
آموزش گام به گام بانک اطلاعاتی با پایتون

تألیف:

دکتر رمضان عباس نژادورزی



فن آوری نوین

سرشناسه	:	عباس نژاد ورزی، رمضان، ۱۳۴۸ -
عنوان و نام پدیدآور	:	آموزش گام به گام بانک اطلاعاتی با پایتون / تالیف رمضان عباس نژاد دورزی.
مشخصات نشر	:	بابل: فناوری نوین، ۱۳۹۹.
مشخصات ظاهری	:	۲۰۸ص: جدول.
شابک	:	۹۷۸-۶۰۰-۷۲۷۲۴۱-۱: ۴۵۰۰۰۰۰۰ریال
وضعیت فهرست نویسی	:	فیبا
یادداشت	:	کتابنامه.
موضوع	:	پایتون (زبان برنامه نویسی کامپیوتر)
موضوع	:	Python (Computer program language)
رده بندی کنگره	:	QA۷۶/۷۳
رده بندی دیویی	:	۰۰۵/۱۳۳
شماره کتابشناسی ملی	:	۶۱۲۳۶۶۶

www.fanavarienovin.net

تلفن: ۰۱۱-۳۲۲۵۶۶۸۷

بابل، کد پستی ۴۷۱۶۷-۷۳۴۴۸

فن آوری نوین

آموزش گام به گام بانک اطلاعاتی با پایتون

تألیف: رمضان عباس نژاد دورزی.

نوبت چاپ: چاپ اول

سال چاپ: بهار ۹۹

شمارگان: ۲۰۰

قیمت: ۴۵۰۰۰ تومان

نام چاپخانه و صحافی: دفتر فنی سورنا

شابک: ۹۷۸-۶۰۰-۷۲۷۲-۴۱-۱

نشانی ناشر: بابل، چهارراه نواب، کاظم بیگی، جنب مسجد منصور کاظم بیگی، طبقه اول

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

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

فهرست مطالب

۷	فصل اول: مروری بر بانک اطلاعاتی
۷-۱	تعریف سیستم مدیریت بانک اطلاعات
۷-۲	دلایل استفاده از بانک اطلاعات
۹-۳	مراحل طراحی بانک اطلاعاتی
۹-۳-۱	تعیین کاربرد اصلی بانک اطلاعاتی
۹-۳-۲	تعیین جداول موردنیاز بانک اطلاعاتی
۱۰-۳-۳	تعیین فیلدهای موردنیاز بانک اطلاعاتی
۱۳-۳-۴	تعریف رابطه‌های بین جداول
۱۴-۳-۵	بهبودسازی طراحی
۱۴-۴	مدل‌های داده‌ای متداول
۱۵-۴-۱	مدل داده ثبت سفارش مشتری
۱۵-۴-۲	مدل داده کنترل موجودی
۱۹-۴-۳	مدل داده مدیریت سرمایه
۲۰-۴-۴	مدل داده مدیریت پروژه
۲۲-۴-۵	مدل داده مدیریت آموزش
۲۵-۴-۶	مدل داده مدیریت منابع انسانی
۳۱-۴-۷	مدل داده مدیریت حقوق و دستمزد
۳۳-۴-۸	مدل داده کرایه اتومبیل
۳۷	فصل دوم: بانک‌های اطلاعاتی رابطه‌ای و دستورات SQL
۳۷-۲-۱	پایگاه داده‌های رابطه‌ای
۳۷-۲-۲	پایگاه داده‌های رایج برای برنامه‌های کاربردی در زبان پایتون
۳۸-۲-۳	پایگاه داده SQLite
۴۰-۲-۳-۱	ویژگی‌های SQLite در زبان پایتون
۴۰-۲-۳-۲	کاربرد SQLite در زبان پایتون
۴۲-۲-۳-۳	مزایا و معایب SQLite در زبان پایتون
۴۳-۲-۳-۴	معایب SQLite در پایتون
۴۳-۲-۴	بانک اطلاعاتی MySQL
۴۴-۲-۴-۱	ورود به بانک اطلاعاتی MySQL
۴۴-۲-۴-۲	تایپ و اجرای دستورات SQL
۴۴-۲-۴-۳	ایجاد بانک اطلاعاتی
۴۵-۲-۴-۴	حذف بانک اطلاعاتی موجود
۴۵-۲-۴-۵	ایجاد و تغییر ساختار جدول
۴۷-۲-۴-۶	انواع داده در MySQL
۵۰-۲-۴-۷	ایجاد جدول
۵۳-۲-۴-۸	تغییر ساختار جدول با دستور SQL
۵۵-۲-۴-۹	حذف جدول با دستور SQL
۵۵-۲-۴-۱۰	ورود، ویرایش، حذف و بازیابی اطلاعات
۵۷-۲-۴-۱۱	دستور INSERT

۶۰	ویرایش رکوردهای جدول
۶۱	حذف رکوردهای جدول
۶۲	پرس و جوی باز یابی اطلاعات
۶۵	باز یابی اطلاعات از جدول با دستور SELECT
۷۰	مرتب سازی رکوردها
۷۱	توابع تجمعی
۷۱	گروه بندی اطلاعات
۷۳	پرس و جوی فرعی
۷۶	پیوند جداول (رابطه)
۸۲	فصل سوم: بانک اطلاعاتی و پایتون
۸۲	۳-۱ پایتون و MySQL
۸۲	۳-۱-۱ نصب گرداننده MySQL در پایتون
۸۲	۳-۱-۲ بررسی صحت MySQL Connector نصب شده
۸۲	۳-۱-۳ اتصال به MySQL با استفاده از Connector پایتون
۸۴	۳-۱-۴ کد گذاری کاراکتر (Character Encoding)
۸۴	۳-۱-۵ تراکنش ها
۸۴	۳-۱-۶ منطقه زمانی
۸۴	۳-۱-۷ مدهای SQL
۸۵	۳-۱-۸ عیب یابی و مدیریت خطا
۸۵	۳-۱-۹ فعال و غیر فعال کردن خاصیت ها با استفاده از flags سرویس گیرنده
۸۵	۳-۱-۱۰ بافر کردن مجموعه نتایج
۸۵	۳-۱-۱۱ تبدیل انواع
۸۵	۳-۱-۱۲ اتصال از طریق SSL
۸۶	۳-۱-۱۳ سازگاری با رابط های اتصال دیگر
۸۶	۳-۱-۱۴ ماژول mysql.connector
۸۷	۳-۱-۱۵ کلاس connection.MySQLConnection
۹۴	۳-۱-۱۶ کلاس cursor.MySQLCursor
۹۸	۳-۱-۱۷ کلاس cursor.MySQLCursorBuffered
۹۹	۳-۱-۱۸ کلاس constants.ClientFlag
۹۹	۳-۱-۱۹ کلاس constants.FieldType
۹۹	۳-۱-۲۰ کلاس constants.SQLMode
۱۰۰	۳-۱-۲۱ کلاس constants.CharacterSet
۱۰۰	۳-۱-۲۲ کلاس constants.RefreshOption
۱۰۰	۳-۱-۲۳ ایجاد بانک اطلاعاتی در MySQL با پایتون
۱۰۱	۳-۱-۲۴ ایجاد جدول در MySQL در پایتون

