدیریت پایگاهداده‌ها.
۳. آشنایی با مفاهیم بنیادی چون پرسش و تراکنش و امنیت و ذخیره‌سازی داده‌های بزرگ و ترمیم خرابی‌ها و حفظ جامعیت اطلاعات و نظایر آن‌ها، که توسط نرم‌افزارهای قدرتمند اجرا و اعمال می‌شوند.
۴. خوانندگان گرامی با ابعاد مختلف چون مدل‌های گوناگون پایگاهداده‌ها، لایه‌ها و مراحل معماری و طراحی پایگاهداده‌ها آشنا می‌شوند تا در فصل‌های بعدی به کندوکاو آن‌ها بپردازند.

مقدمه

بیشتر مخاطب‌های این نوشتار دانشجویان و دانشآموختگان عزیزی هستند که تا برنامه‌سازی در زبان‌های سطح بالا (مانند سی و پاسکال و جاوا) پیش‌رفته‌اند و با مفاهیم مقدماتی چون داده و اطلاعات و... آشنا هستند. برای این افراد احتمالاً این سؤال پیش‌آمده که چرا برنامه‌سازی در سطح وسیع و تکنیکی، این همه، گیر و گرفتاری دارد! این سؤال و جنبه‌های دیگر مشکلات برنامه‌سازی چون حفاظت از داده‌ها و ذخیره و بازیابی امن و آسان و بهینه اطلاعات باعث پیدایش پایگاهداده‌ها^۱ در دهه ۱۹۶۰ شد. در این مشی، داده‌ها به خوبی نگهداری می‌شوند، دستیابی به آن‌ها امن و سریع است و استخراج اطلاعات به راحتی صورت می‌گیرد.

مثال:

زبان‌های سطح بالا را زبان روالی^۲ نیز می‌نامند، زیرا برنامه‌ساز باید ریزبه‌ریز، به کامپیوتر بگوید چه کاری انجام دهد. مثلاً برای جستجوی ساده یک نام در یک لیست، باید برنامه در یک حلقة تکرار از ابتدا تا انتهای لیست به‌دنبال آن نام بگردد (حلقة تکرار و گشتن را باید برنامه‌ساز سامان دهد). در مقابل، در پایگاهداده‌ها دستورهای سطح بالاتری وجود دارد که به راحتی این کار را انجام می‌دهد. به عنوان نمونه در زبان پایگاهی SQL کافی است نوشته شود «در لیست فلان، داده بهمان را انتخاب کن».

طبعی است که این تکنولوژی به یکباره به دست نیامده و نسل‌های قبل از خود داشته است. هدف اصلی برنامه‌سازی، دست کم در آن دوران، سپردن داده‌ها به حافظه کامپیوتر، بازیابی آن‌ها و انجام عمل‌های مختلف روی داده‌ها می‌باشد. سرعت عمل، دقت، جلوگیری از دستیابی‌های غیرمجاز و... بخش‌های جدایی‌ناپذیر این روند هستند. در راستای این هدف، چندین رده و نسل فناورانه به وجود آمده است.

۱-۱ نسل‌های مختلف پایگاهداده

۱-۱-۱ سیستم پروندهای^۳

در این سیستم، کاربران مختلف با پروندهایی که داده‌های مورد نیاز آن‌ها را در خود داشتند، برنامه‌های جداگانه‌ای اجرا می‌کردند و اطلاعات مورد نیاز را به دست

1. Data Base
2. Procedural Language
3. Filing System

می‌آوردند. در موارد زیادی داده‌های کاربران مختلف هم‌پوشانی داشتند یعنی اقلام داده در پرونده‌های مختلف تکرار می‌شدند. این پرونده‌ها به صورت عادی در قفسه‌هایی نگهداری می‌شدند و تحت مدیریت نرم‌افزار نبودند. به عبارت دیگر انسان‌ها به پرونده‌ها دسترسی داشتند و احتمال دست‌یابی غیرمجاز و تغییر داده‌ها وجود داشت. نگارنده تجربیات تلخی در رابطه با استفاده از این نوع پرونده‌ها در سال‌های دور دارد. در حالی که دستیابی غیرمجاز به پرونده‌ها صورت گرفته بود، ولی قابل اثبات نبود، زیرا هیچ نرم‌افزاری این دستیابی‌ها را کنترل نمی‌کرد.

۱-۱-۲ سیستم مدیریت داده‌ها^۱

در این سیستم نرم‌افزارهای مدیریتی وارد صحنه شده و تا حدی از دستیابی‌های افراد که می‌تواند خطرآفرین باشد، کاسته‌اند. این سیستم‌ها همه نیازها را پاسخ نمی‌دادند. در حقیقت این سیستم‌ها شروع یک انقلاب بودند که ذخیره و بازیابی و نگهداری و حفاظت از داده‌ها را از دست انسان‌ها گرفت و به نرم‌افزارها سپرد. از آنجاکه نرم‌افزارها بیشتر قابل کنترل هستند و می‌توانند هرگونه تغییر یا عملی را به راحتی، به صورت محترمانه گزارش کنند، عملکرد آن‌ها از انسان‌ها قابل اعتمادتر و امن‌تر است.

به عنوان نمونه یک سیستم بانکی که به‌طور خودکار هر عمل و تغییر روی حساب افراد را به آن‌ها گزارش می‌کند، بهتر از افرادی است که ممکن است به‌دلایلی، گاهی سوءاستفاده کنند. می‌توان گفت که سیستم مدیریت داده‌ها به دوران کودکی و ناپختگی این انقلاب که می‌کوشد مدیریت سیستم را به نرم‌افزار بسپارد، بیشتر شبیه است.

۱-۱-۳ سیستم مدیریت پایگاه‌داده‌ها^۲ (DBMS)

در این نوع سیستم استفاده از نرم‌افزارهای کنترلی داده‌ها به بلوغ می‌رسد، به‌طوری‌که به‌ندرت و در شرایط استثنائی به دخالت انسان نیاز است؛ به عبارت دیگر، در این سیستم‌ها، داده‌ها به‌طور کامل در اختیار نرم‌افزار DBMS است که امنیت و حفاظت از آن‌ها را تا حد بسیار زیادی تضمین می‌کند و از دستیابی انسان‌ها به داده‌ها و اطلاعات، مگر در موارد بسیار نادر، جلوگیری می‌کند.

1. Data Management System

2. Data Base Management System (DBMS)

مفهوم پایگاهداده‌ها در این نرمافزار نهفته است. در بخش‌های آینده، این انقلاب نرمافزاری را از جنبه‌های مختلف تئوری و عملی، از طراحی بهینه در سطوح مختلف و پیاده‌سازی کارا و امن تا بهره‌برداری از پایگاهداده‌ها که می‌تواند یک مجموعه بسیار بزرگ و پراکنده (نامتمرکز) باشد، مطالعه خواهیم کرد.

۱-۱-۴ سایر سیستم‌ها

کاربردهای مختلف، نیازهای متفاوت و پیشرفت روزافزون علم و فناوری باعث به وجود آمدن سیستم‌های دیگر شد که برخلاف موارد بالا بیشتر خاص منظوره هستند. سیستم مدیریت جغرافیایی، سیستم داده‌کاوی و سیستم پایگاه دانش، نمونه‌ای از این سیستم‌ها هستند.

از سوی دیگر، سیستم مدیریت پایگاهداده‌ها نیز، به سمت و سوی تخصص‌های مختلف حرکت کرد و سیستم‌هایی چون سیستم هوشمند مدیریت پایگاهداده‌ها، سیستم بی‌درنگ مدیریت پایگاهداده‌ها، سیستم توزیعی مدیریت پایگاهداده‌ها و... به وجود آمدند. این‌گونه سیستم‌ها از حوصله این کتاب خارج‌اند. بخشی از آن‌ها در کتاب و درس دیگری به نام «پایگاهداده پیشرفتی» مورد کنکاش قرار می‌گیرند.

۲ مدیریت پایگاهداده‌ها

جان کلام این است که در پایگاهداده‌ها (پایگاهداده) همه چیز توسط نرمافزار توانمند و قابل اعتمادی به نام نظام مدیریت پایگاهداده انجام و کنترل می‌شود. کاربران پایگاهداده پرسش‌های^۱ خود را به این نرمافزار تحويل می‌دهند و پاسخ خود را (که ممکن است سربالا باشد!) دریافت می‌کنند. اگر کاربر مجاز به استفاده از داده‌های مورد نظر و انجام کارهای درخواستی باشد، DBMS نتایج را در اختیار او قرار می‌دهد و پرونده‌ها را به روز می‌کند و نیز ریزکارهای انجام‌شده را طوری یادداشت‌برداری می‌کند (در کارنامه پایگاهداده) که همواره قابل پیگیری و ردیابی و در صورت لزوم بازگرد باشد.

مثال:

در یک سیستم معمولی، کاربر بانک به شعبه‌ها مراجعه می‌کند و آن‌ها (از طریق

DBMS یا بدون آن) از حساب او پول بر می‌دارند و تحویلش می‌دهند. اگر پول کاربر در شهر دیگری باشد، ممکن است به آن دسترسی نداشته باشد. در یک سیستم پایگاه‌داده‌ای کاربر به یک خودپرداز مراجعه می‌کند. خودپرداز که توسط DBMS مدیریت می‌شود، ضمن تعامل با کاربر از حساب او کسر و پول را تحویلش می‌دهد. طبیعی است که کاربر نمی‌تواند از حساب شخص دیگری پول بردارد یا بیش از موجودی برداشت یا از حد مجاز برداشت در روز، تجاوز کند. اگر خودپرداز در بین راه دچار مشکل شود و نتواند درخواست کاربر را اجرا و پول و کارت را در اختیارش قرار دهد، پول برداشت شده را بر می‌گرداند و به حسابش واریز می‌کند. این عمل نوعی بازگرد است.

در روش‌های سابق، کاربر، پروندهای را در اختیار می‌گرفت و عملیات موردنظر خود را روی آن انجام می‌داد. در پایگاه‌داده، تمامی پروندها فقط و فقط در اختیار این نرم‌افزار قدرتمند (DBMS) است و کاربران درخواست‌های خود را تسليم می‌کنند تا در صورت صلاح دید آن را انجام دهد. مثلاً اگر کسی بخواهد از حساب بانکی خود پول برداشت کند، درخواست او تسليم DBMS می‌شود. این نرم‌افزار پس از کترل‌های لازم و اطمینان از صحت کار، پول را از حساب کسر می‌کند و در اختیار کاربر قرار می‌دهد. برای راهبری این نرم‌افزار، دادن اطلاعات به آن (مثل اطلاعات امنیتی درباره اجازه استفاده کاربران از پروندهای مختلف)، انجام تغییرات و امور مشابه به دو نوع نیروی انسانی نیاز است:

الف) مدیر پایگاه‌داده^۱، که مسئولیت تصمیم‌گیری و طراحی این‌گونه موارد را به عهده دارد. این مسئولیت ممکن است به یک شخص و یا در پایگاه‌داده بزرگ به یک تیم محول شود.

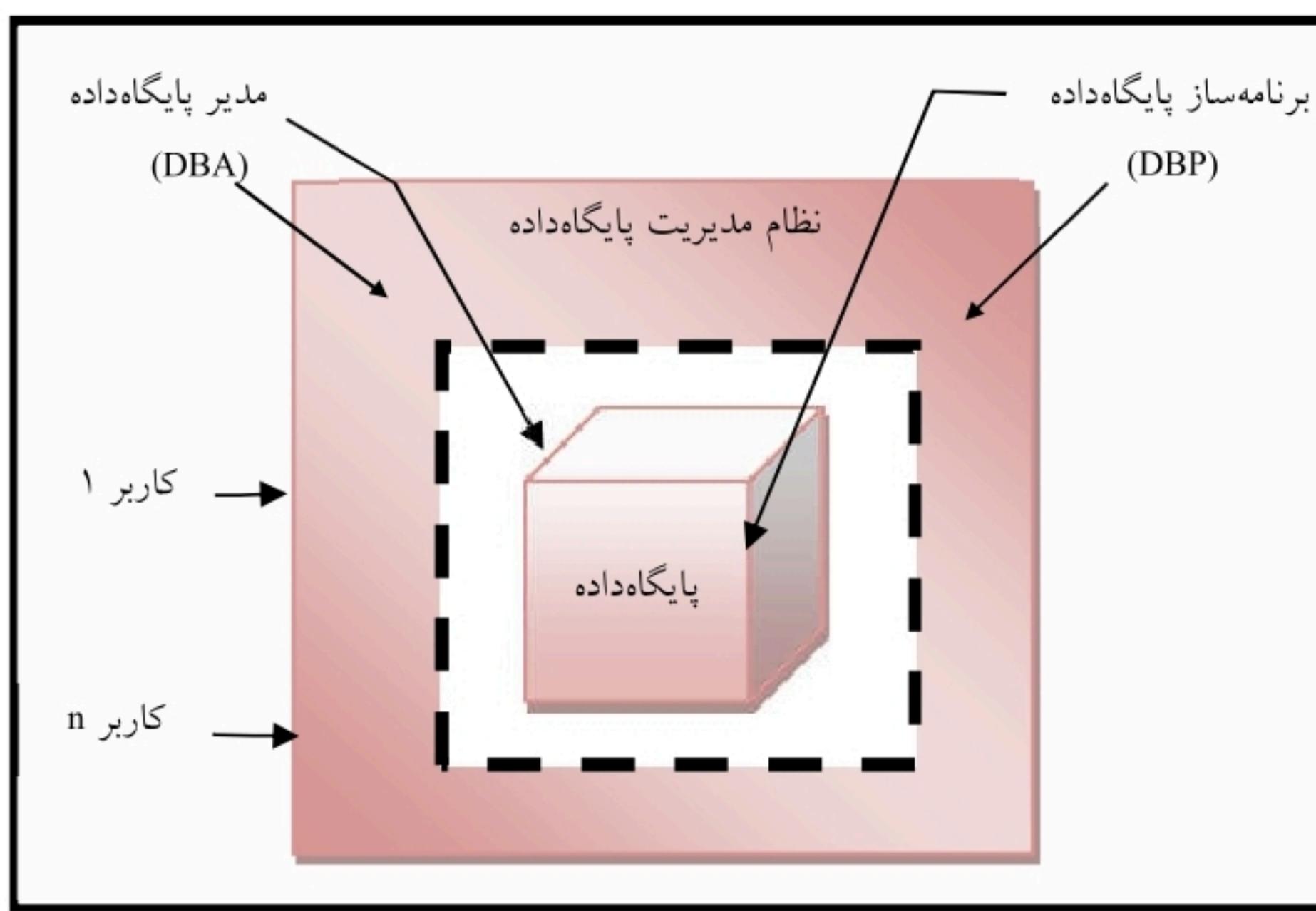
ب) برنامه‌ساز پایگاه‌داده^۲، که تصمیمات مدیر را پیاده‌سازی می‌کند. برنامه‌ساز را نباید با اپراتور اشتباه گرفت. اپراتورها مسئول اجرای برنامه‌ها و راهبری سیستم هستند، در حالی که برنامه‌سازان تصمیمات مدیر را به صورت برنامه پیاده‌سازی می‌کنند تا توسط اپراتورها اجرا شود.