۱۰۱	۳-۱-۲۵	بررسی وجود داشتن جدول
۱۰۲	۳-۱-۲۶	درج اطلاعات در جدول MySQL در پایتون
۱۰۳	۳-۱-۲۷	درج چند سطر در جدول
۱۰۴	۳-۱-۲۸	به دست آوردن ID آخرین رکورد
۱۰۵	۳-۱-۲۹	دستور UPDATE در MySQL در پایتون
۱۰۶	۳-۱-۳۰	جلوگیری از SQL Injection
۱۰۷	۳-۱-۳۱	حذف رکوردهای جدول
۱۰۸	۳-۱-۳۲	حذف جدول
۱۰۹	۳-۱-۳۳	حذف بانک اطلاعاتی
۱۰۹	۳-۱-۳۴	دستور SELECT در MySQL در پایتون
۱۱۰	۳-۱-۳۵	انتخاب ستون‌ها
۱۱۱	۳-۱-۳۶	کاراکترهای Wildcard
۱۱۲	۳-۱-۳۷	مرتب‌سازی نتایج MySQL در پایتون
۱۱۳	۳-۱-۴۳	رویه‌های ذخیره شده
۱۱۳	۳-۱-۳۸	مرتب‌سازی به صورت نزولی
۱۱۴	۳-۱-۳۹	استفاده از پارامتر LIMIT مربوط به MySQL در پایتون
۱۱۵	۳-۱-۴۰	شروع از موقعیت دلخواه
۱۱۶	۳-۱-۴۱	پیوند جداول MySQL در پایتون
۱۲۱	۳-۱-۴۳	رویه‌های ذخیره شده
۱۲۵	۳-۱-۴۴	مدیریت تراکنش
۱۳۱	۳-۲	بانک اطلاعاتی SQLite
۱۳۱	۳-۲-۱	ماژول SQLite۳
۱۳۱	۳-۲-۲	برقراری اتصال با بانک اطلاعاتی
۱۳۱	۳-۲-۳	قطع ارتباط با بانک اطلاعاتی
۱۳۱	۳-۲-۴	قطع ارتباط با بانک اطلاعاتی
۱۳۶	۳-۳	پایتون و بانک اطلاعاتی PostgreSQL
۱۳۷	۳-۳-۱	نصب Psycopy۲
۱۳۷	۳-۳-۲	ایجاد بانک اطلاعاتی PostgreSQL
۱۴۱	۳-۳-۳	اتصال به پایگاه داده PostgreSQL و اجرای دستورات SQL
۱۵۳	۳-۴	پایتون و بانک اطلاعاتی اکسس
۱۵۳	۳-۴-۱	نصب گرداننده بانک اطلاعاتی اکسس در پایتون
۱۵۴	۳-۴-۲	بررسی صحت pyodbc نصب شده
۱۵۴	۳-۴-۳	ایجاد اتصال با بانک اطلاعاتی اکسس
۱۶۰		فصل چهارم: پیاده‌سازی پروژه‌های مختلف بانک‌های اطلاعاتی با پایتون
۲۰۸		منابع

مقدمه

امروزه حجم زیادی از اطلاعات ذخیره و بازیابی می‌شوند. برای جلوگیری از افزونگی داده (تکرار بی‌مورد داده‌ها)، بی‌نظمی و ایجاد سازگاری بین گزارش‌ها از بانک اطلاعات استفاده می‌شود. پایگاه داده‌ای که اغلب در توسعه برنامه‌های کاربردی و وب‌سایت‌ها در پایتون مورد استفاده قرار می‌گیرد، مجموعه‌ای از جداول رابطه‌ای است که به نام پایگاه داده‌های رابطه‌ای (Relational Database) می‌شناسیم. پایگاه داده‌های رابطه‌ای در حقیقت داده‌ها را در قالب جداول ذخیره می‌کنند. ارتباطات بین جداول به‌عنوان کلید خارجی تعریف می‌شوند. یک کلید خارجی، یک مرجع منحصر به فرد از یک ردیف در یک جدول که با ردیف دیگری در یک جدول یا جداول دیگر در ارتباط است، می‌باشد.

پیاده‌سازی روش‌های ذخیره‌سازی اطلاعات در بانک‌های اطلاعاتی بسیار پیچیده است. به‌طور مثال، SQLite یک پایگاه داده است که در پایتون قرار دارد. این پایگاه داده برای کلیه داده‌ها که در قالب یک بانک اطلاعاتی هستند، از یک فایل استفاده می‌کند. ولی پایگاه‌های داده دیگر مانند PostgreSQL، MySQL، Oracle و Microsoft SQL Server، طرح‌های بسیار پیچیده‌ای را ارائه می‌دهند تا ویژگی‌های پیشرفته‌ای را ارائه دهند که برای ذخیره‌سازی داده‌های برنامه‌های کاربردی و تحت وب مفید هستند. در این کتاب بسته‌های پایتون برای انجام پرس‌وجو از پایگاه داده‌های مختلف مانند PostgreSQL، MySQL، SQLite و Microsoft Access شرح داده است. علاوه بر این عملیات مختلف از قبیل وصل شدن به پایگاه داده، ورود، ویرایش، حذف و بازیابی رکوردها، ایجاد و استفاده از رویه‌های ذخیره شده و پیاده‌سازی تراکنش‌ها در بانک اطلاعات بیان گردیده است.

کتاب حاضر با بهره‌گیری از سال‌ها تجربه در امر تدریس، تألیف کتب کامپیوتر و مهم‌تر از همه برنامه‌نویسی در زمینه بانک اطلاعات تدوین شده است. از ویژگی‌های جالب و برجسته این کتاب، بیان مثال‌های متنوع کاربردی، حل گام به گام آن‌ها و توضیح کامل مثال‌های بیان شده، می‌باشد.

از تمامی اساتید و دانشجویان عزیز تقاضا داریم، هرگونه اشکال، ابهام در متن کتاب، پیشنهاد و انتقادات را به آدرس پست الکترونیک fanavarienovin@gmail.com ارسال نمایند.

در پایان امیدوارم این اثر نیز مانند دیگر آثار مورد توجه جامعه انفورماتیک کشور، اساتید و دانشجویان عزیز قرار گیرد.

مؤلف

fanavarienovin@gmail.com

مروری بر بانک اطلاعاتی

امروزه سازمان‌ها، مؤسسات، ادارات و شرکت‌ها با حجم عظیمی از داده‌ها سروکار دارند. به‌عنوان مثال، فرض کنید بخواهید اطلاعات مربوط به مکالمات شرکت مخابرات یکی از استان‌ها را نگهداری کنید. به‌طوری که در یک سال حدود ۲۵ میلیارد رکورد جمع‌آوری می‌گردد. نگهداری، پردازش و بازیابی این حجم اطلاعات از طریق فایل‌های معمولی زمان‌بر است. برای جلوگیری از تکرار بی‌مورد داده‌ها (افزونگی داده‌ها)، ایجاد سازگاری بین گزارش‌ها و صرفه‌جویی در میزان حافظه، به‌کارگیری بانک اطلاعات به‌صورت یک ضرورت درآمده است. یعنی، بدون استفاده از بانک اطلاعاتی نمی‌توان اطلاعات مربوط به مکالمات تلفن ثابت یک استان را نگهداری و ذخیره کرد. از طرف دیگر، اکثر برنامه‌های کاربردی که با داده‌ها سروکار دارند، داده‌ها را در بانک اطلاعات ذخیره می‌نمایند و از طریق بانک اطلاعات آن را پردازش می‌کنند.

بانک‌های اطلاعاتی متعددی وجود دارند که از جمله می‌توان Oracle، SQL Server، SQLite، Access، MySQL و DB2 را نام برد. هر یک از این بانک‌های اطلاعاتی کاربرد خاصی دارند.

۱-۱. تعریف سیستم مدیریت بانک اطلاعات

سیستم مدیریت بانک اطلاعاتی، مکانیزم نگهداری رکوردها^۱ است. یعنی، بانک اطلاعات مخزنی برای نگهداری از داده‌ها است که کاربران آن می‌توانند اعمالی از قبیل:

۱. افزودن جداول خالی به بانک اطلاعاتی

۲. افزودن رکوردهایی به جداول بانک اطلاعاتی

۳. تغییر ساختار جداول

۴. حذف رکوردهای بانک اطلاعاتی

۵. تغییر داده‌های بانک اطلاعاتی

۶. اجرای پرس‌وجو^۲ بر روی جداول

به عبارت ساده‌تر، سیستم مدیریت بانک اطلاعاتی، سیستمی رایانه‌ای است که هدف آن ذخیره و بازیابی داده‌ها می‌باشد. بانک اطلاعاتی داده را پردازش نموده به اطلاعات تبدیل کرده، آن‌ها را بازیابی می‌نماید.

۲-۲. دلایل استفاده از بانک اطلاعات

همان‌طور که می‌دانید، برنامه‌های کاربردی سطح بالا داده‌ها را ذخیره کرده و به روشی که برای کاربران قابل فهم و مفید باشد، این اطلاعات را به کاربران ارائه می‌دهند. به‌عنوان مثال، Google اطلاعات مربوط به

^۱- Records

^۲- Query

جاده‌ها و راه‌ها را ذخیره می‌کند و در نتیجه می‌توانیم با استفاده از برنامه Maps برای رانندگی از یک مکان به مکان دیگر از این اطلاعات بهره بگیریم. انتخاب مسیرهای رانندگی نیز امکان‌پذیر است. زیرا، داده‌ها به صورت ساختاریافته ذخیره می‌شوند.

پایگاه داده‌ها امکان ذخیره‌سازی ساختاری قابل‌اعتماد و سریع داده‌ها را ایجاد می‌کنند. همچنین، یک چارچوب (Framework) در اختیار ما می‌گذارند تا از طریق آن‌ها بتوانیم داده‌های خود را به‌سادگی ذخیره و بازیابی کنیم.

شاید از خودتان پرسید، بانک اطلاعاتی چه مزایایی دارد؟ پاسخ به این سؤال به مواردی از قبیل اندازه سیستم، تعداد کاربران سیستم و غیره بستگی دارد. هرچه اندازه سیستم بزرگ‌تر شود و تعداد کاربران بیشتر گردد، ضرورت به کارگیری بانک اطلاعاتی بیشتر خواهد شد. برای بیان مزایای بانک اطلاعاتی مثال زیر را در نظر بگیرید:

فرض کنید، برای فروشگاه ساده‌ای بانک اطلاعاتی طراحی می‌کنید. این بانک اطلاعاتی بسیار کوچک و ساده است و شاید امتیازات استفاده از بانک اطلاعاتی به چشم نیاید. اما همین بانک اطلاعاتی را برای فروشگاه زنجیره‌ای بزرگ در نظر بگیرید که دارای انبارهای زیادی است و موجودی انبارها به سرعت تغییر می‌کند. امتیازات سیستم بانک اطلاعاتی نسبت به سیستم سنتی که رکوردها بر روی کاغذ نگه‌داری می‌شوند، در این گونه موارد، بیش‌تر به چشم می‌آیند. برخی از مزایای بانک اطلاعاتی در زیر آمده‌اند:

✚ **فشرده‌گی:** چون داده‌های بانک اطلاعاتی دارای ساختار هستند، نیازی به نگه‌داری فایل‌های متنی حجیم نیست و از ورود داده‌های نامنظم (بدون ساختار) جلوگیری می‌کند. بنابراین، باعث فشرده‌سازی اطلاعات می‌گردد.

➤ **سرعت:** سیستم می‌تواند سریع‌تر از انسان داده‌ها را بازیابی و به‌هنگام کند. مخصوصاً سیستم توزیع شده باشد (مانند فروشگاه زنجیره‌ای)، پاسخ‌گویی به درخواست‌های **سراسری** و **موردی** بسیار سریع‌تر انجام می‌گردد.

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

➤ **دسترسی:** در هر زمان اطلاعات دقیق و به‌هنگام شده در اختیار قرار می‌گیرند. زیرا، اطلاعات به صورت مجتمع و یک‌پارچه نگه‌داری می‌شوند.

➤ **حفاظت:** داده‌ها می‌توانند در مقابل دستیابی غیرقانونی و غیرمجاز حفظ شوند. زیرا، اطلاعات در یک نقطه نگه‌داری می‌گردند. بنابراین، می‌توان از طریق فایروال، تعریف حساب کاربری و کلمه عبور از ورود افراد غیرمجاز جلوگیری کرد.

البته این مزایا در محیط چند کاربره که بانک‌های اطلاعاتی بزرگ و پیچیده باشند، چشم‌گیرتر است. اما، یک امتیاز ویژه در چنین محیطی وجود دارد و آن عبارت است از: **سیستم بانک اطلاعات موجود می‌شود تا مؤسسه بر روی داده‌هایش کنترل مرکزی داشته باشد** (این موضوع از اهمیت ویژه‌ای برخوردار است). این وضعیت با

وضعیتی که در مؤسسات بدون بانک اطلاعاتی کار می‌کنند، متفاوت است. در مؤسسات فاقد بانک اطلاعات، هر برنامه کاربردی فایل‌های خاص خودشان را دارند، گاهی نیز نوارها و دیسک‌های مخصوصی به خود دارند. بنابراین، داده‌ها پراکنده‌اند و کنترل بر روی داده‌ها با روش سیستماتیک دشوار است.

۳-۱. مراحل طراحی بانک اطلاعاتی

طراحی هر سیستم نرم‌افزاری از جمله بانک اطلاعاتی دارای فرآیند منظمی است. در این بخش فرآیند ایجاد یک بانک اطلاعاتی را می‌آموزیم. گام‌های طراحی هر بانک اطلاعاتی در زیر آمده است:

- تعیین کاربرد اصلی بانک اطلاعاتی
- تعیین جداول مورد نیاز بانک اطلاعاتی
- تعیین فیلدهای مورد نیاز جداول
- تعریف روابط بین جداول
- بهینه‌سازی طراحی

۱-۳-۱. تعیین کاربرد اصلی بانک اطلاعاتی

این مرحله به شما کمک خواهد کرد تا نوع اطلاعاتی را که باید از بانک اطلاعاتی استخراج گردد، معین نمایید. در این صورت می‌توانید مشخص کنید که چه موضوعاتی را برای ثبت داده‌ها باید تفکیک نمایید (جداول تشکیل دهنده بانک اطلاعاتی‌تان چه هستند) و داده‌هایی که در جداول قرار می‌گیرند، چه نوع‌اند. فیلدهای تشکیل دهنده جداول‌تان دارای چه نوع هستند. برای نیل به این هدف باید با کاربران بانک اطلاعاتی مصاحبه کنید. فرم‌های اولیه را از آن‌ها دریافت نمایید و تعیین کنید این بانک اطلاعاتی باید به چه نیازهای آن‌ها پاسخ دهد. به عنوان مثال، بانک اطلاعاتی را در نظر بگیرید که برای پخش، خرید و فروش محصولات به کار می‌رود. در این بانک باید به لیستی از پرسش‌های زیر پاسخ داده شود:

۱. در طول ماه یا سال گذشته چه مقدار از کالاهای موجود فروخته شده است (آیا این موجودی‌ها به تفکیک کالا نیاز است)؟
۲. بهترین مشتریان در چه مناطقی زندگی می‌کنند؟
۳. تأمین‌کننده پرفروش‌ترین کالای ما چه شرکتی است؟
۴. چه اطلاعاتی از کالا، تأمین‌کنندگان و مشتریان باید نگه‌داری شود؟
۵. آیا فروش به صورت نسیه انجام می‌شود؟ اگر چنین است، آیا اعتبار هر مشتری با مشتری دیگر متفاوت است یا خیر؟
۶. سؤالات دیگر

۲-۳-۱. تعیین جداول مورد نیاز بانک اطلاعاتی

شاید در طول فرآیند طراحی بانک اطلاعاتی، تعیین جداول یکی از مشکل‌ترین مراحل باشد. چون گزارش‌های قابل چاپی که از بانک اطلاعاتی می‌گیرید و پرسش‌هایی که مایلید پاسخ داده شوند، لزوماً ساختار جداولی که آن‌ها را تولید می‌کنند، تعیین نمی‌کنند. کاربران به شما می‌گویند چه می‌خواهند، اما مشخص نمی‌کنند.