شکل ۱-۱ نظام پایگاه‌داده^۳ را نشان می‌دهد. تنها کسانی که می‌توانند دور از چشم نظام مدیریت پایگاه‌داده (DBMS) به داده‌ها دسترسی داشته باشند، مدیر و

1. Data Base Administrator (DBA)
2. Data Base Programmer (DBP)
3. Data Base System (DBS)

برنامه‌سازان مجاز پایگاه‌داده هستند. طبیعی است که نرم‌افزار نظام مدیریت پایگاه‌داده نمی‌تواند همه نوع کاری انجام دهد. مدیر و برنامه‌سازان مجاز می‌توانند از طریق نرم‌افزارهای کمکی^۱ نیز به پایگاه‌داده دسترسی داشته باشند. مثلاً اگر نیاز به سازماندهی مجدد باشد، این عمل با دستور مدیر توسط برنامه‌ساز مجاز و با کمک نرم‌افزار کمکی ویژه‌ای انجام می‌شود.

در (شکل ۱-۱) پنج عنصر اصلی پایگاه‌داده دیده می‌شوند: کاربر، راهبر، داده، نرم‌افزار و سخت‌افزار.



شکل ۱-۱. نظام پایگاه‌داده

کاربران (end users) یا users از چندوچون داده‌ها و نرم‌افزارها اطلاع چندانی ندارند. آن‌ها با بخش کوچکی از داده‌ها کار دارند و ممکن است پرسش‌هایی داشته باشند و اطلاعاتی دریافت کنند یا در داده‌ها تغییری ایجاد نمایند که توسط DBMS انجام می‌گیرد. راهبران که شامل مدیر و برنامه‌ساز پایگاه‌داده می‌شوند، راهبری DBMS را بر عهده دارند (مانند بارگذاری، رفع اشکالات، برآوردن نیازمندی‌ها). علاوه بر مدیر و برنامه‌ساز پایگاه‌داده، افراد دیگری مانند طراحان پایگاه‌داده هم در سیستم کار می‌کنند که درجای خود از آن‌ها صحبت می‌شود.

داده‌ها، اقلام ساده دارای ارزش و معنی هستند که در پایگاه‌داده‌ها ذخیره و بازیابی می‌شوند. از مشخصات کاربران، حق دستیابی و نحوه استفاده آن‌ها از اقلام داده تا اطلاعات مربوط به پرونده‌ها، همه و همه از نوع داده‌ها هستند.

نرم‌افزار اصلی، همان نظام مدیریت پایگاه‌داده است (DBMS) و نرم‌افزارهای دیگری برای بارگذاری DBMS یا شکل‌دهی و زیباسازی اطلاعات خروجی و کارهای مشابه نیز مورد استفاده قرار می‌گیرند.

سخت‌افزار، شکل کامپیوترا (رایانه‌ها) و ابزار جانبی آن‌ها است که اجرای نرم‌افزارها و سایر موارد توسط آن‌ها انجام می‌گیرد. مثلاً دستگاه خودپرداز بخشی از سخت‌افزار است. مهم‌ترین سخت‌افزار در پایگاه‌داده‌ها، حافظه جانبی کامپیوتر (چون داده‌ها روی آن ذخیره می‌شود) و کیفیت و سرعت کار با آن است و متداول‌ترین نوع حافظه جانبی، دیسک نام دارد. اهمیت این نوع سخت‌افزار از آن جهت است که پایگاه‌داده‌ها خاصیت ماناگی دارد، یعنی آنچه در پایگاه‌داده‌ها ذخیره می‌شود برای مدت‌های طولانی نگهداری می‌شود، به روز می‌شود و مورد استفاده قرار می‌گیرد. سایر سخت‌افزارها شامل پردازشگرهای کامپیوتر، ابزار دیگر آن و سخت‌افزار هم‌رسانی، مانند مودم است که ارتباط‌های از راه دور و نزدیک را سامان می‌دهد.

درباره سیستم مدیریت پایگاه‌داده (DBMS) که بخش عمده و بازیگر اصلی این صحنه است، گفتنی‌ها فراوان است. بعضی از آن‌ها را در زیر می‌آوریم تا در جای خود مورد کنکاش بیشتر قرار گیرند. ابتدا پایگاه‌داده را تعریف می‌کنیم.

۱-۲-۱ تعریف پایگاه‌داده

تعریف متعددی از دیرباز برای پایگاه‌داده‌ها ارائه شده که هر کدام جنبه‌هایی از آن را مورد توجه خاص قرار داده‌اند [روحانی، ۱۳۹۲]. در اینجا تعریف جامعی بر مبنای همه جنبه‌های پایگاه‌داده‌ها ارائه می‌کنیم.

تعریف: پایگاه‌داده، مجموعه‌ای از داده‌های پایا (ماندگار) و یکپارچه و مرتبط با یکدیگر است که به صورت بهینه روی یک یا چند رایانه توسط نظام مدیریت پایگاه‌داده‌ها ذخیره، بازیابی و کنترل می‌شود و به صورت اشتراکی مورد استفاده یک یا چند کاربر یا سیستم کاربردی قرار می‌گیرد.

۲-۱ انواع DBMS

از دهه ۱۹۷۰ تاکنون انواع مختلف DBMS طراحی، تولید و ارائه شده است. پابه‌پایی پیشرفت در علم و تکنولوژی کامپیوتر به‌ویژه پایگاه‌داده‌ها، این نرم‌افزار نیز پیشرفت کرده و انواع مختلف آن، چه از نظر مدل پایگاه‌داده، سیستم‌عامل، محیط‌های نرم‌افزاری، نوع رایانه، نوع واسط کاربر و غیره ارائه شده است. به عنوان نمونه SQLServer را می‌توان نام برد که نسل‌های متفاوتی دارد و از متداول‌ترین نظام‌های مدیریت پایگاه داده رابطه‌ای است و همچنین اوراکل که از کامل‌ترین آن‌ها است. این دو سیستم، در محیط سیستم‌عامل ویندوز بسیار متداول‌اند، ولی تحت برخی سیستم‌عامل‌های دیگر هم ارائه شده‌اند. یکی از مهم‌ترین ابعاد که باید در خرید و نصب این سیستم‌ها مورد نظر باشد، این است که آیا این نرم‌افزار، پایگاه‌داده نامتمرکز^۱ را پشتیبانی می‌کند یا خیر. در حالت نامتمرکز، داده‌ها و پرسش‌ها روی چند سایت پخش شده‌اند.

مثال:

در سیستم بانکداری تحت شبکه T (مانند شتاب) حساب کاربران در بانک‌های مختلف نگهداری می‌شود و درخواست کاربران نیز از خودپردازهای مختلف صادر می‌شود. کاربری که از خودپرداز سپه، درخواست برداشت پول از حسابش را در بانک ملی می‌دهد، عملاً دارد یک عملیات نامتمرکز انجام می‌دهد. اگر شما بخواهید عملیات نامتمرکز در محیط پایگاه‌داده انجام دهید باید DBMS از نوع نامتمرکز را روی رایانه‌های خود نصب کرده باشید.

معمولًا DBMS‌های عمومی و پیشرفته پرتابل هستند، یعنی می‌توان نسخه‌های مختلف آن‌ها را روی کامپیوترهای متفاوت نصب کرد. می‌توان داده‌های اصلی را مثلاً روی یک کامپیوتر بزرگ چون Work Staion و Mainframe و حتی^۲ Super Computer ذخیره کرد و از رایانه‌های کوچک‌تر مانند PC از محل‌های مختلف به آن داده‌ها دستیابی داشت. به این‌گونه معماری‌ها، «مشتری-کارگزار»^۳ می‌گویند. این‌گونه پایگاه‌های داده که امروزه بسیار متداول هستند، در واقع پایگاه‌داده توزیعی (نامتمرکز) نیستند. شرط توزیعی بودن این است که داده‌ها توزیع شده باشند.

1. Distributed

۲. معمولًا نرم‌افزارهای سوپر کامپیوتر از موارد دیگر متفاوت هستند، زیرا پردازش موازی بخشی از ماهیت آن‌ها است.

3. Client-Server

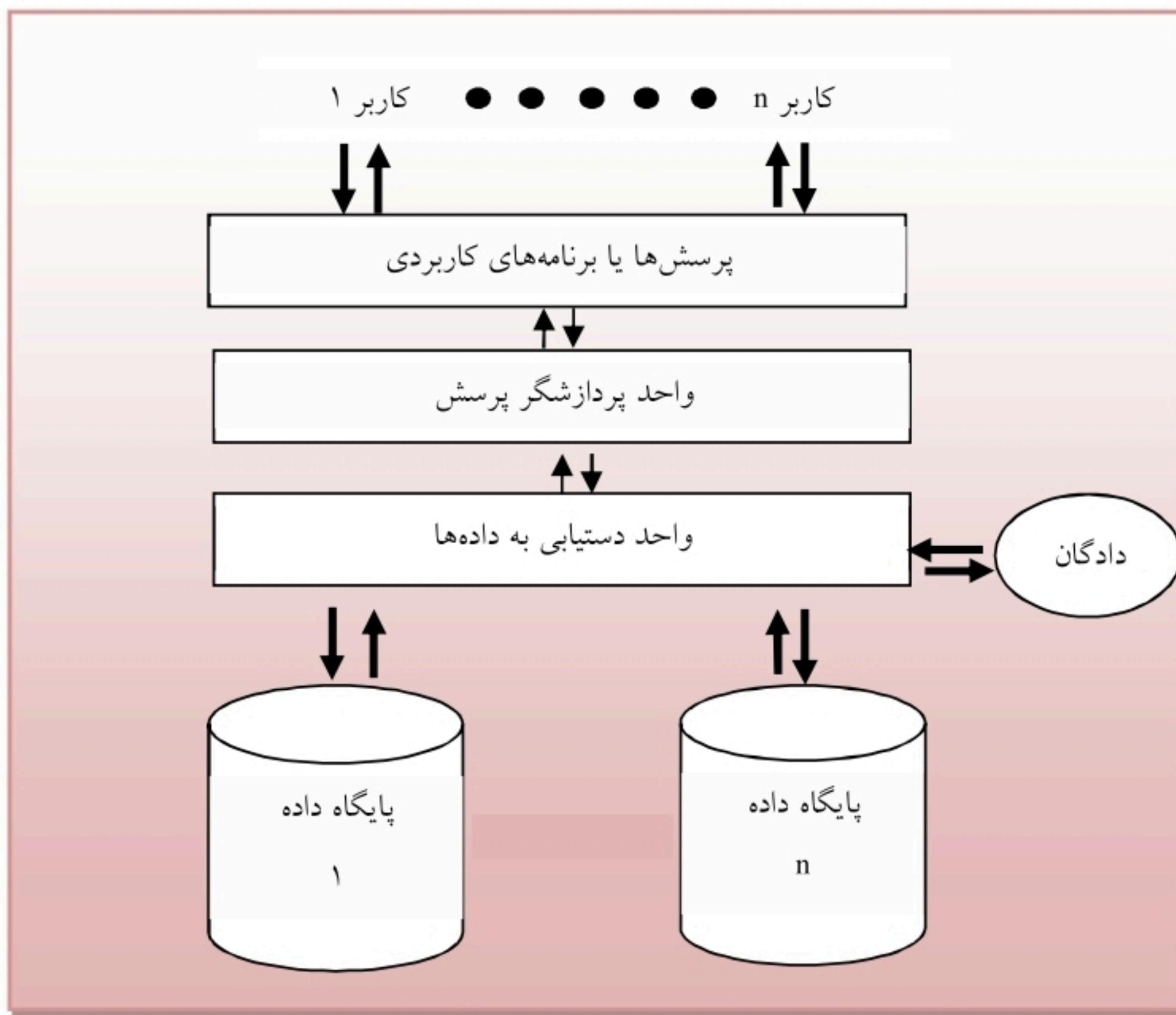
نوع دیگری از این معماری که به «چند مشتری - چند کارگزار» موسوم است، به این صورت است که داده‌ها روی چند کارگزار پخش شده‌اند و رایانه‌های مشتری در جاهای مختلف به یک یا چند کارگزار دسترسی دارند.

نرم‌افزار DBMS ممکن است تک‌منظوره یا چند‌منظوره باشد. تک‌منظوره یعنی برای کاربرد خاصی طراحی و ارائه شده‌است، مانند سیستم‌های مدیریت بانکداری. سیستم‌هایی که در دانشگاه‌ها متداول‌اند چند‌منظوره هستند یعنی هر پایگاه‌داده معمولی را می‌توان روی آن‌ها ایجاد و اجرا کرد. از نظر واسط کاربر نیز سیستم‌های مدیریت پایگاه‌داده متفاوت هستند، یعنی از دستورهای معمولی SQL و خروجی‌های ساده تا واسط کاربرهای گرافیکی وجود دارند. امروزه برنامه‌سازی با SQL با استفاده از واسط کاربرهای گرافیکی در محیط ویندوز بسیار متداول است.

DBMS اجزای ۲-۳

نرم‌افزار DBMS اجزای مختلفی دارد. نمای بیرونی این نرم‌افزار شامل «واحد پردازشگر پرسش^۱» که برنامه‌های کاربردی را نیز دربر می‌گیرد و «واحد دستیابی به داده‌ها»^۲ است. داده‌ها که روی حافظه‌های جانبی (رسانه‌ها) ذخیره شده‌اند در اختیار این واحد قرار دارند. شکل ۲-۱ (برگرفته از [روحانی، ۱۳۹۲] با اندکی تغییر) نمای بیرونی DBMS را نشان می‌دهد.

نمای درونی DBMS نیز از لایه‌های مختلف (معمولًاً سه لایه) تشکیل شده که هماهنگ با سایر نرم‌افزارها چون کامپایلرهای مدیریت پروندهای مدیریت حافظه، مدیریت هم‌روندي و همگي تحت نظارت سیستم عامل وظایف خود را انجام می‌دهند. می‌توان گفت که نزدیک‌ترین لایه به سخت‌افزار رایانه همان سیستم عامل است و پس از آن، سیستم مدیریت پایگاه‌داده‌ها در لایه دوم قرار دارد. سایر نرم‌افزارها چون برنامه‌های کاربردی، توابع ذخیره‌شده و ابزار تولید برنامه‌ها در لایه سوم قرار دارند.



شکل ۱-۲. نمای بیرونی نظام مدیریت پایگاه‌داده‌ها

۱-۳ ویژگی‌ها و امتیازهای پایگاه‌داده‌ها

۱-۳-۱ پرسش

به‌طورکلی داده‌پردازی دو هدف عمده دارد: نگهداری داده‌ها و پرسش یا تراکنش^۱ با امنیت، سرعت و کیفیت مورد انتظار.

مثال:

در یک سیستم پایگاهی بانکداری، نگهداری سوابق و موجودی حساب‌های بانکی به صورت امن و قابل اعتماد، هدف اول است. هدف دوم درخواست‌های کاربران است، مانند لیست عملیات یا برداشت پول. به‌طور معمول به درخواست‌هایی که فقط خواندنی هستند یعنی تغییری در داده‌ها ایجاد نمی‌کند، پرسش گفته می‌شود و درخواست‌هایی مانند واریز پول که موجودی را تغییر می‌دهند، تراکنش نام دارند. گاهی

تراکنش به هر دو معنی و گاهی پرسش به هر دو معنی به کار می‌رود. در ادامه همین فصل به تراکنش خواهیم پرداخت.

مهم‌ترین مزیت پایگاهداده‌ها این است که کاربران همواره می‌توانند تراکنش‌ها یا پرسش‌های جدید مطرح کنند (هرچند فقط تراکنش‌ها و پرسش‌های مجاز، پاسخ داده می‌شوند).

مثال:

در یک سیستم بانکداری کاربران نمی‌توانند از موجودی کاربران دیگر سؤال کنند. این پرسش غیرمجاز است.

مثال:

در یک سیستم آمار جمعیت، لازم است تا سال‌ها سؤال‌های تازه‌ای پرسیده شود. می‌توان از ترکیب‌های مختلف سن و جنسیت، وضع کار، تحصیل، تأهل، آدرس و دَهها مورد دیگر سؤال‌های بی‌شماری مطرح کرد.

نکته اصلی این است که در یک سیستم پایگاهداده، همواره می‌توان سؤال‌هایی مطرح کرد که از ابتدای طراحی سیستم مطرح نبوده‌اند؛ به عبارت دیگر لزومی ندارد از ابتدای طراحی سیستم پایگاهداده، به همه خروجی‌های آن پرداخت. همواره می‌توان خروجی‌های جدیدی از سیستم گرفت و تعداد و تنوع آن‌ها بستگی به تنوع داده‌های ذخیره‌شده دارد. این مفهوم را پرسش‌های بی‌پایان می‌نامیم. پرسش‌های بی‌پایان در برنامه‌نویسی عادی جایی ندارد و بزرگ‌ترین امتیاز پایگاهداده‌ها است.

مطالعه موردي

می‌خواهیم اطلاعات یک صندوق قرض‌الحسنه را به صورتی ساده سامان دهیم.
حل ۱: در یک زبان برنامه‌سازی سطح بالا (مانند C و پاسکال و جاوا) جدول‌هایی طراحی می‌کنیم که داده‌ها را نگهداری کنند و توابعی می‌نویسیم که اطلاعات را استخراج کنند.

در ابتدایی‌ترین حالت می‌توان دو جدول زیر را داشت:

customer (cno, vname, DOB^۱, Sex, tel, adres)

مشتری (آدرس، تلفن، جنسیت، تاریخ تولد، نام مشتری، شماره مشتری)

account(ano, cno, typ, sdate, atyp, balance)

1. DOB: Date Of Birth

حساب (موجودی، نوع عملیات، تاریخ عملیات، نوع حساب، شماره مشتری، شماره حساب)

در زبان‌های سطح بالا، عملیات بانکی حتماً باید از پیش مشخص باشند، توابعی مانند واریز، برداشت، موجودی،... و آنچه در عابر بانک‌ها مشاهده می‌شود. اگر بخواهیم آن‌ها را عوض کنیم باید برنامه جدیدی را برای سیستم بنویسیم.

حل ۲: در یک زبان پایگاه‌داده مانند اس - کیو - ال (SQL) می‌توان همین دو جدول را داشت و توابع را نیز آماده کرد. تفاوت اصلی این است که همواره می‌توان توابع جدید (پرسش‌ها یا تراکنش‌ها) نوشت و نیازی به تغییر سیستم نرم‌افزاری نیست.

۲-۳ تراکنش

برنامه کاربران که در محیط پایگاه‌داده اجرا می‌شود، تراکنش نام دارد. تفاوت اصلی یک تراکنش با یک برنامه معمولی در محیط غیربانکی، این است که تراکنش همواره به نظام مدیریت پایگاه‌داده (DBMS) تسلیم می‌شود و این نظام در اعمال هرگونه کترل و به‌تعویق انداختن و ساقط‌کردن آن آزادی عمل دارد. هدف اصلی از این‌گونه کترل‌ها، حذف و تعویق‌ها حفظ جامعیت و صحت پایگاه‌داده است. در پایگاه‌داده آنچه در درجه اول اهمیت قرار دارد، داده است نه برنامه. داده‌های پایگاه‌داده را مانا^۱ می‌نامند، زیرا برنامه‌ها می‌آیند و می‌روند، اما داده‌ها می‌مانند. مثلاً یک حساب بانکی زمانی باز و به‌طور متناوب به آن پول واریز و از آن برداشت می‌شود. یک برنامه که پولی را به حساب می‌ریزد یا برداشت می‌کند در صورت لزوم به تعویق می‌افتد یا رد می‌شود. مهم این است که موجودی حساب‌ها اشتباه نباشد. بدیهی است که صحت داده‌ها از صحت برنامه‌ها نشأت می‌گیرد.

از همان ابتدای پیدایش پایگاه‌داده، این سؤال ذهن متخصصین را به خود مشغول داشته است: چه کترل‌هایی لازم است روی برنامه‌ها اعمال شود تا صحت و جامعیت پایگاه‌داده تضمین شود؟ آقای جیم - گری در سال ۱۹۸۱ [J. Gray 1981] ثابت کرد که چهار کترل زیر لازم است روی تمامی تراکنش‌ها در پایگاه‌داده اعمال شود تا صحت و جامعیت داده‌ها تضمین شود (موسوم به خواص ACID)

۱. یکپارچگی^۱

این خاصیت به «همه یا هیچ» موسوم است. منظور این است که یا تمامی دستورالعمل‌های یک تراکنش باید اجرا شود و یا هیچ‌کدام از آن‌ها. به عنوان مثال تراکنشی را در نظر بگیرید که مبلغی پول را از حسابی به حساب دیگر (مثلاً در شهر یا کشور دیگری) منتقل می‌کند. این تراکنش شامل دو بخش است. بخش اول، پول را از حساب اول برداشت و بخش دوم همان پول را به حساب دوم واریز می‌کند. این دو بخش ممکن است روی دو کامپیوتر جداگانه اجرا شوند. فرض کنید بخش اول تراکنش اجرا شود، اما ناگهان ارتباط با ماشین دوم قطع شود و بخش دوم قابل انجام نباشد. بدیهی است که باید پول را به همان حساب اول بازگرداند تا صحت و جامعیت پایگاهداده حفظ شود. این عمل معادل این است که بگوییم هیچ دستورالعملی از تراکنش اجرا نشده است.

۲. هم‌خوانی^۲

این خاصیت می‌گوید که هر تراکنش باید تمامی قوانین جامعیت پایگاهداده را رعایت کند. علاوه‌بر این فرض می‌شود که تراکنش یک برنامه صحیح است. مثلاً در برنامه انتقال پول اگر مبلغ برداشت شده با مبلغ واریزشده به حساب دیگر برابر نباشد، تراکنش غلط است. غالباً چنین مواردی توسط نظام مدیریت پایگاهداده به‌طور خودکار قابل‌کترل نیست. بسیار اتفاق می‌افتد که کاربران برنامه‌های غلط اجرا می‌کنند و نتیجه غلط می‌گیرند. به عنوان مثال مبلغ اضافی انتقال می‌دهند. همواره نمی‌توان جلوی چنین اشتباهاتی را گرفت. بنابراین بخشی از خاصیت هم‌خوانی، قابل‌کترل نیست و باید فرض شود.

خاصیت هم‌خوانی به این صورت بیان می‌شود:

هر تراکنش اگر به‌نهایی اجرا شود پایگاهداده را از حالتی صحیح، به‌حالت صحیح دیگری منتقل می‌کند.

تراکنش ممکن است دو نوع پایان داشته باشد:

- پایان ناموفق که آن را «سقوط^۳» می‌نامند.
- پایان موفق که آن را «تثبیت^۴» می‌نامند.

1. Atomicity

2. Consistency

3. Abort

4. Commit

سؤال: آیا دو خاصیت یکپارچگی و هم‌خوانی برای صحت تراکنش کافی نیست؟
برطبق این دو خاصیت هر تراکنش یا به‌طور صحیح به انجام می‌رسد، یا ساقط می‌شود
و پایگاهداده به حالت اولیه بازمی‌گردد. آیا این کافی نیست؟ خیر!

نکته اینجا است که ممکن است تراکنش‌های هم‌روند روی یکدیگر اثر مخرب داشته باشند، یعنی دو یا چند تراکنش که از همدیگر بی‌اطلاع هستند، ناخودآگاه به‌طور ناهمانگ داده‌ها را تغییر دهند. همچنین اثر تراکنش‌های ثبیت‌شده به‌طور تصادفی (مثلًا خرابی دیسک) از بین برود. این دو پدیده دو خاصیت دیگر از خواص چهارگانه ACID را شامل می‌شود که در زیر می‌آید.

۳. انزوا^۱

برطبق این خاصیت، اثر تراکنش‌های هم‌روند و موازی روی یکدیگر چنان است که گویا هر کدام در انزوا (کاملاً جدا از یکدیگر) انجام می‌شود. توجه به این نکته، حائز اهمیت است که در پایگاهداده تراکنش‌های هم‌روند وجود دارند ولی هم‌روندی آن‌ها کنترل می‌شود تا اثر مخرب روی هم نداشته باشند. این عمل توسط بخشی از نظام مدیریت پایگاهداده (DBMS) به نام واحد کنترل هم‌روندی^۲ انجام می‌شود.

۴. پایائی^۳

براساس این خاصیت، تراکنش‌هایی که به مرحله ثبیت برستند اثر آن‌ها ماندنی است و هرگز به‌طور تصادفی از بین نمی‌روند. مثلًا اگر مبلغی به حسابی واریز و تراکنش مربوطه انجام یافته اعلام شود حتی در صورت وقوع آتش‌سوزی در آن شعبه بانک، مشتری متضرر نخواهد شد، یعنی عمل واریز، قبل از اعلام انجام موفق، در جای دیگری ثبت شده است.

دو عمل یکپارچگی و پایائی توسط واحدی از نظام مدیریت پایگاهداده به نام واحد مدیریت ترمیم^۴ انجام می‌گیرد. مفهوم تراکنش و بخش‌های آن چون کنترل هم‌روندی و مدیریت ترمیم پیچیده و دراز دامن است و در درس پایگاهداده پیش‌رفته بررسی می‌شود.

1. Isolation

2. Concurrency Control Unit

3. Durability

4. Recovery Management Unit

۳-۳-۱ ذخیره و بازیابی بهینه داده‌ها

گفتیم که نظام مدیریت پایگاهداده‌ها دو وظیفه اساسی دارد: نگهداری داده‌ها و پرسش. در بخش گذشته پرسش را بررسی کردیم. در اینجا به نگهداری و مدیریت بهینه داده‌ها می‌پردازیم.

۱. امکان مدل‌سازی داده‌ها

در پایگاهداده می‌توان از ساختارهایی که از دیرباز با ذهن انسان سازگارتر است استفاده کرد و از مفاهیم دور از ذهن (مانند تعداد زیادی ۰ و ۱) که کامپیوتراهای رقمی با آن به راحتی کار می‌کنند، پرهیز کرد. مثلاً ذهن آدمی، با جدول، آشنایی نزدیکی دارد. به عنوان نمونه می‌توان پایگاهداده بانکداری را در قالب جداول به هم پیوسته مانند جداول مشتری و حساب و... مدل‌سازی کرد. کار کردن با این جداول برای کاربران بسیار راحت‌تر از کار کردن مستقیم با پرونده‌های رقمی است.

۲. کنترل متمرکز

در سیستم‌های مبتنی بر پرونده افراد گوناگون با پرونده‌های مختلف و بعضًا تکراری کار می‌کنند. در پایگاهداده همه این داده‌ها تحت کنترل نظام مدیریت پایگاهداده‌ها ذخیر و بازیابی می‌شوند و از ناهمماهنگی پرهیز می‌شود.

۳. اشتراك داده‌ها بین کاربران

نظام مدیریت پایگاهداده پرونده‌های مختلف را در اختیار می‌گیرد و بخش‌های مختلف آن را در دسترس کاربران مجاز قرار می‌دهد. در این راستا، کاربران از پرونده‌های مختلف به‌طور مشترک استفاده می‌کنند تا از تکرار و دوباره‌کاری‌های مخرب و زمان‌گیر جلوگیری می‌شود.

۴. شاخص گذاری

شاخص وسیله‌ای است که دسترسی به داده‌ها را سریع‌تر و آسان‌تر می‌سازد. مثلاً در یک شرکت یا اداره که پرونده‌های مختلف به صورت دستی در بایگانی‌های بزرگ نگهداری می‌شوند، می‌توان دفتر شاخص داشت و به راحتی محل نگهداری هر پرونده را پیدا کرد. از همین مفهوم توسط نظام مدیریت پایگاهداده برای دسترسی به داده‌ها استفاده می‌شود و شاخص گذاری نام دارد.

۵. کاهش افزونگی داده‌ها

افزونگی یعنی تکرار بی‌رویه. در پایگاهداده، تکرار داده‌ها اجتناب‌ناپذیر است.

مثلاً برای برقراری ارتباط بین مشتری و حساب مجبوریم در یکی از جداول، کلید دیگری را تکرار کنیم. از سوی دیگر، به جای کلید می‌توان همه ستون‌های یک جدول را در جدول دیگر تکرار کرد که تکراری بی‌رویه است. در فصل نرمال‌سازی جداول به افزونگی داده‌ها در مدل رابطه‌ای خواهیم پرداخت.

۶. دستیابی موازی و هم‌رونده‌ی داده‌ها و اطلاعات

کاربران پایگاهداده باید بتوانند هم‌زمان به داده‌های آن دسترسی داشته باشند. مثلاً خودپردازهای مختلف بانک‌ها باید همواره به کاربران خود خدمات بدeneند و در کار یکدیگر اخلال نکنند. نظام مدیریت پایگاهداده امکانات مناسبی برای دستیابی موازی و هم‌رونده‌ی داده‌ها و اطلاعات فراهم می‌آورد. در این راستا سرعت عملکرد سیستم به طور چشم‌گیری افزایش می‌یابد و صحبت داده‌ها و اطلاعات نیز کترل می‌شود.

۷. عدم وجود ناسازگاری

معمولأً نسخه‌های مختلفی از یک داده در بانک‌های اطلاعات ذخیره می‌شود. مثلاً موجودی یک حساب بانکی در جاهای مختلف ثبت می‌شود تا در صورت بروز مشکلاتی چون سرقت اطلاعات، زلزله و سونامی بتوان به اطلاعات صحیح دسترسی داشت. در این راستا باید سازگاری و هم‌خوانی نسخه‌های مختلف حفظ شود. مثلاً اگر شخصی مبلغی را به حسابی واریز می‌کند، باید این مبلغ در تمام نسخه‌های آن حساب که ممکن است در شهرهای مختلف باشند، به نحو مناسبی ثبت شود.

۸. امنیت و جامعیت

صحیح و کامل بودن داده‌ها و عملیات روی آن‌ها در پایگاهداده ابعاد گوناگونی دارد که در اینجا به دو مورد از آن‌ها اشاره می‌کنیم:

الف) امنیت^۱، به معنای محافظت داده‌ها در برابر خطرات طبیعی از قبیل آتش‌سوزی و نیز جلوگیری از دستیابی غیرمجاز به آن‌ها است. تأمین امنیت داده‌ها بسیار مشکل و امنیت صدد رصد به‌ویژه در بانک‌های اطلاعات نامت مرکز، تقریباً غیرممکن است. راه‌های مختلفی برای جلوگیری از دستیابی غیرمجاز به داده‌ها وجود دارد که متداول‌ترین آن‌ها استفاده از رمز و تغییر اقلام داده با الگوریتم‌های خاص (مثلاً تبدیل موجودی حساب بانکی افراد به مقداری نامشخص و محاسبه موجودی واقعی در

موقع لزوم) است. با این همه همواره ممکن است کسانی پیدا شوند که این رمزها و قفل‌ها را بگشایند.