کنند که این اطلاعات چگونه باید در داخل جداول مختلف دسته‌بندی شوند. یعنی، فرم‌هایی که از کاربران دریافت می‌کنید را نمی‌توانید به همین صورت داخل جداول بریزید. زیرا:

۱. ممکن است اطلاعات تکراری وارد شود (افزونگی داده داشته باشید). فرض کنید یک مشتری چند سفارش مختلف داده باشد. اگر برای هر سفارش مشتری، اطلاعاتش از قبیل نام، شماره تلفن و نشانی را دریافت کنید، ممکن است برای یک مشتری اطلاعات تکراری داشته باشید. این عمل علاوه بر ورود داده‌های تکراری و زائد (افزونگی داده)، امکان بروز خطا را در هنگام ورود داده چند برابر خواهد کرد.

۲. فرض کنید نشانی (آدرس) مشتری تغییر یابد. در این صورت باید تمام سفارشات مشتری را پیدا کرده، نشانی او را تغییر دهید. این عمل ممکن است موجب بی‌نظمی شود.

۳. از دست دادن اطلاعات مفید، فرض کنید یک مشتری سفارشی را داده اما بعداً آن را لغو نماید. اگر این سفارش را از جدولی پاک کنید که هم شامل اطلاعات مربوط به خود مشتری و هم اطلاعات سفارشات است، داده‌های مربوط به مشخصات مشتری نیز به همراه داده‌های سفارش حذف خواهد شد. در صورتی که ممکن است مایل باشید مشخصات مشتری جدید را در بانک اطلاعاتی خودتان ثبت کنید تا کاتالوگ محصولات جدید خود را برای او بفرستید.

برای رفع این مشکلات، بهتر است جدول مشتریان را جداگانه در نظر بگیرید. برای این منظور باید جداول بانک اطلاعاتی را نرمال کنید^۱. نرمال‌سازی موجب کاهش افزونگی داده، کاهش بی‌نظمی و کاهش داده‌های Null می‌شود.

بانک اطلاعاتی را در نظر بگیرید که اطلاعات یک انتشارات را نگهداری می‌کند. این بانک اطلاعاتی دارای جداول زیر است (جدول ۱-۱):

- جدول Books، اطلاعات کتاب‌ها از قبیل شابک کتاب، عنوان، تعداد صفحات و غیره را ذخیره می‌کند.
- جدول Publishers، کد ناشر، نام ناشر، شماره تلفن و غیره را برای ناشران نگهداری می‌کند.
- جدول Authors، اطلاعات مؤلفین از قبیل کد مؤلف، نام مؤلف، نام خانوادگی، سن و غیره را نگه‌داری می‌کند.

➤ جدول AutBook، اطلاعات شابک کتاب، مؤلف و مبلغ حق تالیف را نگه‌داری می‌کند.

➤ جدول PubBook، اطلاعات شابک، کد ناشر، تاریخ نشر و حق نشر را نگه‌داری می‌کند.

➤ جدول GroupBook، اطلاعات کد گروه کتاب و نام گروه کتاب را نگه‌داری می‌کند.

➤ ۳-۳-۱. تعیین فیلدهای مورد نیاز بانک اطلاعاتی

➤ برای تعیین فیلدهای یک جدول، باید نوع اطلاعات آن را مشخص کنید. یعنی، داده‌ها، اشخاص، اشیا یا رویدادهای ثبت شده در جدول را بدانید. می‌توانید فیلدها را به‌عنوان مشخصه‌های^۲ (خواص) یک

^۱. نرمال‌سازی جدول، یعنی شکستن یک جدول به چند جدول دیگر. مباحث مربوط به نرمال‌سازی را می‌توانید در کتاب اصول طراحی پایگاه داده از انتشارات فن‌آوری نوین مؤلفین رمضان عباس نژاد ورزی، علیرضا عظیمی و باقر رحیم‌پور کامی ببینید.

^۲. Properties (Attributes)

جدول به حساب آورید. هر رکورد (سطر) در جدول، شامل فیلدها یا مشخصه‌های است. به‌عنوان مثال، فیلد "آدرس" در جدول "مشتریان" شامل آدرس همه مشتریان است. به عبارت دیگر، هر رکورد در جدول مشتریان شامل داده‌هایی درباره یک مشتری است و فیلد آدرس شامل آدرس آن مشتری می‌باشد. در هنگام تعیین فیلدهای جداول به نکات زیر دقت کنید:

جدول ۱-۱ جدول انتشارات.						
نام جدول	نام فیلد	هدف	نوع	کلید اولیه	تهی	کلید خارجی
Books (کتاب‌ها)	ISBN	کد کتاب یا شابک	Varchar(10)	بله	نه	نه
	Title	عنوان کتاب	Varchar(10)	نه	نه	نه
	Page	تعداد صفحات	int	نه	نه	نه
	Price	قیمت کتاب	Decimal(15, 0)	نه	نه	نه
	EditNo	شماره ویرایش	int	نه	بله	نه
	PrintNo	نوبت چاپ	int	نه	بله	نه
	groupID	کد گروه کتاب	Varchar(6)	نه	نه	بله
Publishers (ناشران)	pubID	کد ناشر	Varchar(10)	بله	نه	نه
	pName	نام ناشر	Varchar(50)	نه	نه	نه
	Tel	شماره تلفن ناشر	Varchar(15)	نه	بله	نه
	URL	آدرس وبسایت ناشر	Varchar(100)	نه	بله	نه
	CityName	نام شهر ناشر	Varchar(50)	نه	بله	نه
	bFname	نام مدیر انتشارات	Varchar(20)	نه	نه	نه
	bLname	نام خانوادگی مدیر انتشارات	Varchar(20)	نه	نه	نه
	CountBookPrint	تعداد کتاب‌های چاپ شده	int	نه	نه	نه
Authors (مؤلفان)	atID	کد مؤلف	Varchar(10)	بله	نه	نه
	aFname	نام مؤلف	Varchar(20)	نه	نه	نه
	aLname	نام خانوادگی مؤلف	Varchar(20)	نه	نه	نه
	Age	سن مؤلف	int	نه	نه	نه
	Ranking	مدرک مؤلف	Varchar(30)	نه	بله	نه
	Email	پست الکترونیک مؤلف	Varchar(50)	نه	بله	نه
	Mobile	موبایل مؤلف	Varchar(15)	نه	بله	نه
	SumPayment	مجموع حق تالیف	Decimal(18, 0)	نه	بله	نه
	AutBook (کتاب‌های تالیف شده)	ISBN	کد کتاب	Varchar(10)	بله	نه
atID		کد مؤلف	Varchar(10)	بله	نه	بله
Payment		حق تالیف	Decimal(18, 0)	نه	نه	نه
PubBook (کتاب‌های نشر)	ISBN	شابک	Varchar(10)	بله	نه	بله
	PubID	کد ناشر	Varchar(10)	بله	نه	بله

نه	بله	نه	Varchar(8)	تاریخ نشر	PubDate	یافته)
نه	بله	نه	Decimal(18, 0)	حق نشر	Payment	
نه	نه	نه	Varchar(6)	کد گروه کتاب	groupID	GroupBook (گروه کتاب)
نه	نه	بله	Varchar(30)	نام گروه کتاب	groupName	

۱. هر فیلد را مستقیماً به موضوع آن جدول مربوط کنید، فیلدی که به موضوع جدول دیگری مربوط می‌گردد، جایش در همان جدول است. به عنوان مثال، فیلد "آدرس مشتری" مربوط به جدول مشتریان است. دیگر نیاز نیست در جدول سفارشات در نظر گرفته شود. در مراحل بعدی طراحی بانک اطلاعاتی (هنگامی که رابطه بین جداول را تعریف کنید)، خواهید دید که چگونه می‌توان داده‌های فیلدهای مختلف چندین جدول را با هم ترکیب نمود.