ب) جامعیت^۱ به معنای صحت داده‌ها و پردازش‌ها و پیروی از مقررات سیستم است. مثلاً موجودی واقعی حساب‌های بانکی نباید منفی باشد و یا شخص نتواند بیش از موجودی خود از حسابش برداشت کند. باید توجه کرد که در خلال انجام یک تراکنش، ممکن است جامعیت زیر سؤال رود! نوعی از جامعیت که به هم‌خوانی موسوم است یعنی اقلام داده در کل سیستم با هم در تضاد نباشند. مثلاً در نسخه‌ای از پرونده بانک، موجودی شخص با نسخه دیگر پرونده متفاوت نباشد.

۹. امکان ترمیم داده‌ها و اطلاعات

در مواردی ممکن است برای سیستم اتفاقی بیافتد یا عملیاتی ناتمام بماند. مثلاً در انتقال مبلغی از حساب به حساب دیگری در شهری دور، ممکن است بخش دوم قابل انجام نباشد (مثل خرابی شبکه). در این صورت لازم است پایگاهداده به حالت صحیح اولیه برگردد. این عمل، ترمیم نام دارد. نظام مدیریت پایگاهداده همواره پس از وقوع خطا، سیستم را به حالتی جامع برمی‌گرداند.

۱۰. تأمین استقلال داده‌ای^۲

در برنامه‌سازی معمولی با زبان‌های روالی، معمولاً تغییر پرونده‌ها باعث تغییر برنامه‌ها می‌شود. مثلاً اگر یک فیلد به پرونده دانشجو اضافه شود لازم است برنامه‌های مربوطه تغییر کنند، تا بتوانند این پرونده را بخوانند. این واقعیت بسیار تلخ است، زیرا برنامه‌های کامپیوتری عمر طولانی دارند و برنامه‌سازان تغییر می‌کنند و پیداکردن آن‌ها سخت است و تغییر برنامه‌های کامپیوتری دیگران از آن هم سخت‌تر. نظام مدیریت پایگاهداده که نرم‌افزار قدرتمندی است و از لایه‌های مختلفی تشکیل شده، می‌تواند از پس این مشکلات برآید. به عبارت دیگر در پایگاهداده رابطه مستقیم بین برنامه‌های کاربران و پرونده‌ها وجود ندارد، و این دو، تا حد زیادی از یکدیگر مستقل هستند.

۱۱. بهینه‌سازی پرسش^۳

پرسش، که به کاربران پایگاهداده امکان استخراج اطلاعات گوناگون و بسیاری

1. Integrity

2. Data Independence

3. Query Optimization

می‌دهد، از بزرگ‌ترین امتیازات این سیستم‌های مبتنی بر پرونده این امتیاز وجود ندارد، یعنی برای استخراج هر اطلاعاتی باید برنامه‌ای نوشت. در پایگاهداده این مهم به سادگی با استفاده از زبان‌های بیانی انجام می‌شود. از سوی دیگر نظام مدیریت پایگاهداده، برنامه‌های بیانی کاربران را، که ممکن است حرفه‌ای نباشند، بهینه می‌کند. مثلاً پرسشی که توسط کاربری نوشته شده و چند دقیقه طول می‌کشد، ممکن است به پرسشی معادل (با همان خروجی) تبدیل شود که در چند ثانیه انجام می‌شود.

۱۲. کاهش هزینه‌ها

پایگاهداده با طراحی و پیاده‌سازی بهینه و به کارگیری روش‌ها و الگوریتم‌های کارا می‌تواند هزینه نگهداری و حفاظت داده‌ها و استخراج اطلاعات را به‌ویژه در درازمدت کاهش دهد.

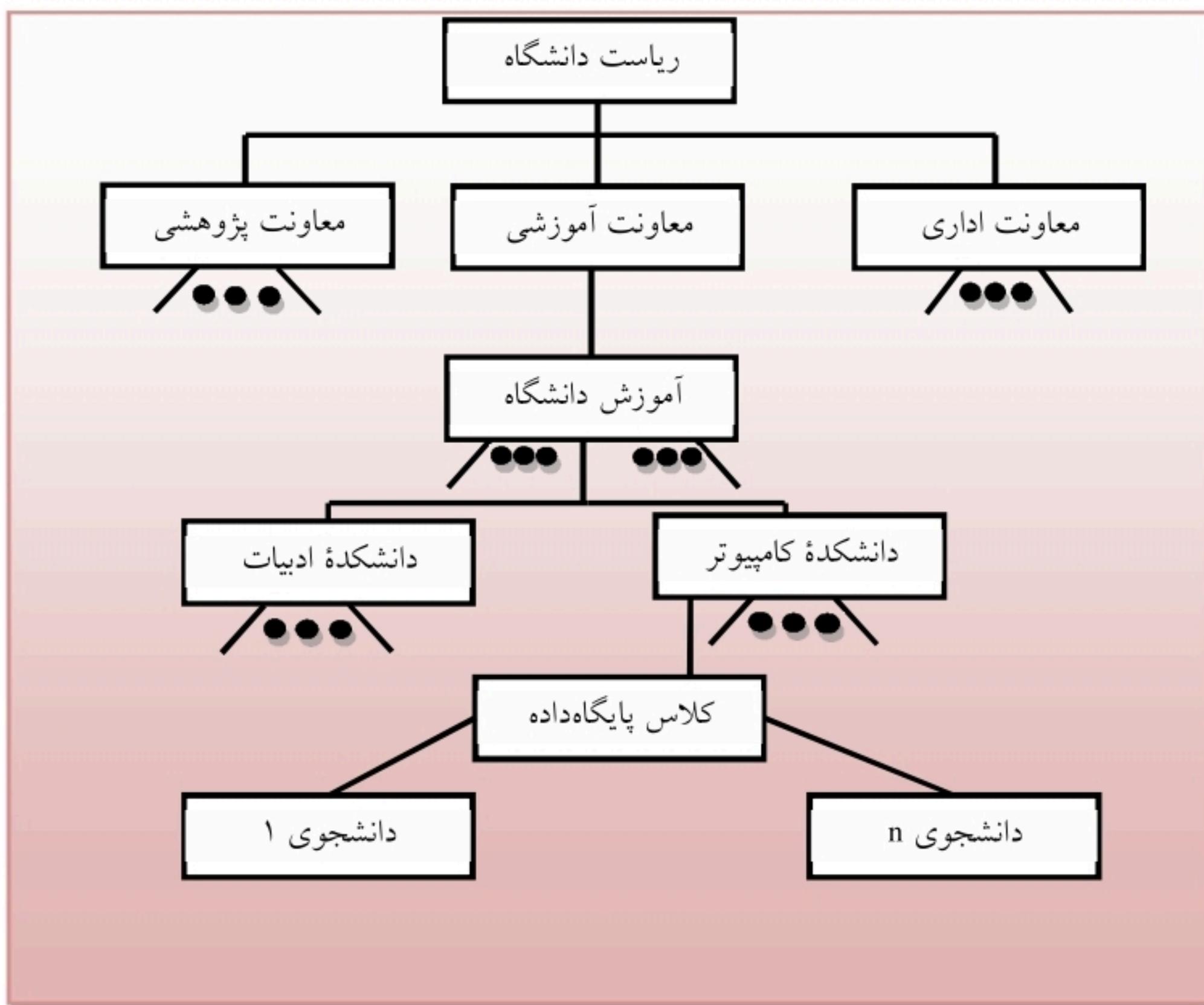
۱۳. سایر موارد

پایگاهداده امتیازهای دیگری هم دارد که فهم آن‌ها نیازمند شناخت مفاهیم آن است. مثلاً پایگاهداده نامتمرکز (توزیعی) که نیاز امروز جوامع بشری و در همه‌جا مورد استفاده است، امتیازهای گسترده‌ای دارد. خوانندگان ارجمند در حین مطالعه این کتاب و نیز در درس پایگاهداده پیش‌رفته، به تدریج به این امتیازها پی‌خواهند برد.

۴-۱ مدل‌های پایگاهداده

۱-۱ مدل‌های ابتدایی (پیش‌رابطه‌ای)

از دهه ۱۹۶۰ که مفهوم پایگاهداده‌ها ذهن پژوهشگران را به خود معطوف کرد، این سؤال مطرح بود که پایگاهداده چیست؟ به عبارت دیگر پایگاهداده‌ها را، با کدام ساختمان داده‌ها می‌توان بهتر بیان کرد؟ اولین ساختمان داده‌هایی که پیشنهاد شد «درخت» بود که مبنای مدل سلسله‌مراتبی^۱ قرار گرفت. مثلاً پایگاهداده یک دانشگاه به صورت درختی تصور می‌شد که ریشه آن ریاست دانشگاه و فرزندان ریشه معاونت‌ها و فرزندان آن‌ها سایر بخش‌ها را تشکیل می‌دادند (شکل ۳-۱). این مدل پیگیری شد و به پیاده‌سازی انجامید و به مرحله تجاری نیز رسید. عمده‌ترین کاستی‌های این مدل دو چیز بود:



شکل ۱-۳. نمایی از پایگاهداده دانشگاه در مدل سلسله مراتبی

۱. همه کاربردها به صورت درخت نبودند. مثلاً یک پایگاهداده بانکداری با این مدل سازگاری ندارد، زیرا مشتریان یک بانک مستقل از یکدیگر هستند و خاصیت سلسله مراتبی ندارند.
 ۲. ارتباط اجزاء در درخت فقط پدر-فرزندی است. مثلاً اگر در این شکل دانشجوی ۱ با آموزش دانشگاه ارتباطی داشته باشد (که دارد) نمی‌توان این ارتباط را مستقیماً بیان کرد، یعنی باید از شاخه‌های درخت بالا رفت تا به آموزش دانشگاه وصل شد.
- برای رفع کاستی‌های مدل سلسله مراتبی، مدل شبکه‌ای^۱ پیشنهاد شد که زیرساخت آن گراف است. در این مدل تصور می‌شود که هر پایگاهداده یک گراف است، که هر جزء (گره) آن می‌تواند با جزء دیگر آن ارتباط داشته باشد. آبروی مدل سلسله مراتبی درست اما چشم آن کور شد. پایگاهداده دانشگاه را در نظر بگیریم که همه اجزای آن با هم در ارتباط هستند (شکل ۱-۴). پیچیدگی این پایگاهداده آنقدر زیاد

است که باید فاتحه امنیت و جامعیت و... را خواند. مثلاً یک دانشجو در این پایگاه‌داده می‌تواند مستقیماً با هر استاد یا مسئول یا واحد یا اداره‌ای ارتباط داشته باشد. چگونه می‌توان این ارتباط‌ها را کنترل کرد!

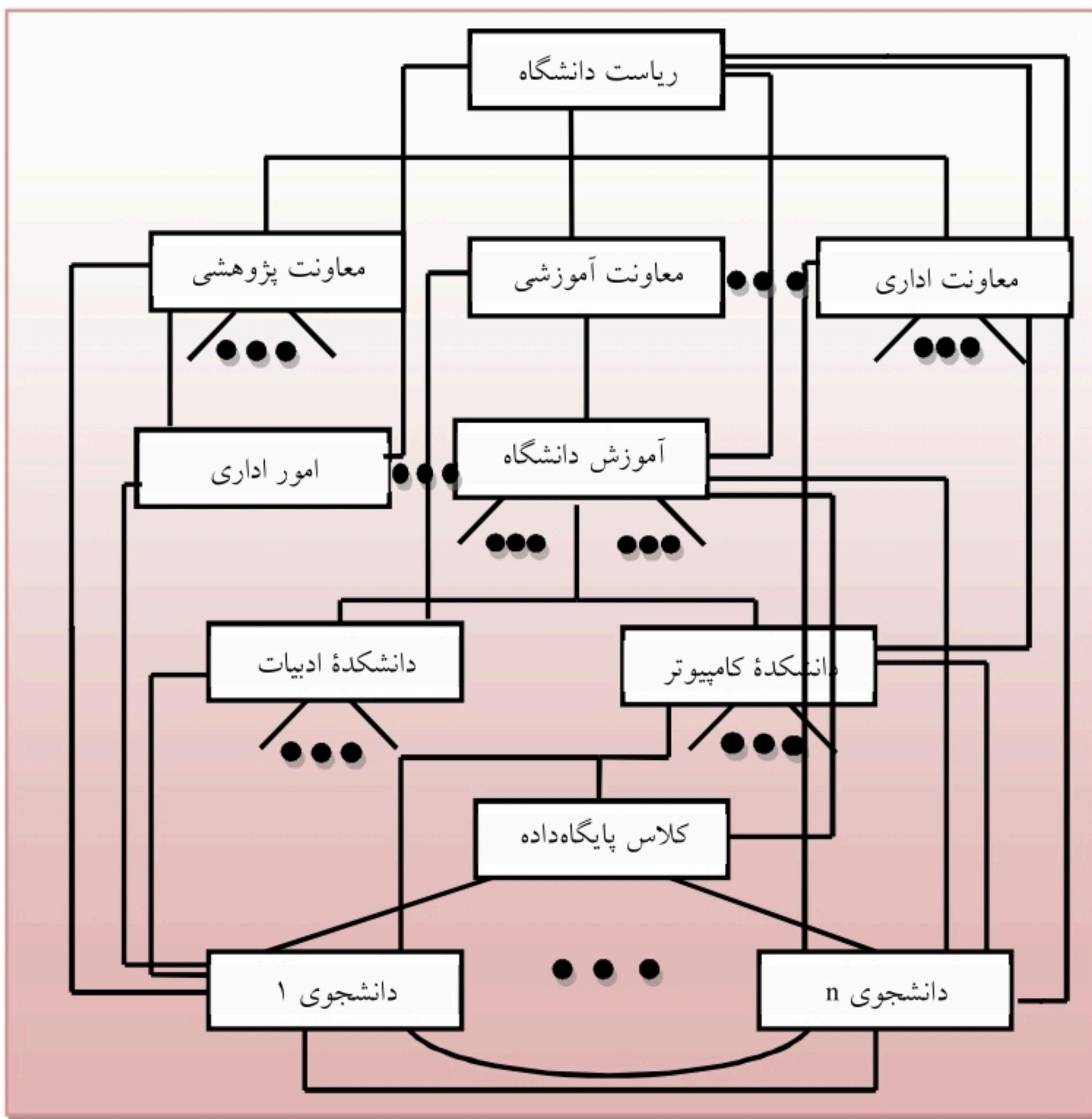
ساختار داده مدل شبکه‌ای که نوعی گراف جهت‌دار به نام شبکه است، در گروه^۱ وابسته به ANSI پیشنهاد شد. اولین نظام مدیریت پایگاه‌داده شبکه‌ای که توسط همین گروه طراحی شده بود، در سال ۱۹۷۱ مورد تأیید ANSI قرار گرفت. این نظام مدیریت پایگاه‌داده شبکه‌ای را، (IDMS)^۲ نام‌گذاری کردند و نام دیگر آن (CODASYL)^۳ بود. مدل شبکه‌ای نیز گاهی به همین نام خوانده می‌شود. این مدل نتوانست جای عمدہ‌ای در پایگاه‌داده‌ها باز کند.

۴-۲ مدل رابطه‌ای^۴

در سال ۱۹۷۰، آقای E.F.Codd در پایگاه‌داده‌ها به وجود آورد. [E.F.Codd 1970] او با استفاده از دو مفهوم نسبتاً آسان مجموعه و جدول، مدل رابطه‌ای را ارائه داد. هرچند این مدل قدمتی چندین ساله دارد، اما هنوز هم در دنیای پایگاه‌داده‌ها مطرح است. شاید بالاترین سطح مقبولیت در تاریخ مهندسی نرم‌افزار کامپیوتر، به این مدل و محصولات متنوعی که بر مبنای آن ساخته شده تعلق داشته باشد. موقعيت این مدل تا حدی است که امروزه اکثر کاربران، مفهوم پایگاه‌داده را در قالب آن می‌شناسند.

در مدل رابطه‌ای، داده‌ها به صورت رکوردهای مرتب سازماندهی می‌شوند و پایگاه‌داده به صورت مجموعه‌ای از رابطه‌های به هم پیوسته طراحی می‌شود. عمدترين دليل موقعيت بي نظير مدل رابطه‌ای سادگي آن است. رابطه که به صورت جدول پياده‌سازی می‌شود، مفهومی است که همگان با آن آشنا هستند و با آن سروکار داشته‌اند. دليل دیگر موقعيت مدل رابطه‌ای، پشتوانه نظری بسيار قوي آن است. به دليل سرمایه‌گذاري مادي و معنوی که از نياز مؤسسات مختلف به پایگاه‌داده نشأت می‌گيرد، تعداد بسيار زيادي کتاب و مقاله در اين مقوله به رشته تحرير درآمده و تمامی زوایاي کار روشن و به صورت نظری اثبات شده است.

1. Database Task Group
2. Integrated Data Management System
3. Conference On Data System Languege
4. Relational Model



شکل ۴-۱. نمایی از پایگاه‌داده دانشگاه در مدل شبکه‌ای

در مدل رابطه‌ای هر یک از پدیده‌ها، به‌طور مستقل یک رابطه (جدول) هستند، که ستون‌های آن جدول را صفت‌های آن پدیده و سطرهای آن را داده‌های مربوطه تشکیل می‌دهند. در شکل ۱-۵ جداول پایگاه‌داده دانشگاه دیده می‌شود. در این شکل، جدول دانشجو دارای صفت‌هایی چون کلید (شماره) و نام دانشجو و غیره و هر سطر آن داده‌های یک دانشجو است.

ارتباط بین جدول‌ها به دو صورت قابل بیان است:

۱. کلید خارجی با تکرار کلید جدولی در جدول دیگر:

مثلاً با تکرار کلید گروه آموزشی در جدول استاد نشان می‌دهیم که هر استاد متعلق به کدام گروه آموزشی است.

۲. تعریف جدول ارتباط: ارتباط‌های پیچیده را به این صورت نمایش می‌دهیم. در شکل ۱-۵ جدول «ارائه دروس» نوعی ارتباط بین درس و استاد و ترم تحصیلی برای ارائه دروس است که زمان و مکان و سایر مشخصات هر درس را نیز دربر می‌گیرد.

Student(<u>Skey</u> , name, city, average, Gkey)	جدول دانشجو ←
Professor (<u>Pkey</u> , name, degree, sex, age, office, <u>Gkey</u>)	جدول استاد ←
Group (<u>Gkey</u> ,name, manager,Clgkey)	جدول گروه آموزشی ←
Coledge (<u>Clgkey</u> , name, city, manager, adres)	جدول دانشکده ←
Course (<u>Ckey</u> , name, unit, Gkey)	جدول درس ←
Section(<u>Seckey</u> , <u>Ckey</u> , <u>Pkey</u> , term, time, place)	جدول ارائه دروس ←
Enroll(<u>Seckey</u> , <u>Ckey</u> , <u>Skey</u> , <u>term</u> , score)	جدول ثبت نام ←

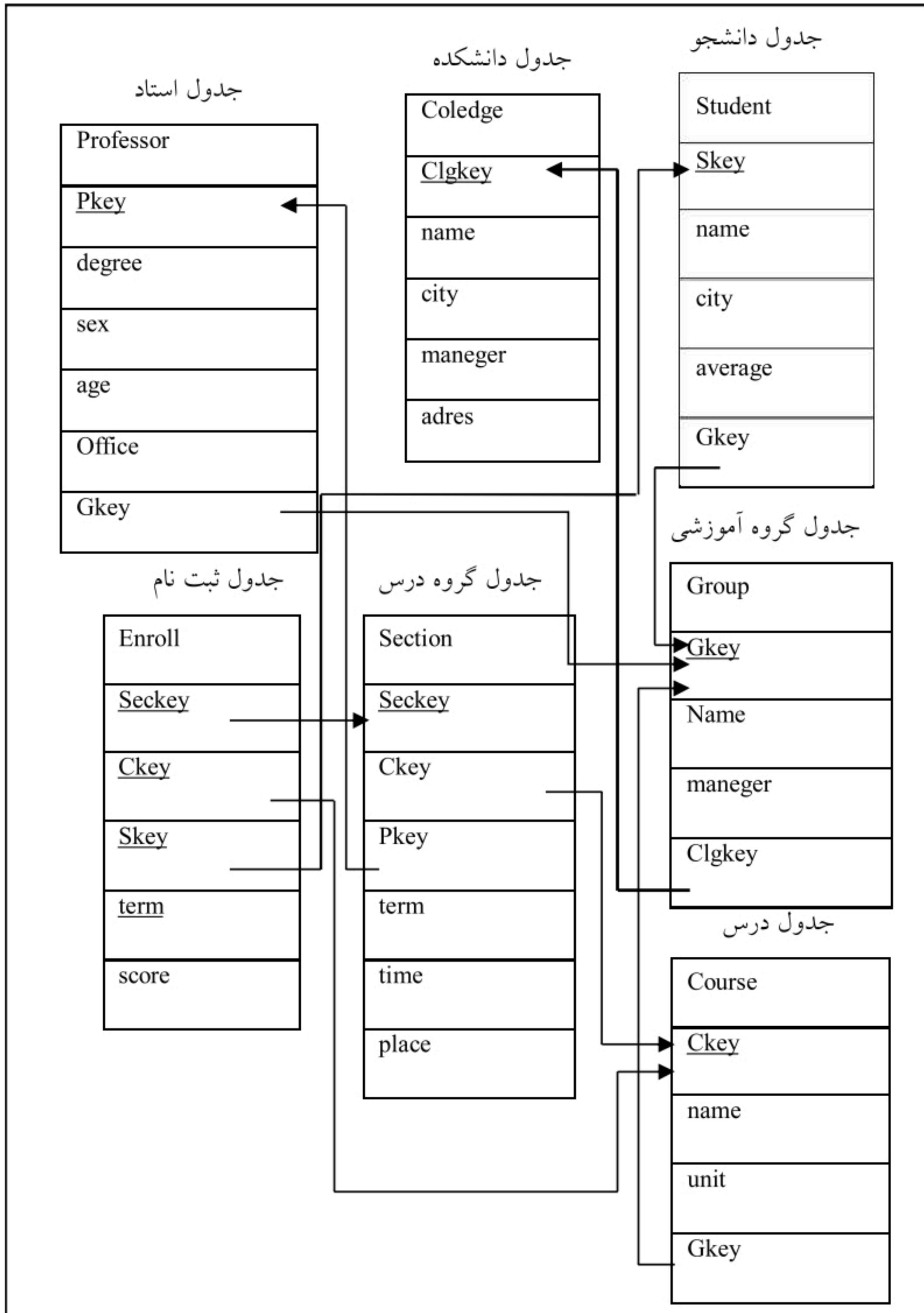
شکل ۱-۵. جداول پایگاهداده دانشگاه در مدل رابطه‌ای

هر جدول کلید مستقلی دارد که برای هر عضو آن مقدار ویژه‌ای را دربر می‌گیرد (تکراری نیست). مقایسه این پایگاهداده با معادلهای آن در مدل‌های سلسله‌مراتبی و شبکه‌ای، سادگی و فراگیری این مدل را نمایش می‌دهد. تأکید می‌کنیم که ارتباط بین جدول‌ها با تکرار کلید جدولی در جدولی دیگر، برقرار می‌شود. مثلاً با تکرار کلید جدول گروه در هر سطر جدول دانشجو، نشان می‌دهیم که آن دانشجو به کدام گروه آموزشی تعلق دارد.

روش دیگری برای نمایش پایگاهداده در این مدل، نمودار شمای پایگاهداده رابطه‌ای است. در این نمودار هر جدول، در یک مستطیل جداگانه می‌آید. نام جدول در بالای آن به صورت پرنگ و نام ستون‌های آن در ادامه می‌آید و زیر کلید خط می‌کشند. کلیدهای خارجی هر جدول، توسط پیکان‌هایی به جدول اصلی مرتبط می‌شود.

شکل ۱-۶ نمودار شمای پایگاهداده دانشگاه را در مدل رابطه‌ای نمایش می‌دهد. نمودار شمای پایگاهداده رابطه‌ای، روشهای استاندارد و نسبتاً ساده است که تمامی جداول پایگاهداده و ارتباط بین آنها را نمایش می‌دهد.

مدل رابطه‌ای با همه عظمتش، برای کاربردهای تجاری و سنتی پایگاهداده ساخته شده و پاسخگوی بسیاری از نیازهای زمان ما نیست. مثلاً پایگاهداده حاوی صوت، تصویر، متن^۱ و... را نمی‌توان در قالب رکورد نمایش داد.



شکل ۱-۶. نمودار شمای پایگاه‌داده دانشگاه در مدل رابطه‌ای

۳-۴ مدل‌های جدید (فرا رابطه‌ای)

به عنوان یک نمونه، به پایگاهداده یک شبکه اجتماعی (مانند فیسبوک، تلگرام یا ...) نگاه کنید که در آن تعداد بسیار زیادی کاربر در قالب میلیون‌ها گروه، سازماندهی می‌شوند و به صورت لحظه‌ای، حجم زیادی از داده‌های نوشتاری، صوتی و تصویری مرتبط با یکدیگر را به صورت لینک شده رُدوبُدل می‌کنند و به اشتراک می‌گذارند. آیا ساماندهی چنین پایگاهداده‌ای در مدل رابطه‌ای امکان‌پذیر است؟ بعید به نظر می‌رسد. کوشش‌های بسیار زیادی برای پاسخگویی به این نیازها به عمل آمده و مدل‌های متنوعی پیشنهاد و پیاده‌سازی شده‌است. از مهم‌ترین این مدل‌ها می‌توان مدل شئ‌گرا^۱، مدل تابعی^۲، مدل منطقی^۳ و مدل استنتاجی^۴ و مدل XML را نام برد. همچنین کوشش‌های موفقی در پیشبرد مدل رابطه‌ای در راستای نیازهای زمان صورت گرفته است. مدل‌های شئ‌گرا و شئ - رابطه‌ای گوی سبقت را ربوده و در پاسخگویی به نیازهای جدید حرف اول را می‌زنند. در فصل‌های آینده به مدل شئ - رابطه‌ای خواهیم پرداخت. خوانندگان گرامی می‌توانند برای آگاهی از سایر مدل‌ها به مراجع [M.Piattini,O.Diaz, 2000] و [C.Delobel, 1995] و [J.F.Fortier, 1997]

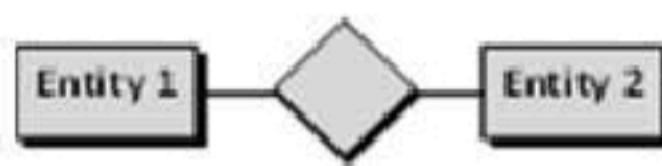
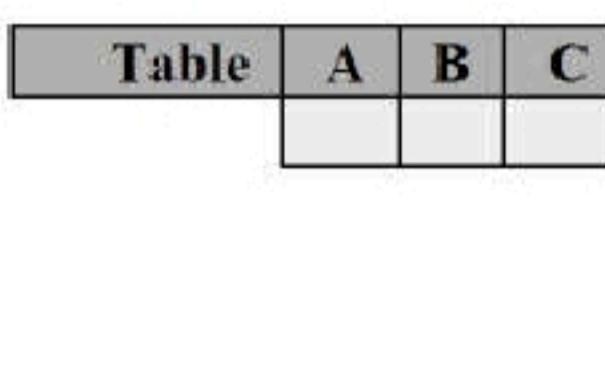
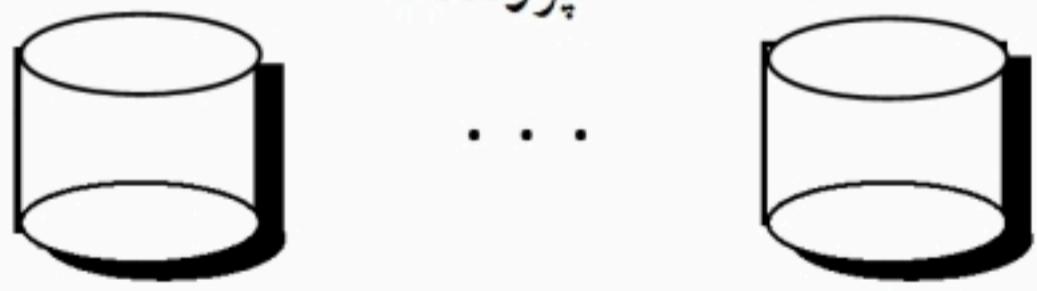
۱-۵ معماری پایگاهداده‌ها

۱-۵-۱ معماری چندلایه‌ای

از آنچه تاکنون بیان شد به خوبی دریافت می‌شود که پایگاهداده‌ها و نظام مدیریت آن بسیار پیچیده و لایه‌لایه است. از همان سال‌های آغازین این سؤال مطرح بوده است که چگونه می‌توان این لایه‌ها و بخش‌های متفاوت را از کاربران تا حافظه کامپیوتر در یک معماری کلی به هم متصل کرد. محافل مختلف علمی و دست‌اندرکاران فنی این رشته، پیشنهادهای گوناگونی ارائه دادند تا اینکه مؤسسه استانداردسازی ANSI-SPARC در سال ۱۹۷۵ با بهره‌گیری از یافته‌های دانشمندان و متخصصان، یک معماری سه‌سطحی^۵ را اعلام کرد.

1. Object-Oriented
2. Functional
3. Logical
4. Deductive
5. three-level architecture

این معماری با اکثر مدل‌های پایگاه‌داده هم‌خوانی داشت و از مدل خاصی پیروی نمی‌کرد.