۲. داده‌های حاصل از محاسبات را ذخیره نکنید، یعنی فضای برای فیلدهای مشتق در جدول در نظر نگیرید. به عنوان مثال، در هنگام خرید یا فروش محصولات، چنانچه قیمت و درصد تخفیف را داشته باشید، نیازی نیست مبلغ تخفیف را در جدول نگهداری کنید یا وقتی قیمت و تعداد اقلام کالا را داشته باشید، نیازی نیست مبلغ کل را نگهداری نمایید.

۳. همه اطلاعات لازم را در جدول قرار دهید، برای این منظور کافی است به اطلاعاتی که در مرحله کاربرد اصلی بانک اطلاعاتی جمع‌آوری کرده‌اید، رجوع کرده، به کاغذها، فرم‌ها و گزارش‌ها نگاهی کرده تا مطمئن شوید، تمام اطلاعات را در جدول در نظر گرفته‌اید یا می‌توانید از ترکیب چند فیلد به اطلاعات موردنظر برسید.

۴. اطلاعات را در کوچک‌ترین واحدهای منطقی قرار دهید، ممکن است نام کامل اشخاص (نام و نام خانوادگی) یا آدرس (کشور، استان، شهر، خیابان، کوچه و ...) را در یک فیلد قرار داده باشید. اگر بیش از یک اطلاعات را در یک فیلد قرار دهید، بعداً ایجاد پرس‌وجو بر اساس تک‌تک آن‌ها دشوار خواهد شد. سعی کنید اطلاعات را به قطعات منطقی و مستقل کوچک‌تری بشکنید. یعنی، فیلدهای مرکب در جدول نداشته باشید (اولین شرط نرمال‌سازی فرم یک است).

۵. فیلدهای چند مقداری در جدول نداشته باشید. یعنی اگر جدول‌تان دارای فیلد چند مقداری مانند مدرک تحصیلی است، برای فیلد چند مقداری یک جدول مجزا در نظر بگیرید.

۶. فیلدهای کلید اولیه^۱ را تعیین کنید، باید هر جدول بانک اطلاعاتی دارای یک فیلد یا ترکیب گروهی از فیلدها باشد، که هر رکورد جدول را به صورت منحصر به فرد^۲ (یکتا^۳) مشخص می‌کند. این فیلد معمولاً یک شماره هویت منحصر به فرد (ID) مانند شماره پرسنلی، شماره دانشجویی، کد کالا، شماره ملی و غیره است. اگر برای جدولی نمی‌توانید فیلد کلید اولیه پیدا کنید یا ترکیب همه فیلدهای جدول را به عنوان فیلد کلید اولیه در نظر بگیرید یا فیلدی به جدول اضافه کنید که به عنوان فیلد کلید اولیه جدول باشد. فیلدهای کلید اولیه جداول مختلف بانک اطلاعاتی انتشارات در زیر آمده‌اند:

¹.Primary Key

².Unique

³.Foreign Key

- در جدول **Books**، فیلد ISBN (شابک) کلید اولیه است. زیرا، هیچ دو کتابی شابک یکسان ندارند.
- در جدول **Publishers**، فیلد PubID (کد ناشر) کلید اولیه است. زیرا، هیچ دو ناشری شماره پروانه یکسانی ندارند.
- در جدول **Authors**، فیلد atID (کد مؤلف) کلید اولیه است. چون، هیچ دو مؤلفی کد یکسانی ندارند.
- در جدول **AutBook**، ترکیب فیلدهای ISBN و atID کلید اولیه است.
- در جدول **PubBook**، ترکیب فیلدهای ISBN و PubID کلید اولیه است.
- در جدول **GroupBook**، فیلد groupID کلید اولیه است.

۴-۳-۱. تعریف رابطه‌های بین جداول

همان‌طور که بیان گردید، یک جدول را به چند جدول کوچک‌تر می‌شکنیم تا افزونگی، بی‌نظمی و داده‌های NULL کاهش یابد. به‌عنوان مثال، اطلاعات مشتریان و فروش‌های انجام‌شده در دو جدول مجزا و مستقل ذخیره می‌شوند. اکنون، اگر پرس‌وجویی داشته باشیم که بخواهد اطلاعات را از این دو جدول بازیابی نماید باید ارتباط بین این جداول را برقرار نماییم. برای این منظور، حداقل به یک فیلد مشترک بین این دو جدول نیاز است. این فیلد یا فیلدها باید در یک جدول کلید اولیه یا فرعی و در جدول دیگر کلید خارجی^۳ باشد. سه نوع رابطه را می‌توان بین جداول تعریف کرد:

◆ رابطه یک به چند ◆ رابطه چند به چند ◆ رابطه یک به یک

◆ **رابطه یک به چند**، متداول‌ترین نوع رابطه در یک بانک اطلاعاتی رابطه‌ای است. در یک رابطه یک به چند، یک رکورد (سطر) در یک جدول (نظیر A) می‌تواند بیش از یک رکورد متناظر در جدول دیگر (مانند B) رابطه داشته باشد. ولی یک رکورد در جدول B تنها یک رکورد متناظر در جدول A دارد. به‌عنوان مثال، جدول محصولات و خرید دارای ارتباط یک به چند هستند.

◆ **رابطه چند به چند**، یک رکورد در یک جدول (مانند A) می‌تواند بیش از یک رکورد متناظر در جدول دیگر (نظیر B) باشد و بالعکس.

◆ **رابطه یک به یک**، هر رکورد جدول A نمی‌تواند بیش از یک رکورد متناظر در جدول B داشته باشد و نیز هر رکورد جدول B نمی‌تواند بیش از یک رکورد متناظر در جدول A داشته باشد. به‌عنوان مثال، جدول رانندگان و خودروها را در بانک اطلاعاتی حمل‌ونقل در نظر بگیرید. هر راننده، فقط و فقط راننده یک ماشین است و هر خودرو فقط یک راننده دارد.

- بین جداول **Books** و **AutBook**، فیلد ISBN ارتباط را برقرار می‌کند. در این ارتباط، فیلد ISBN در جدول **Book** کلید اولیه است. اما، ISBN در جدول **AutBook** کلید خارجی است. این ارتباط چند به چند است. چون، یک کتاب می‌تواند چند مؤلف داشته باشد و چند کتاب می‌تواند توسط یک مؤلف تألیف شود.
- بین جداول **Books** و **GroupBook**، فیلد groupID ارتباط را برقرار می‌کند. در این ارتباط، فیلد groupID در جدول **GroupBook** کلید اولیه است، ولی این فیلد در جدول **Books** کلید خارجی می‌باشد. ارتباط بین این دو جدول یک به چند است. زیرا، چند کتاب می‌توانند به یک گروه اختصاص یابند.

➤ فیلد pubID برای برقرار ارتباط بین جداول Publishers و PubBook به کار می‌رود. در این ارتباط، فیلد موجود در جدول Publishers کلید اولیه است. اما، این فیلد در جدول PubBook کلید خارجی است. این ارتباط چند به چند است. زیرا، یک ناشر می‌تواند چند کتاب را انتشار دهد و چند ناشر نیز می‌توانند یک کتاب را با همکاری هم چاپ کنند.