دیدهای کاربران مختلف (views)		لایه خارجی
کل بانک بدون توجه به مدلی خاص	 <pre> class Stud attr s#: int +<PK> idx_stud() </pre>	لایه ادراکی عام
کل بانک در قالب مدل انتخابی	 <pre> class Stud attr S# method gpa() end </pre>	لایه ادراکی خاص
برای پروندهای بزرگ	شناخت‌ها	لایه داخلی
کل بانک روی رسانه‌ها	 پروندها	لایه فیزیکی

شکل ۱-۷. معماری پنج لایه‌ای پایگاه‌داده‌ها

با افزودن لایه‌های داخلی و فیزیکی، یک معماری کامل‌تر پنج لایه‌ای، به دست می‌آید که ما آن را بررسی می‌کنیم.

شکل ۱-۷ این معماری را نمایش می‌دهد. بالاترین لایه (لایه خارجی) دیدهای مختلف کاربران را دربر می‌گیرد. این لایه، تنها لایه‌ای است که به کاربران مربوط می‌شود. هر کاربر با بخشی از پایگاه‌داده سروکار دارد و فقط اجازه دستیابی به بخشی از آن، به او داده می‌شود. مثلاً در پایگاه‌داده بانکداری، کاربران (کارمندان) بخش ارز، نه با بخش‌هایی مثل وام‌ها سروکار دارند و نه باید بتوانند به آن اطلاعات دسترسی داشته باشند.

قانون پنهانسازی اطلاعات که می‌گوید «به هر کس به همان اندازه اطلاعات بده که نیاز دارد، نه بیشتر»، در اینجا صادق است. بنابراین در لایه خارجی، دیدهای مختلف کاربران مطرح است و هر کدام بخشی از بانک را می‌بینند.

دیدهای مختلف ممکن است با هم اشتراک داده داشته باشند. مثلاً ممکن است ستون‌های A و B و D از یک جدول در اختیار کاربر ۱ و ستون‌های A و C از آن در اختیار کاربر ۲ قرار گیرد (ستون A مشترک است). این مسئله چنان‌که در فصل‌های بعدی خواهیم دید، مشکل‌آفرین می‌شود. لازم به تأکید است که لایه خارجی تنها لایه‌ای است که کاربران با آن سروکار دارند. لایه‌های دیگر به مدیر و برنامه‌سازان پایگاهداده مربوط می‌شود. مدیران پایگاه داده‌های بزرگ، معمولاً برنامه‌سازان خاص با تخصص‌های مختلف را، برای لایه‌های مختلف در نظر می‌گیرند.

لایه ادراکی عام، فقط در مرحله طراحی معنایی پایگاهداده، مطرح می‌شود. پس از طراحی نهایی و کامل بانک و انتخاب یک مدل برای پیاده‌سازی آن، این لایه در فرایند طراحی منطقی پایگاهداده‌ها به لایه ادراکی خاص تبدیل می‌شود.

در واقع در یک پایگاهداده که در مرحله بهره‌برداری است لایه ادراکی عام، فقط در مستندات آن وجود دارد و آنچه پیاده‌سازی شده و مورد استفاده قرار می‌گیرد، لایه ادراکی خاص است. چون اکثر افراد که کاربران پایگاهداده هستند با طراحی آن سروکار ندارند، یک لایه ادراکی بیشتر نمی‌شناسند، که همان لایه ادراکی خاص است. معروف‌ترین مدل‌هایی که در تصویر ادراکی عام مقبولیت یافته‌اند مدل‌های EER و UML هستند که بررسی خواهند شد. موفق‌ترین مدل‌های پایگاهداده که در پیاده‌سازی مورد استفاده قرار گرفته‌اند، مدل سنتی رابطه‌ای و مدل‌های جدید شئ‌گرا و شئ‌رابطه‌ای می‌باشند، که در فصل‌های بعدی بررسی خواهند شد.

لایه‌های داخلی و فیزیکی به‌نحوه ذخیره‌سازی داده‌ها و دادگان^۱ روی رسانه‌ها و نیز حافظه اصلی و حافظه catche مربوط می‌شود. فهم بخش‌های مختلف این دو لایه، مخصوصاً مفاهیم پرونده^۲ و شاخص^۳ که عمدت‌ترین عامل سرعتِ عمل در پایگاهداده است، در فصل آخر به اختصار آمده است.

1. Meta Data

2. File

3. Index

گذر از یک لایه به لایه بعدی از طریق تبدیل^۱ و توسط نظام مدیریت پایگاهداده (DBMS) به‌طور خودکار انجام می‌شود. وقتی کاربر دستوری صادر می‌کند، این دستور به دید ویژه او مربوط می‌شود، که معمولاً مجازی است (یعنی عیناً وجود خارجی ندارد). این دستور توسط واحد تبدیل لایه خارجی به ادراکی، به دستوراتی تبدیل می‌شود، که در آن‌ها از ساختارهای واقعی (مثل جداول اصلی بانک) استفاده شده است. خروجی این مرحله به عنوان ورودی مرحله بعد تلقی می‌شود. در آن مرحله، واحد تبدیل لایه ادراکی به داخلی و فیزیکی، این دستورات را که در مدل خاصی صادر شده‌اند، با کمک سیستم‌عامل به دستورات اجرایی تبدیل می‌کند که نتیجه آن، دستیابی به رسانه‌ها و اطلاعات مورد نظر است.

۱-۵-۲ اجزای لایه‌ها

در این پنج سطح اجزای مهمی وجود دارد که فهم هریک از آن‌ها برای شناخت پایگاهداده‌ها و محیط پایگاهی لازم است. مهم‌ترین این اجزاء در زیر آمده‌اند. در لایه خارجی مهم‌ترین مفهوم، دیدهای کاربران است. هر دسته از کاربران (مانند دانشجویان یا اساتید) به بخش خاصی از پایگاهداده و نیز عملیات ویژه‌ای روی آن‌ها محدود می‌شوند. مثلاً دانشجو فقط به درس‌های خودش دسترسی دارد و فقط می‌تواند نمره‌اش را مشاهده کند، درحالی که هر استاد به همه دروس خود و همه دانشجویان آن‌ها دسترسی دارد و می‌تواند نمره‌ها را تغییر دهد. مجموعه داده‌ها و عملیات ویژه هر نوع کاربر را، دید آن کاربر می‌نامیم؛ بنابراین لایه خارجی هر پایگاهداده، مجموعه دیدهای کاربران آن است. همان‌گونه که بیان شد دیدهای کاملاً از هم مجزا نیستند. در مثال استاد و دانشجو، معمولاً دیدهای دانشجویان بخشی از دیدهای اساتید هستند. اگر استاد نمره دانشجویی را تغییر دهد، دانشجو بلافاصله می‌تواند آن را مشاهده کند.

دید، حالت انتزاعی دارد، یعنی مثلاً تغییری که استاد در نمره دانشجو می‌دهد، توسط عملیات تبدیل، به لایه ادراکی و نهایتاً به لایه‌های داخلی و فیزیکی منتقل می‌شود و از روی آن‌ها دید دانشجو اصلاح می‌شود. کاربران از ابزار مختلفی به نام واسط کاربر که ممکن است فرم‌ها و اسکال متفاوتی داشته باشند، بهره می‌برند.

لایه ادراکی مستقیماً توسط DBMS مدیریت می‌شود، در پایگاه‌داده‌های بزرگ، مدیر و برنامه‌ساز پایگاه‌داده مسئول این بخش هستند. یادآوری می‌شود که در لایه ادراکی کل پایگاه‌داده تعریف می‌شود. زبان‌های پایگاه‌داده (مانند SQL) امکانات لازم جهت تعریف و بارگذاری کل پایگاه‌داده را فراهم می‌کنند. همچنین تعریف دیدهای کاربران نیز توسط مدیر و برنامه‌ساز تصمیم‌گیری و اجرا می‌شود؛ و زبان‌های پایگاه‌داده‌ها این امکان را نیز فراهم می‌آورند (در معرفی SQL به آن خواهیم پرداخت). لایه‌های داخلی و فیزیکی، بیشتر در سطح سیستم‌عامل اجرا می‌شوند و نتایج به سطح DBMS تبدیل می‌شود.

عملیات تبدیل، بین لایه‌های مختلف پیچیدگی خاصی دارد. مشکل اصلی، برقراری ارتباط بین بخش‌های مختلف است که ممکن است در جاهای مختلف پخش باشند. به عنوان نمونه در یک سیستم پایگاه‌داده بانکداری، اگر در بانک مرکزی تصمیمی سراسری، گرفته می‌شود یا تغییری می‌دهند، در سطح کشور باید این تصمیم به صورت فنی تسریٰ پیدا کند. این تسریٰ باید خودکار باشد، یعنی مثلاً اگر نرخ بهره یکسان می‌شود، باید کامپیوترها و پایگاه‌داده‌های همه بانک‌ها آن را یکنواخت اجرا کنند.

همچنین اگر فیلد یا نام جدیدی اضافه می‌شود، باید همگان از آن آگاهی یابند تا در پرسش‌ها و برنامه‌های خود، به کار برنند. باز اصلی این بخش را دادگان بر دوش می‌کشد که به دو بخش لغتنامه داده‌ها و کاتالوگ سیستم تقسیم می‌شود:

الف) لغتنامه داده‌ها^۱

در یک سیستم پایگاه‌داده‌ها، اسمی زیادی مورد استفاده می‌گیرند. از آنجایی که افراد زیاد و متفاوتی در یک مجموعه پایگاه‌داده درگیر هستند و در مواردی چون پایگاه‌داده‌ها، به صورت نامتمرکز، در محل‌های مختلف جغرافیایی پراکنده هستند، مرجعی برای ایجاد یکنواختی و هماهنگی در نام داده‌ها و معنای آن‌ها ضروری است. این مرجع، لغتنامه داده‌ها نام دارد. لغتنامه داده‌ها شبیه لغتنامه معمولی، تمامی اسمی استفاده شده در سیستم و معنای آن‌ها را دربر می‌گیرد. اگر افراد مجاز بخواهند از وجود یا معنای کلمات (مانند نام جداول) باخبر شوند، می‌توانند به لغتنامه داده‌ها مراجعه کنند. از مرحله طراحی پایگاه‌داده تا پیاده‌سازی آن، هرگاه طراح برای مفهومی،

1. Data Dictionary

نامی برگزیند، باید آن را در لغتنامه داده‌ها، همراه با معنا و فرمت آن و... وارد کند. این اسمی، شامل تمامی نام‌های استفاده شده در سیستم می‌باشد (مثل نام جدول یا شیء یا صفت). در پایگاهداده‌های مدرن، نرمافزارهای ویژه‌ای برای کار با لغتنامه‌ها وجود دارد. توسط این نرمافزارها می‌توان اسمی را در لغتنامه وارد یا در آن جستجو کرد. این نرمافزارها از اشتباهاتی، شبیه واردکردن یک نام با دو معنای مختلف^۱ و یا دو نام برای یک مفهوم^۲، جلوگیری می‌کنند. مثلاً ممکن است در یک سیستم دانشجویی، شخصی کلمه stud# no و دیگری کلمه stud را، برای شماره دانشجویی انتخاب کند. نرمافزار لغتنامه داده‌ها، از بروز چنین اشتباهاتی جلوگیری می‌کند.

ب) کاتالوگ سیستم^۳

علاوه بر اسمی داده‌ها و مشخصات آن‌ها، اطلاعات دیگری نیز در مورد پایگاهداده باید نگهداری شود. مثلاً اطلاعات مربوط به حق دستیابی افراد به داده‌های مختلف، تاریخ ایجاد و بهروز درآوردن داده‌ها، تعداد نسخه‌های هر پرونده و ترتیب زمانی آن‌ها، ایجادکننده پرونده‌ها، اندازه هر جدول یا شیء و غیره. این‌گونه اطلاعات در کاتالوگ سیستم نگهداری می‌شود. کاتالوگ سیستم، اطلاعات لغتنامه داده‌ها را نیز دربر می‌گیرد. به عبارت دیگر لغتنامه داده‌ها، زیرمجموعه کاتالوگ سیستم است و به دلیل کاربرد ویژه‌ای که دارد، مجزا شده و برای کار با آن نرمافزار طراحی شده است. سیستم پایگاهداده به‌طور خودکار و به کمک کاربران، اطلاعات موجود در کاتالوگ سیستم را همواره به‌روز نگه می‌دارد. به این اطلاعات، دادگان می‌گویند، که به معنی داده در مورد داده است.

یکی از مفاهیم اساسی که در معماری پایگاهداده‌ها رعایت می‌شود، استقلال داده‌ای است.

منظور از استقلال داده‌ای، مستقل بودن ذخیره‌سازی داده‌ها از کاربرد آن‌ها است. هریک از مدل‌های پایگاهداده از نوعی تجرید یا انتزاع بهره می‌برند. مثلاً مدل رابطه‌ای، از تجریدی به نام رابطه (جدول) استفاده می‌کند. داده‌ها هرچه باشند در قالب چند جدول ریخته می‌شوند. استفاده کاربر از این جداول هیچ ربطی به نحوه ذخیره‌سازی داده‌ها ندارد. در واقع نحوه ذخیره‌سازی داده‌ها روی رسانه‌ها، از دید کاربران مخفی

1. Homonym

2. Synonym

3. System Catalog

می‌ماند. اگر تغییری در ذخیره‌سازی داده‌ها وارد شود (مثلاً دیسک عوض شود)، برنامه‌های کاربردی هیچ تغییری نمی‌کنند. به این نوع استقلال، استقلال فیزیکی داده‌ها^۱ می‌گویند. جالب است که تغییر لایه ادراکی پایگاهداده هم از دید کاربران و برنامه‌های آن‌ها مخفی می‌ماند و به اصطلاح عطف به مسبق نمی‌شود. مثلاً اگر جدولی چهار ستون داشته و برنامه‌هایی روی آن ستون‌ها نوشته شده باشد، در صورتی که ستون پنجمی به آن اضافه شود، برنامه‌های سابق نیاز به دستکاری ندارند و به همان شکل قابل اجرا هستند. کافی است دیدهای جدید با استفاده از ستون جدید، در برنامه‌ها مورد استفاده قرار گیرد. طبیعی است که دیدهای سابق، از وابستگی به دیدهای جدید به دور بمانند. این نوع استقلال را، استقلال منطقی داده‌ها^۲ می‌نامند.

۱-۶ طراحی پایگاهداده‌ها

بخش عمده طراحی پایگاهداده‌ها، در واقع طراحی شمای پایگاهداده^۳ است. مجموعه ساختارهای طراحی شده در یک پایگاهداده، بدون توجه به داده‌هایی که در آن‌ها قرار می‌گیرند، شمای آن پایگاهداده نام دارد. مثلاً در مدل رابطه‌ای، که با رابطه (جدول) سروکار دارد، شمای یک پایگاهداده را جداول و ستون‌های آن جداول و ارتباط بین آن‌ها تشکیل می‌دهند. نوع داده هر ستون (مثلاً رشته، عدد صحیح,...) به شمای پایگاهداده مربوط می‌شود، ولی تعداد سطرهای موجود در جدول، ربطی به شمای پایگاهداده ندارد. در مدل شئ‌گرا، کلاس‌های تعریف شده برای یک بانک، بدنۀ کلاس‌ها^۴، که صفات و مشخصات اشیاء را تعریف می‌کنند و نیز مجموعه تمام پیام‌هایی که ممکن است بین اشیاء رد و بدل شود، شمای آن پایگاهداده را تشکیل می‌دهد. تعداد اشیاء آن پایگاهداده ربطی به شمای پایگاهداده ندارد، چون هر لحظه ممکن است کم یا زیاد شود. به عبارت دیگر، شمای چنین بانکی، انواع^۵ مختلف اشیائی است که در آن می‌توانند وجود داشته باشند، نه خود آن اشیاء.

1. Physical Data Independence
2. Logical Data Independence
3. Database Schema
4. Class Body
5. Types

۱-۶-۱ مراحل طراحی پایگاهداده‌ها

طراحی پایگاهداده، همانند سایر سیستم‌های نرمافزاری به مراحلی چون امکان‌سنجی، بررسی نیازها و محدودیت‌ها، تجزیه و تحلیل راه حل‌ها و غیره نیاز دارد. علاوه بر این‌ها در مقوله پایگاهداده، باید لایه‌های پنج‌گانه بانک را در نظر گرفت. مراحل طراحی پایگاهداده‌ها را می‌توان به صورت زیر خلاصه کرد:

۱. مطالعه و شناخت مسئله
۲. تعیین نیازها و محدودیت‌ها
۳. مدل‌سازی معنایی (بدون توجه به پیاده‌سازی)
۴. طراحی منطقی (پس از انتخاب مدلی برای پیاده‌سازی)
۵. طراحی داخلی (پروندها و شاخص‌ها در سطح منطقی)
۶. طراحی فیزیکی (نحوه قرارگرفتن داده‌ها روی رسانه‌ها)
۷. طراحی تراکنش‌ها و واسطه‌های کاربری

شکل ۱-۸-۱ مراحل طراحی پایگاهداده را نمایش می‌دهد:

بررسی و تجزیه و تحلیل	امکان‌سنجی.
مدل‌سازی معنایی	<ul style="list-style-type: none"> - شناخت نیازها و محدودیت‌ها انتخاب راه حل‌ها. - تجزیه و تحلیل راه حل‌ها و انتخاب راه حل مناسب.
طراحی منطقی	<ul style="list-style-type: none"> - شناخت بخش‌های مختلف سیستم و ارتباط بین آن‌ها. - شناخت دقیق واحدهای هر بخش و ارتباط بین آن‌ها. - طراحی شماتیک بانک با استفاده از مدلی مانند EER یا UML
طراحی داخلی و فیزیکی	<ul style="list-style-type: none"> - انتخاب مدل مناسب. - تبدیل طراحی معنایی به ساختارهای مدل انتخابی. - کدکردن بانک با استفاده از زبان مناسب.
طراحی تراکنش‌ها و دیدها و واسطه‌های کاربری	<ul style="list-style-type: none"> - طراحی پرونده‌ها و شاخص‌ها روی رسانه‌ها با استفاده از انواع سازمان پرونده - طراحی دیدهای کاربران در طول زمان با در نظر گرفتن مسائل امنیتی

شکل ۱-۸-۱ مراحل طراحی پایگاهداده‌ها

۱-۶-۲ ابزار طراحی پایگاه‌داده‌ها

پس از بررسی کامل مسئله و شناخت نیازها و محدودیت‌ها، مدل‌سازی معنایی صورت می‌گیرد. در این مرحله تمام بخش‌ها در ارتباط با یکدیگر قرار می‌گیرند.

متداول‌ترین روش‌های این‌کار، مدل‌های EER و UML هستند، که در فصل بعد بررسی خواهند شد. پس از نهایی‌شدن طراحی معنایی، مدل لازم برای پیاده‌سازی آن انتخاب می‌شود. انتخاب مدل بسیار مهم است و بستگی به طبیعت کار مورد نظر دارد. هرچند مدل‌های EER و UML هیچ‌گاه پیاده‌سازی نشده و اصولاً ابزار طراحی هستند نه پیاده‌سازی، ولی می‌توان بانک طراحی‌شده در این مدل‌ها را، به ساختارهای رابطه‌ای یا شئ‌گرا تبدیل کرد. طراحی منطقی و طراحی فیزیکی به حجم، انواع پرس‌وجو و سایر مشخصات بانک مورد نظر بستگی دارد. طراحی و بهینه‌سازی پایگاه‌داده در مدل رابطه‌ای، روشی کاملاً مشخص دارد که به نرمال‌سازی^۱ جداول مشهور است. بعضی از نظام‌های جدید مدیریت پایگاه‌داده (DBMS)، دارای ابزار کمکی هستند که در طراحی و پیاده‌سازی پایگاه‌داده مورد استفاده قرار می‌گیرند.^۲ از جمله این‌گونه ابزارها می‌توان از رسم نمودار EER، تبدیل آن به جداول، کار با لغت‌نامه داده‌ها، تهیه نمودارها و گزارش‌ها، طراحی شئ‌گرا و رسم نمودارهای آن و مهندسی وارون، که طراحی را از سیستم پیاده‌شده استخراج می‌کند (مهندسی معکوس^۳) نام برد. ابزارهای نرم‌افزاری ویژه‌ای عرضه شده‌اند که طراحی مفهومی عام را در قالب EER می‌گیرند و طراحی مفهومی خاص را در مدل رابطه‌ای تولید می‌کنند.

۱-۷ جداول نمونه

در پایان این فصل پایگاه‌داده نشر کتاب را که در همه فصل‌ها مورد استفاده قرار می‌گیرد در فرم جداول در مدل رابطه‌ای می‌آوریم:

personel(pno, name, typ, dob, sex, adres)

(آدرس، جنسیت، تاریخ تولد، نوع، نام شخص، کلید) کارکنان

Author (ano, name, typ, dob, sex, adres, degree, major, resum)

(رزومه، رشته، مدرک تحصیلی، آدرس، جنسیت، تاریخ تولد، نوع، نام مؤلف،

کلید) مؤلف

1. Normalization
2. Computer Aided Software Engineering (CASE)
3. Reverse Engineering

Book (bno, title, dop, typ, version, ano)

(کلید مؤلف، نسخه، نوع کتاب، تاریخ چاپ، نام کتاب، کلید) کتاب

Publisher (pno, title, typ, doe, adres)

(آدرس، تاریخ تأسیس، نوع انتشارات، عنوان ناشر، کلید مدیر نشر) ناشر

Publish (pubno, pno, bno, dop, Uprice, num)

(تیراژ، قیمت واحد، تاریخ نشر، کلید کتاب، کلید مدیر نشر، کلید) انتشار

Store (sno, title, doe, aders, pno)

(کلید مدیر فروشگاه، آدرس، تاریخ تأسیس، نام فروشگاه، کلید) کتابفروشی

Rep (rno, title, typ, doe, adres, pno)

(کلید مسئول انبار، آدرس، تاریخ تأسیس، نوع، نام انبار، کلید) انبار

sell(sno, bno, num, name, sdate, price, adres)

(آدرس، قیمت، تاریخ فروش، نام خریدار، تعداد، کلید کتاب، کلید) فروش

خلاصه فصل اول

در این فصل کوشیدیم شناختی فراگیر، اما مقدماتی، از پایگاهداده‌ها ارائه دهیم یعنی با اقیانوسی پهناور به نام پایگاهداده‌ها آشنا شدیم، اما عمق این اقیانوس را در نیافتیم. آموختیم که پردازش داده‌ها در چه شکل‌هایی انجام شده تا به صورت پایگاهداده‌های پیچیده و گستردۀ امروزی در آمده است. آموختیم که داده‌ها در زندگی بشر بسیار حیاتی و تأثیرگذارند و باید از خطاهای گریزناپذیر انسانی به دور باشند. آموختیم که کنترل داده‌ها توسط نرم‌افزارهای نظام مدیریت پایگاهداده‌ها (DBMS) چگونه می‌تواند امنیّت و جامعیّت پایگاهداده‌ها را تأمین و ذخیره و بازیابی حجم وسیعی از داده‌ها را تسهیل کند. آموختیم که نظام مدیریت پایگاهداده‌ها چه لایه‌ها و اجزایی دارد، چگونه قابل مدل‌سازی است، چه ویژگی‌ها و امتیازهایی دربر دارد، و چگونه طراحی و پیاده‌سازی می‌شود. عمده‌ترین مفاهیم پایگاهداده‌ها چون جدول، تراکنش، پرسش و بهینه‌سازی آن، جامعیّت و امنیّت، شاخص و... نیز معرفی شدند تا در فصل‌های آینده مورد کنکاش قرار گیرند.

خودآزمایی تشریحی فصل اول

۱. یک تراکنش در پایگاهداده بانکداری مثال بزنید و اهمیت خواص ACID را روی آن شرح دهید.

حل:

شروع تراکنش انتقال پول

الف) کارت خود را وارد کنید. {ورود کارت به دستگاه}
 ب) رمز کارت را وارد کنید. {بررسی صحت رمز و واکنش مناسب دستگاه}
 ت) مبلغ مورد نظر را وارد کنید. {کنترل‌های لازم جهت انتقال مبلغ}
 ث) شماره کارت مقصد را وارد کنید. {بررسی حساب مقصد و اقدام جهت انتقال مبلغ}

ج) کارت و رسید خود را بردارید.

پایان تراکنش انتقال پول

اهمیت خواص ACID این است که در صورت بروز هرگونه مشکلی، خطایی در انتقال پول صورت نگیرد. مثلاً اگر در خلال انجام مرحله ۴، یعنی کسر مبلغ از حساب مبدأ، اتصال به حساب مقصد و واریز مبلغ به حساب مقصد مشکلی پیش آید، چه باید کرد؟ مثلاً اگر پول از حساب مبدأ کسر شده باشد ولی انتقال آن به حساب مقصد با مشکلاتی مانند قطع برق یا اینترنت روبرو شود، در این صورت دستگاه کارهای کارهای قبلی را ملغی می‌کند.

به عبارت دیگر مبلغ کسر شده مجدداً توسط دستگاه به حساب مبدأ واریز و تراکنش ساقط می‌شود و به اطلاع کاربر می‌رسد.
 ۲. آیا طراحی معنایی در طراحی پایگاه داده‌ها همواره ضروری است؟ چه امتیازی دارد؟

حل:

خیر، بسیار راهگشاست ولی گاهی ضروری نیست. اگر محیط و مدل پیاده‌سازی پایگاهداده مورد نظر، از ابتدا مشخص باشد می‌توان در قالب آن مدل، طراحی را انجام داد. مثلاً اگر قرار است پایگاه مورد نظر در مدل رابطه‌ای پیاده‌سازی شود، می‌توان از ابتدا جداول را طراحی کرد.

عمده‌ترین دلایل استفاده از طراحی معنایی، عبارت‌اند از:

الف) بدون پرداختن به ریزه‌کاری‌های پیاده‌سازی، می‌توان روی طراحی مرکز شد و سیستم کامل و همه‌جانبه‌ای ارائه داد که در شرایط متفاوت به صورت‌های نسبتاً مختلفی پیاده‌سازی شود.

ب) در سیستم‌های ناشناخته که از ابتدا ویژگی‌ها و ابعاد آن‌ها مشخص نیست، می‌توان به صورتی سیستماتیک اجزای سیستم و ارتباط آن‌ها را با یکدیگر تعیین کرد.

ت) در سیستم‌های خیلی بزرگ می‌توان کار را به چند بخش تقسیم کرد، هر بخش را جداگانه طراحی کرد و درنهایت به یکدیگر ارتباط داد. در مثال‌های کوچکی که اغلب در کتاب‌های درسی مطرح می‌شود، به دلیل طولانی نشدن مطلب، معمولاً این جنبه کار دیده نمی‌شود.

ث) اگر کار بزرگ باشد و برای پرداختن به آن به یک تیم نیاز باشد، در این صورت این‌گونه ابزارها به تقسیم کار و ترکیب نتایج در یک قالب هماهنگ، کمک می‌کنند.

ج) نمودار طراحی معنایی، دید یکسان از پایگاهداده ایجاد می‌کند یعنی ذهنیت‌های دست‌اندرکاران، به هم نزدیک می‌شود و برداشت یکنواخت و مشابهی از چیستی مسئله ایجاد می‌کند.

ح) نمودار طراحی معنایی را می‌توان در شرایط مختلف به صورت‌های متفاوت و در مدل‌های گوناگون پیاده‌سازی کرد. حتی می‌توان بخش‌های یک سیستم را در مدل‌های متفاوت پایگاهداده پیاده‌سازی کرد، به شرطی که با یکدیگر هم‌خوانی داشته باشند.

۳. پایگاهداده مقدماتی برای آموزش رانندگی در قالب مدل رابطه‌ای طراحی نمایید.

حل:

الف) شناخت عمومی از این پایگاهداده به ملزومات زیر می‌انجامد:

- جدول برای آموزشگاه رانندگی شامل ستون‌هایی چون آدرس، تاریخ تأسیس، نام، کد آموزشگاه

- جدول برای مشخصات مدیر شامل ستون‌هایی چون آدرس، شناسه کسب، جنسیت، تاریخ تولد، نام
- جدول برای مشخصات مربی شامل ستون‌هایی چون آدرس، پایه آموزش، جنسیت، تاریخ تولد، نام، کد مربی
- جدول برای مشخصات دانشجو شامل ستون‌هایی چون آدرس، جنسیت، تاریخ تولد، نام
- جدول برای مشخصات مدیریت شامل ستون‌هایی چون تاریخ ترک کار، تاریخ استخدام، کلید مدیر، کلید آموزشگاه
- جدول برای مشخصات مدرس شامل ستون‌هایی چون تاریخ ترک کار، تاریخ استخدام، کلید مربی، کلید آموزشگاه
- ب) جداول مقدماتی پایگاهداده آموزشگاه را نندگی در شکل ۹-۱ آمده است.
- ت) پرسش‌ها و تراکنش‌های متنوعی می‌توان برای این پایگاهداده در نظر گرفت. نمونه‌های زیر بخشی از آن‌ها است:

Academy (<u>ac#</u> , code, name, doe, adres)	(آدرس، تاریخ تأسیس، نام، کد آموزشگاه، کلید) آموزشگاه
admin(<u>ad#</u> , name, dob, sex, register, adres)	(آدرس، شناسه کسب، جنسیت، تاریخ تولد، نام، کلید) مدیر
Coach (<u>co#</u> , name, dob, sex, degree, adres)	(آدرس، پایه آموزش، جنسیت، تاریخ تولد، نام، کد مربی، کلید) مربی
Student (<u>st#</u> , name, dob, sex, adres)	(آدرس، جنسیت، تاریخ تولد، نام، کلید) کارآموز
Mng (<u>mn#</u> , ac#, ad#, <u>doa</u> , doe)	(تاریخ ترک کار، تاریخ استخدام، کلید مدیر، کلید آموزشگاه، کلید) مدیریت
Training (<u>tr#</u> , ac#, co#, <u>doa</u> , doe)	(تاریخ ترک کار، تاریخ استخدام، کلید مربی، کلید آموزشگاه، کلید) تدریس
Admission (<u>as#</u> , st#, tr#, <u>doa</u> , doe, dure, comm)	(طول دوره، تاریخ پایان، تاریخ شروع، کلید مربی، کلید دانشجو، کلید پذیرش) توضیحات