۵-۳-۱. بهینه‌سازی طراحی

هنگامی که جداول، فیلدها و رابطه‌های بین جداول را تعریف کردید، می‌توانید طرح خود را مطالعه و بازنگری کنید تا هر نوع مشکل یا خطا در طراحی ساختار بانک اطلاعاتی را برطرف و تصحیح نمایید. پس از اینکه جداول را ایجاد کرده‌اید و رابطه‌های بین آن‌ها را تعریف نموده‌اید، چند رکورد آزمایشی در جدول وارد کنید تا ببینید آیا بانک اطلاعاتی طراحی شده نیازها و خواسته‌های موردنظران را پاسخ می‌دهد یا خیر. طرح‌های روشن و واضح از فرم‌ها و گزارش‌های موردنظران رسم کرده، ببینید آیا نتایجی که مدنظران است، ارائه می‌دهد یا خیر؟ داده‌های تکراری (افزونگی داده) که در اثر طراحی ناکارآمد ایجاد شده‌اند، را پیدا کرده و حذف کنید. در این مرحله موارد زیر را نیز کنترل نمایید:

➤ آیا هیچ فیلدی را فراموش نکرده‌اید؟

➤ آیا اطلاعاتی وجود دارند که به آن‌ها نیاز داشته باشید؟ ولی در بانک اطلاعاتی جایی برای آن‌ها در نظر گرفته نباشد. اگر چنین است، آیا این اطلاعات به یکی از جداول موجود تعلق دارد؟ وگرنه، شاید لازم باشد، جدول جدیدی ایجاد نمایید.

➤ آیا برای هر جدول یک کلید اولیه مناسب انتخاب کرده‌اید؟ اگر کلید اولیه برای هر جدول انتخاب نموده‌اید، بررسی کنید که هیچ‌گاه نیاز به ورود مقدار تکراری یا NULL برای کلید اولیه نباشد.

➤ چک کنید، آیا پس از ورود چند رکورد تستی در جدول ممکن است حالتی پیش می‌آید که در آن بعضی از مقادیر درون فیلدهای یک جدول به صورت تکراری در رکوردهای مختلف ظاهر شوند؟ اگر چنین وضعیتی ایجاد شد، جدول مذکور را به دو جدول تقسیم کرده، بین آن‌ها یک رابطه یک به چند ایجاد کنید.

➤ آیا جداولی در بانک اطلاعاتی طراحی شده وجود دارند که فیلدهای زیاد و تعداد محدودی رکورد داشته باشند؟ آیا فیلدهای خالی زیادی در این رکوردها وجود دارند؟ اگر چنین وضعیتی وجود دارد، باید ساختار جدول را طوری تغییر دهید که فیلدهای کمتر و رکوردهای بیشتری داشته باشد.

۴-۱. مدل‌های داده‌ای متداول

در این بخش، به برخی از مدل‌های داده متداول و مفید می‌پردازیم و به دنبال نیل دو هدف زیر هستیم:

۱. دانش اضافی درباره ساخت مدل‌های داده؛

۲. الگوهای مدل داده که می‌تواند در شرکت‌تان نیز مفید باشد.

مدل‌های زیر در بازارهای کار متداول هستند:

- | | | |
|--------------------|----------------------|-----------------------|
| ۱. ثبت سفارش مشتری | ۲. کنترل موجودی | ۳. مدیریت سرمایه |
| ۴. مدیریت پروژه | ۵. مدیریت آموزش | ۶. مدیریت منبع انسانی |
| ۷. مدیریت حقوق | ۸. کرایه وسیله نقلیه | |

۱-۴-۱. مدل داده ثبت سفارش مشتری

شرکتی می‌خواهد یک بانک اطلاعاتی را جهت ثبت فعالیت‌های سفارشات مشتریان ایجاد کند. هر مشتری می‌تواند یک یا چند سفارش بدهد که هر سفارش برای یک یا چند محصول می‌باشد. برای هر سفارش یک صورتحساب ارائه می‌شود که می‌تواند به چندین روش مانند چک، کارت اعتباری، یا پول نقد پرداخت گردد. نام کارمندی که در ابتدا سفارش مشتری را پردازش می‌کند، ثبت می‌شود.

کارمند بخش حمل، مسئول بسته‌بندی سفارش و ارسال آن به آدرس مشتری است. اگر محصول سفارشی در انبار موجود نباشد، باید باربری اعزام شود تا ببیند چه چیزی در انبار موجود است. بنابراین، شاید بیش از یک محموله نیاز باشد تا سفارش انجام گیرد. جداول مدل داده ثبت سفارش مشتری در جدول ۱-۲ آمده است. ارتباطات بین جداول در زیر آمده است.

۱. جداول Order و Customer از طریق فیلد customerNo با یکدیگر ارتباط دارند، به طوری که فیلد customerNo در جدول Order کلید خارجی است. اما، در جدول Customer کلید اصلی است.

۲. جداول Order و Employee از طریق فیلد employeeNo با یکدیگر ارتباط دارند، به طوری که فیلد employeeNo در جدول Order کلید خارجی است. اما، در جدول Employee کلید اصلی است.

۳. جداول OrderDetail و Order از طریق فیلد orderNo با یکدیگر ارتباط دارند، به طوری که فیلد orderNo در جدول OrderDetail کلید خارجی است. اما، در جدول Order کلید اصلی است.

۴. جداول OrderDetail و Product از طریق فیلد productNo با یکدیگر ارتباط دارند، به طوری که فیلد productNo در جدول OrderDetail کلید خارجی است. اما، در جدول Product کلید اصلی است.

۵. جداول Shipment و OrderDetail از طریق فیلدهای orderNo و productNo با یکدیگر ارتباط دارند، به طوری که فیلدهای orderNo و productNo در جدول Shipment کلید خارجی است. اما، در جدول OrderDetail کلید اصلی هستند.

۶. جداول Shipment و Employee از طریق فیلد employeeNo با یکدیگر ارتباط دارند، به طوری که فیلد employeeNo در جدول Shipment کلید خارجی است. اما، در جدول Employee کلید اصلی است.

۷. جداول ShipmentMethod و Shipment از طریق فیلد sMethodNo با یکدیگر ارتباط دارند، به طوری که فیلد sMethodNo در جدول Shipment کلید خارجی است. اما، در جدول ShipmentMethod کلید اصلی است.

۱-۴-۲. مدل داده کنترل موجودی

فرض کنید شرکتی می‌خواهد یک بانک اطلاعاتی را برای کنترل موجودی اش ایجاد نماید. این بانک اطلاعاتی شامل تعدادی محصول می‌باشد. این محصول به چندین نوع از قبیل پوشاک، غذا و لوازم التحریر دسته‌بندی می‌گردد.

وقتی محصول به تهیه‌کننده دوباره سفارش داده می‌شود، کارمند سفارش خرید را ایجاد می‌نماید. رکوردهای پیگیری، واحدهای فروخته‌شده و هرگونه ضایعات را تدارکات می‌بیند. جداول مدل داده کنترل موجودی جدول ۱-۳ آمده است.