شکل ۹-۱. جداول پایگاهداده آموزش را نندگی

*** اطلاعات آموزشگاه‌های رانندگی که قبل از همه تأسیس شده است. (بر اساس کد مشخص می‌شود)

*** اطلاعات آموزشگاه‌های رانندگی که تاریخ تأسیس آن‌ها بین ۱۳۸۰ تا ۱۳۹۰ بوده است.

*** نام کارآموزان و نام آموزشگاه‌هایی که در آن ثبت‌نام کرده‌اند.

*** کارآموزانی که بیشترین ثبت‌نام را داشته‌اند.

*** نام مربيان و آموزشگاه‌ها و کد آموزشگاه.

*** نام مربيانی که فقط در یک آموزشگاه آموزش داشته‌اند.

*** مربيانی که در آموزشگاه‌های مختلف آموزش داشته‌اند.

*** نام مربی و نام آموزشگاه برای مربيانی که در بیش از یک آموزشگاه آموزش داشته‌اند.

*** آموزشگاه‌هایی که به غیر از سال ۹۳ و ۹۴ تأسیس شده‌اند.

*** کارآموزانی که در بیش از یک آموزشگاه ثبت‌نام کرده‌اند.

*** هر آموزشگاه مجموع درآمدش از ثبت‌نام دانشجویان چقدر بوده است.

*** مربيانی که طول دوره آن‌ها بیش از ۳۰ ساعت طول کشیده است.

*** تعداد کارآموزانی که ساکن خیابان پروین هستند.

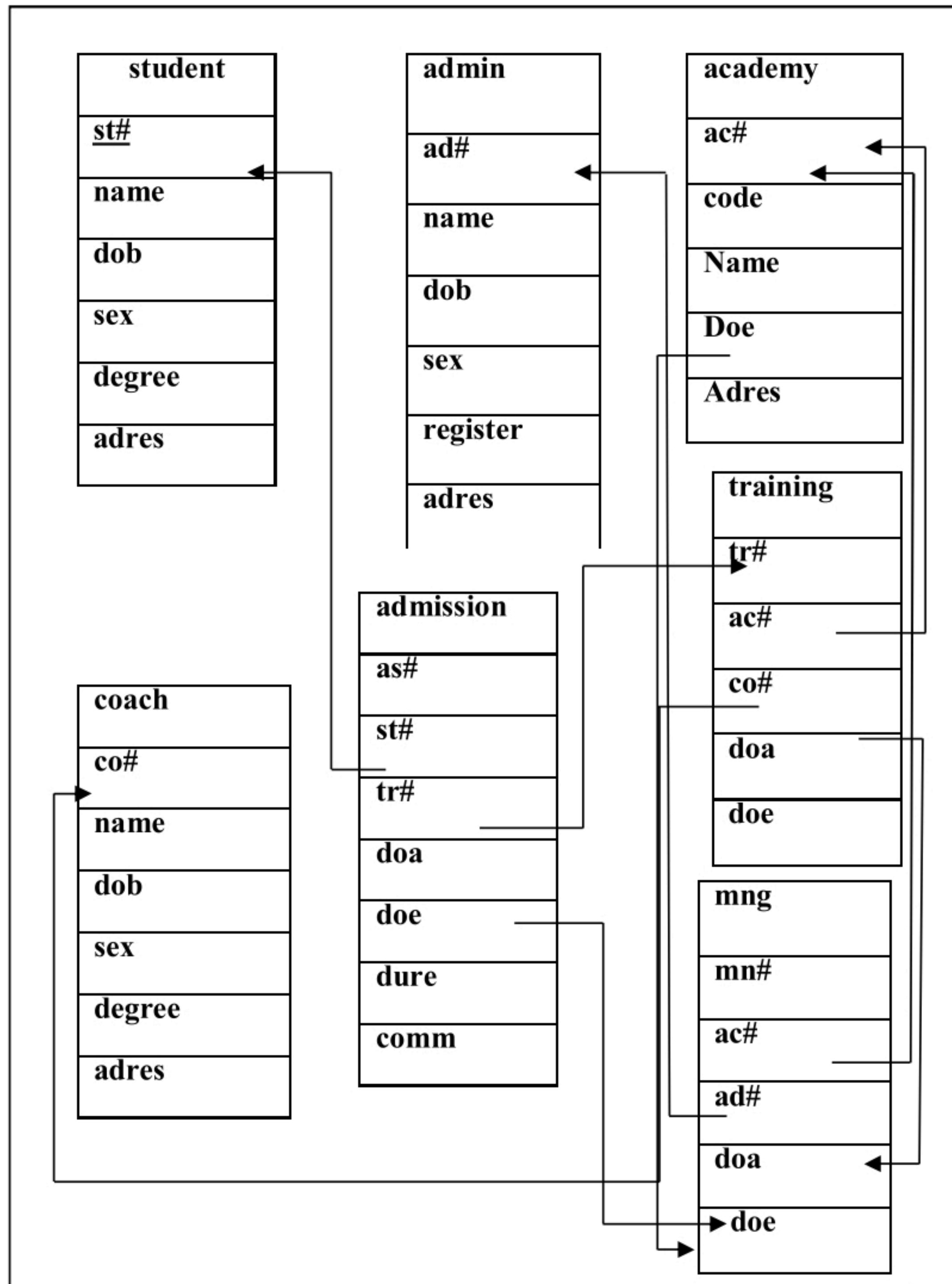
*** بیشینه طول دوره در پذیرش کارآموزان.

*** کد آموزشگاه‌هایی که تعداد مربيان آن بیش از ۸ نفر باشد.

*** نام کارآموزانی که با علی محمدی آموزش داشته‌اند.

ث) نمودار شمایی پایگاهداده رابطه‌ای آموزش رانندگی در شکل ۱۰-۱ آمده است.

ج) طراحی داخلی و فیزیکی این پایگاهداده، پس از انتخاب سخت‌افزار مناسب برای آن، قابل انجام است.



شکل ۱۰-۱. نمودار شمای پایگاهداده آموزش رانندگی

۴. پایگاه داده مقدماتی برای کلینیک دندانپزشکی در قالب مدل رابطه‌ای طراحی کنید.

حل:

الف) شناخت عمومی از این پایگاهداده به ملزومات زیر می‌انجامد:

- جدول برای مشخصات دندانپزشک، شامل ستون‌هایی چون آدرس، جنسیت، تاریخ تولد، شماره شناسنامه، ثبت‌نام، تاریخ ترک کار
 - جدول برای مشخصات منشی، شامل ستون‌هایی چون تاریخ ترک کار، تاریخ استخدام، آدرس، جنسیت، تاریخ تولد، نام
 - جدول برای مشخصات بیمار، شامل ستون‌هایی چون شماره پرونده، آدرس، جنسیت، تاریخ تولد، نام
 - جدول برای پذیرش، شامل ستون‌هایی چون تاریخ مراجعة بعدی، شماره دندان، نوع کار، تاریخ پذیرش، کلید پزشک، کلید منشی، کلید بیمار
- ب) جداول مقدماتی پایگاهداده کلینیک دندانپزشکی در شکل ۱۱-۱ آمده است.

Dentist (de#, name, register, sh#, dob, sex, adres, doa, doe)

(تاریخ تأسیس، تاریخ ترک کار، آدرس، جنسیت، تاریخ تولد، ش شناسنامه، ثبت، نام، کلید
دندانپزشک)

secr (se#, name, dob, sex, adres, doa, doe)

(تاریخ ترک کار، تاریخ استخدام، آدرس، جنسیت، تاریخ تولد، نام، کلید) منشی

Patient (pa#, name, dob, sex, adres, file#)

(ش پرونده، آدرس، جنسیت، تاریخ تولد، نام، کلید) بیمار

admission (ad#), pa#, ndate, typ, de#, rdate

(تاریخ مراجعة بعدی، ش دندانپزشک، نوع کار، تاریخ پذیرش، کلید بیمار، کلید) پذیرش

شکل ۱۱-۱. جداول مقدماتی کلینیک دندانپزشکی

ت) پرسش‌ها و تراکنش‌های متنوعی می‌توان برای این پایگاهداده در نظر گرفت.
نمونه‌های زیر بخشی از آن‌ها است:

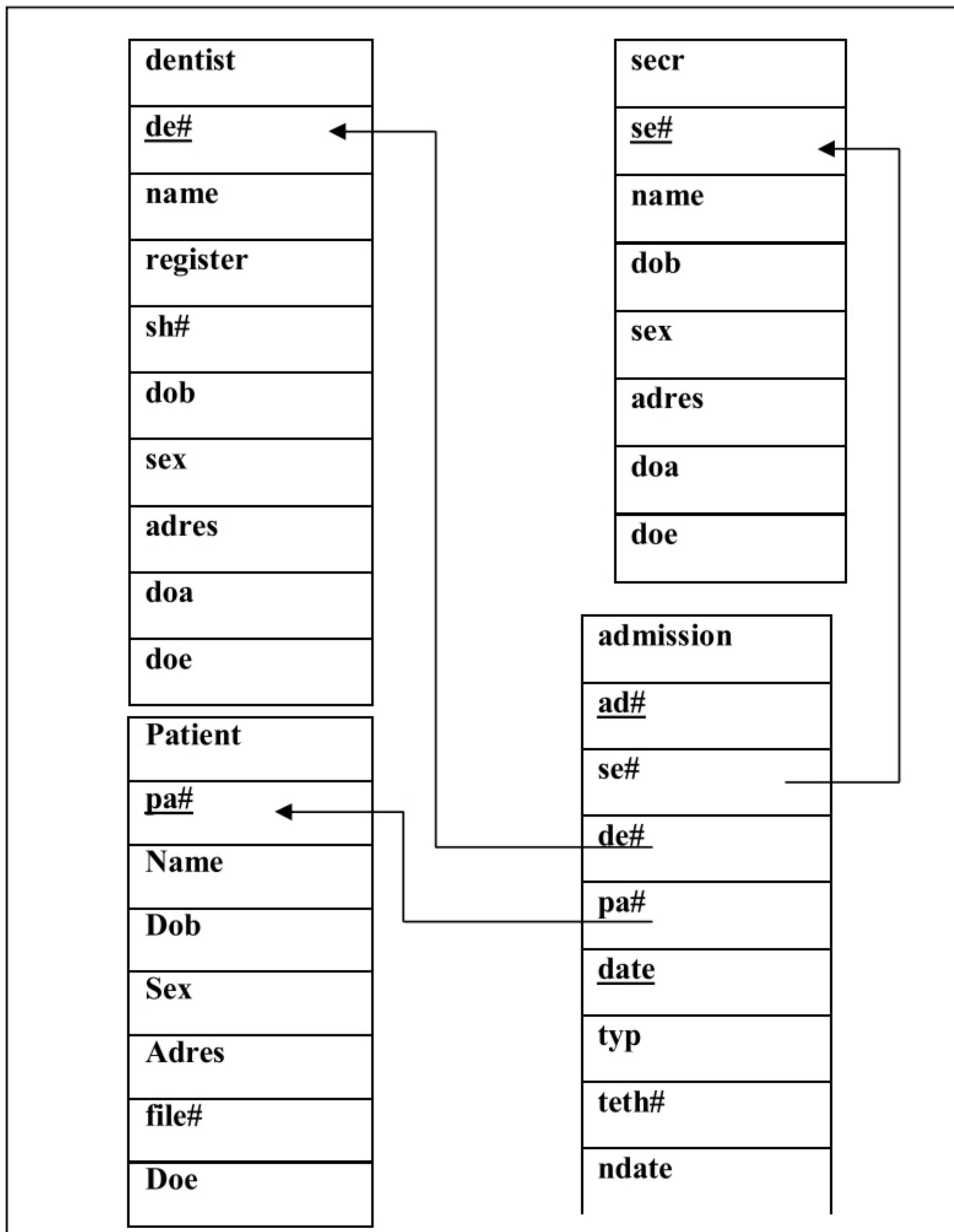
*** دندانپزشکانی که در سال‌های ۱۳۷۰ تا ۱۳۸۰ استخدام و ترک کار داشته‌اند.

*** منشیانی که برای دکتر داودی، پذیرش انجام داده‌اند.

*** تعداد بیمارانی که در این کلینیک پذیرش شده‌اند.

*** نام بیمارانی که فقط توسط یک پزشک پذیرش شده‌اند.

** نام بیمارانی که توسط پزشک‌های مختلف پذیرش شده‌اند.
ث) نمودار شمای پایگاهداده رابطه‌ای کلینیک دندانپزشکی در شکل ۱۲-۱ آمده است.



شکل ۱۲-۱. نمودار شمای پایگاهداده رابطه‌ای کلینیک دندانپزشکی

ج) طراحی داخلی و فیزیکی این پایگاهداده پس از انتخاب سخت‌افزار مناسب برای آن قابل انجام است.

تمرین‌های تشریحی فصل اول

۱. نمودار شمای پایگاهداده دندانپزشکی را در مدل سلسله‌مراتبی رسم کنید.
۲. نمودار شمای پایگاهداده دندانپزشکی را در مدل شبکه‌ای رسم کنید.
۳. اگر امکان استقلال داده‌ای در پایگاهداده فراهم نباشد، چه مشکلی پیش می‌آید؟
۴. تحقیق کنید چگونه در زبان‌های برنامه‌سازی مانند جاوا و سی و پاسکال می‌توان با پایگاهداده‌ها ارتباط برقرار کرد.
۵. اگر امکان مدل‌سازی داده‌ها از پایگاهداده‌ها حذف شود، چه اتفاقی می‌افتد؟
۶. برای وجود ناسازگاری در پایگاهداده آموزش رانندگی مثال بزنید.
۷. چگونه می‌توان در پایگاهداده آموزش رانندگی، از پرداخت حقوق خیلی بالا، پیشگیری کرد؟
۸. برای دستیابی موافقی به داده‌ها در پایگاهداده بانکداری، مثال بزنید.
۹. کنترل هم‌روندي یعنی چه؟
۱۰. اگر خاصیت پایایی از تراکنش حذف شود، چه اتفاقی می‌افتد؟
۱۱. یک مثال برای پایگاهداده نامتمرکز (توزیعی) ارائه کنید.
۱۲. یک مورد مثال بزنید که نرم‌افزار DBMS دچار مشکل می‌شود و دخالت انسان ضروری است.
۱۳. در مورد مفهوم «بازگرد» تحقیق کنید. چگونه نرم‌افزار DBMS می‌تواند کارهای انجام‌شده مثل برداشت پول از یک حساب بانکی را ختی کند؟
۱۴. چرا مفهوم شئ در پایگاهداده نسبت به مفهوم جدول کارایی بیشتری دارد؟ به عبارت دیگر چرا مدل‌های شئ‌گرا و شئ - رابطه‌ای، نسبت به مدل رابطه‌ای، توانمندی بیشتری دارند؟
۱۵. شبکه‌های اجتماعی چگونه می‌توانند حجم وسیعی از داده‌ها را برای تعداد زیادی کاربر به سرعت ذخیره و بازیابی کنند؟
۱۶. چند امتیاز دیگر (علاوه بر امتیازهای پایگاهداده‌ها) برای پایگاهداده نامتمرکز (توزیعی) نام ببرید.
۱۷. چگونه می‌توان از سوءاستفاده برنامه‌سازان پایگاهداده‌ها جلوگیری کرد؟ مثلاً یک کارمند بانک نتواند بخشی از سود بعضی از حساب‌ها را به حساب خود یا دیگری واریز کند.