جدول ۱-۲ جداول ثبت سفارش مشتری.							
کلید خارجی	تهی	کلید اولیه	نوع	هدف	نام فیلد	نام جدول	
نه	نه	بله	int	کد مشتری	customerNo	Customer (مشتری)	
نه	نه	نه	text	نام مشتری	customerName		
نه	نه	نه	text	خیابان مشتری	customerStreet		
نه	نه	نه	text	شهر مشتری	customerCity		
نه	نه	نه	text	استان مشتری	customerState		
نه	نه	نه	text	کد پستی مشتری	customerZipCode		
نه	نه	نه	text	تلفن مشتری	custTelNo		
نه	بله	نه	text	فکس مشتری	custFaxNo		
نه	نه	نه	date	تاریخ تولد	DOB		
نه	نه	نه	bit	وضعیت تأهل	maritalStatus		
نه	نه	نه	text	رتبه اعتبار	creditRating		
نه	نه	بله	int	کد کارمند	employeeNo		Employee (کارمند)
نه	نه	نه	text	عنوان کارمند	title		
نه	نه	نه	text	نام کارمند	firstName		
نه	نه	نه	text	نام خانوادگی کارمند	lastName		
نه	نه	نه	text	آدرس کارمند	Address		
نه	نه	نه	text	تلفن محل کار کارمند	workTelExt		
نه	بله	نه	text	تلفن منزل کارمند	homeTelNo		
نه	بله	نه	text	پست الکترونیکی کارمند	employeeEmailAddress		
نه	نه	نه	text	شماره تأمین اجتماعی	socialSecurityNumber		
نه	نه	نه	date	تاریخ تولد	DOB		
نه	نه	نه	bit	جنسیت کارمند	Sex		
نه	نه	نه	int	حقوق کارمند	Salary		
نه	نه	نه	date	تاریخ شروع به کار کارمند	dateStarted		
نه	نه	بله	int	شماره صورتحساب	invoiceNo	Invoice (صورتحساب)	
نه	نه	نه	date	تاریخ صدور صورتحساب سفارش	dateRaised		
نه	نه	نه	date	تاریخ پرداخت صورتحساب	datePaid		
نه	نه	نه	text	شماره کارت اعتباری	creditCardNo		
نه	نه	نه	text	نام مالک	holdersName		
نه	نه	نه	date	تاریخ انقضاء	expiryDate		
بله	نه	نه	int	کد سفارش	orderNo		

پله	نه	نه	int	شماره روش پرداخت	pMethodNo	
پله	نه	پله	int	کد سفارش	orderNo	Order (سفارش)
نه	نه	نه	date	تاریخ سفارش	orderDate	
نه	نه	نه	text	خیابان صورتحساب	billingStreet	
نه	نه	نه	text	شهر صورتحساب	billingCity	
نه	نه	نه	text	استان صورتحساب	billingState	
نه	نه	نه	text	کد پستی صورتحساب	billingZipCode	
نه	نه	نه	date	تاریخ موعد	promisedDate	
نه	نه	نه	text	وضعیت‌ها	Status	
پله	نه	نه	int	کد مشتری	customerNo	
پله	نه	نه	int	کد کارمند	employeeNo	
نه	نه	پله	int	کد کالا	productNo	
نه	نه	نه	text	نام کالا	productName	
نه	پله	نه	int	کد سریال کالا	serialNo	
نه	پله	نه	int	قیمت هر واحد	unitPrice	
نه	پله	نه	int	موجودی	quantityOnHand	
نه	پله	نه	int		recordLevel	
نه	پله	نه	int		recordQuantity	
نه	پله	نه	datetime		recordLeadTime	
پله	نه	پله	int	کد سفارش	orderNo	OrderDetail (ریز سفارشات)
پله	نه	پله	int	کد کالا	productNo	
نه	نه	نه	int	تعداد سفارش	quantityOrders	
نه	نه	پله	int	کد روش حمل	shipmentNo	Shipment (حمل)
نه	نه	نه	int	تعداد	Quantity	
نه	نه	نه	date	تاریخ حمل	shipmentDate	
نه	نه	نه	text	وضعیت‌های کارمند	employeeStatus	
پله	نه	نه	int	کد سفارش	orderNo	
پله	نه	نه	int	کد کالا	productNo	
پله	نه	نه	int	کد روش حمل	sMethodNo	
پله	نه	پله	int	کد کارمند	employeeNo	
نه	نه	پله	int	کد روش حمل	sMethodNo	ShipmentMethod (روش حمل)
نه	نه	نه	text	روش حمل	shipmentMethod	

ارتباطات بین این جداول در زیر آمده است:

۱. جداول PurchaseOrder و Suppliers از طریق فیلد supplierNo با یکدیگر ارتباط دارند، به طوری که فیلد supplierNo در جدول PurchaseOrder کلید خارجی است. اما، در جدول Suppliers کلید اصلی است.
۲. جداول PurchaseOrder و Employee از طریق فیلد employeeNo با یکدیگر ارتباط دارند، به طوری که فیلد employeeNo در جدول PurchaseOrder کلید خارجی است. اما، در جدول Employee کلید اصلی است.
۳. جداول Transaction و Product از طریق فیلد productNo با یکدیگر ارتباط دارند، به طوری که فیلد productNo در جدول Transaction کلید خارجی است. اما، در جدول Product کلید اصلی است.
۴. جداول PurchaseOrder و Transaction از طریق فیلد purchaseOrderNo با یکدیگر ارتباط دارند، به طوری که فیلد purchaseOrderNo در جدول Transaction کلید خارجی است. اما، در جدول PurchaseOrder کلید اصلی است.

۵. جداول Product و ProductCategory از طریق فیلد purchaseOrderNo با یکدیگر ارتباط دارند، به طوری که فیلد categoryNo در جدول Product کلید خارجی است. اما، در جدول categoryNo کلید اصلی است.

جدول ۱-۳ جداول کنترل موجودی.						
نام جدول	نام فیلد	هدف	نوع	کلید اولیه	تهی	کلید خارجی
Supplier (تهیه کننده)	SupplierNo	کد تهیه کننده	Int	بله	نه	نه
	supplierName	نام تهیه کننده	text	نه	نه	نه
	supplierStreet	خیابان تهیه کننده	text	نه	نه	نه
	supplierCity	شهر تهیه کننده	text	نه	نه	نه
	supplierState	استان تهیه کننده	text	نه	نه	نه
	supplierZipCode	کد پستی تهیه کننده	text	نه	نه	نه
	suppTelNo	تلفن تهیه کننده	text	نه	نه	نه
	suppFaxNo	فکس تهیه کننده	text	نه	بله	نه
	suppEmailAddress	پست الکترونیک تهیه کننده	text	نه	بله	نه
	suppWebAddress	آدرس وبسایت تهیه کننده	text	نه	بله	نه
	contactName	نام نماینده تهیه کننده	text	نه	نه	نه
	contactTelNo	شماره تلفن نماینده	text	نه	نه	نه
	contactFaxNo	فکس نماینده	text	نه	بله	نه
	contactEmailAddress	پست الکترونیک نماینده	text	نه	بله	نه
	paymentTerms	شرایط پرداخت	text	نه	نه	نه
Employee (کارمند)	employeeNo	کد کارمند	int	بله	نه	نه
	title	عنوان کارمند	text	نه	نه	نه
	firstName	نام کارمند	text	نه	نه	نه
	lastName	نام خانوادگی کارمند	text	نه	نه	نه
	Address	آدرس کارمند	text	نه	نه	نه
	workTelExt	تلفن محل کار کارمند	text	نه	نه	نه
	homeTelNo	تلفن منزل کارمند	text	نه	بله	نه
	employeeEmailAddress	پست الکترونیک کارمند	text	نه	بله	نه
	socialSecurityNumber	شماره تأمین اجتماعی	text	نه	نه	نه
	DOB	تاریخ تولد	date	نه	نه	نه
	Sex	جنسیت کارمند	bit	نه	نه	نه
	Salary	حقوق کارمند	int			
dateStarted	تاریخ شروع به کار کارمند	date	نه	نه	نه	
PurchaseOrder (سفارش خرید)	InvoiceNo	کد صورتحساب	int	بله	نه	نه
	recordLeadTime		int	بله	نه	نه
	PurchaseOrderDescription	توضیحات سفارش خرید	date	نه	نه	نه

نه	نه	نه	date	تاریخ سفارش	orderDate		
نه	نه	نه	date	تاریخ موردنیاز	dateRequired		
نه	نه	نه	date	تاریخ ارسال	ShippedDate		
نه	نه	نه	int	هزینه حمل	freightCharge		
بله	نه	نه	int	شماره تأمین کننده	supplierNo		
بله	نه	نه	int	کد کارمند	employeeNo		
بله	نه	بله	int	شماره تراکنش	transactionNo		Transaction (تراکنش)
نه	نه	نه	date	تاریخ تراکنش	transactionDate		
نه	نه	نه	text	توضیحات تراکنش	transactionDescription		
نه	نه	نه	int	قیمت واحد	unitPrice		
نه	نه	نه	int	تعداد سفارش داده شده	unitOrdered		
نه	نه	نه	int	تعداد دریافتی	unitesReceived		
نه	نه	نه	int	تعداد فروخته شده	unitesSold		
نه	نه	نه	int	تعداد برگشتی	unitesWastage		
بله	نه	بله	int	کد کالا	productNo		
بله	نه	بله	int	کد سفارش خرید	purchaseOrderNo		
بله	نه	بله	int	کد کالا	productNo	Product (کالا یا محصول)	
نه	نه	نه	text	نام کالا	productName		
نه	بله	نه	int	شماره سریال کالا	serialNo		
نه	بله	نه	int	قیمت هر واحد	unitPrice		
نه	بله	نه	int	موجودی	quantityOnHand		
					recordLevel		
					recordQuantity		
					recordLeadTime		
نه	بله	بله	int	کد سفارش	orderNo		ProductCategory (ریز سفارشات)
بله	بله	نه	int	کد کالا	productNo		
نه	نه	بله	int	تعداد سفارش	quantityOrders		
نه	نه	بله	int	کد دسته کالا	categoryNo	ProductCategory (دسته کالا)	
نه	نه	نه	text	عنوان دسته کالا	categoryDescription		

۳-۴-۱. مدل داده مدیریت سرمایه

فرض کنید شرکتی می خواهد یک بانک اطلاعاتی را برای نظارت بر هر یک از دارایی هایش از قبیل PCها، پرینترها، اتومبیل ها، میزها، صندلی ها و غیره داشته باشد. دارایی ها به چندین نوع مانند کامپیوترها و اثاثیه ها تقسیم می شوند. هر دارایی به کارمند تخصیص می یابد. طبق قاعده و قانون کارمند بخش مالی هر دارایی را برای مشخص شدن ارزش (قیمت) فعلی آن در بازار بررسی می کند، سپس تاریخ و ارزش فعلی دارایی را ثبت می کند. با توجه به نتیجه ارزیابی، شرکت می تواند درباره فروش دارایی تصمیم گیری کند. همچنین بر اساس قاعده، بر روی هر دارایی تعمیراتی انجام می گیرد. در بعضی موارد تعمیرات به وسیله کارمند انجام می شود، اما در دیگر موارد، دارایی جهت تعمیر به یک شرکت خارجی ارسال می شود. جداول مربوطه به مدل داده مدیریت سرمایه را در جدول ۴-۱ می بینید. ارتباطات بین این جداول در زیر آمده است:

۱. جداول Asset و Employee از طریق فیلد employeeNo با یکدیگر ارتباط دارند، به طوری که فیلد employeeNo در جدول Asset کلید خارجی است. اما، در جدول Employee کلید اصلی است.
۲. جداول Asset و AssetCategory از طریق فیلد assetCategoryNo با یکدیگر ارتباط دارند، به طوری که فیلد assetCategoryNo در جدول Asset کلید خارجی است. اما، در جدول AssetCategory کلید اصلی است.
۳. جداول Asset و Status از طریق فیلد statusNo با یکدیگر ارتباط دارند، به طوری که فیلد statusNo در جدول Asset کلید خارجی است. اما، در جدول Status کلید اصلی است.
۴. جداول Maintenance و Asset از طریق فیلد assetNo با یکدیگر ارتباط دارند، به طوری که فیلد assetNo در جدول Maintenance کلید خارجی است. اما، در جدول Asset کلید اصلی است.
۵. جداول Employee و Maintenance از طریق فیلد employeeNo با یکدیگر ارتباط دارند، به طوری که فیلد employeeNo در جدول Maintenance کلید خارجی است. اما، در جدول Employee کلید اصلی است.
۶. جداول Maintenance و ServiceAgent از طریق فیلد serviceAgentNo با یکدیگر ارتباط دارند، به طوری که فیلد agentNo در جدول Maintenance کلید خارجی است. اما، در جدول ServiceAgent کلید اصلی است.
۷. جداول Valuations و Asset از طریق فیلد assetNo با یکدیگر ارتباط دارند، به طوری که فیلد assetNo در جدول Valuations کلید خارجی است. اما، در جدول Asset کلید اصلی است.
۸. جداول Employee و Valuations از طریق فیلد employeeNo با یکدیگر ارتباط دارند، به طوری که فیلد employeeNo در جدول Valuations کلید خارجی است. اما، در جدول Employee کلید اصلی است.

۴-۴-۱. مدل داده مدیریت پروژه

فرض کنید یک شرکت مشاوره‌ای می‌خواهد یک بانک اطلاعاتی را برای ذخیره اطلاعات مدیریت پروژه‌اش ایجاد نماید. در این بانک اطلاعاتی، هر پروژه برای یک مشتری خاصی می‌باشد و مدیر پروژه مشخصی دارد.

پروژه به تعدادی ماژول‌های کاری تقسیم شده و کارمندان زمان و هزینه هر ماژول کاری را می‌دانند. هر کارمند نقش خاصی دارد که میزان هزینه برای مشتری را تعریف می‌نماید. کارمند می‌تواند به صورت اضافه کار، در چندین ماژول کاری مرتبط با پروژه مشابه کار کند؛ هر ماژول کاری، تعدادی مستندات مرتبط فایل صوتی دارد که هر یک شاید توسط بیش از یک کارمند نوشته شده باشد. جداول مربوط به مدل داده مدیریت پروژه در جدول ۵-۱ آمده است.

ارتباطات بین جداول در زیر آمده است:

۱. جداول Document و WorkPackage از طریق فیلد workPackageNo با یکدیگر ارتباط دارند، به طوری که فیلد workPackageNo در جدول Document کلید خارجی است. اما، در جدول WorkPackage کلید اصلی است.
۲. جداول Employee و Document از طریق فیلد employeeNo با یکدیگر ارتباط دارند، به طوری که فیلد employeeNo در جدول Document کلید خارجی است. اما، در جدول Employee کلید اصلی است.
۳. جداول Document و DocumentAuthor از طریق فیلد documentNo با یکدیگر ارتباط دارند، به طوری که فیلد documentNo در جدول DocumentAuthor کلید خارجی است. اما، در جدول Document کلید اصلی است.
۴. جداول Employee و DocumentAuthor از طریق فیلد employeeNo با یکدیگر ارتباط دارند، به طوری که فیلد employeeNo در جدول DocumentAuthor کلید خارجی است. اما، در جدول Employee کلید اصلی است.

۵. جداول Employee و Role از طریق فیلد roleNo با یکدیگر ارتباط دارند، به طوری که فیلد roleNo در جدول Employee کلید خارجی است. اما، در جدول Role کلید اصلی است.

جدول ۴ - ۱ جداول مدیریت سرمایه.						
نام جدول	نام فیلد	هدف	نوع	کلید اولیه	تهی	کلید خارجی
Asset (مشتري)	assetNo	کد دارایی	int	بله	نه	نه
	assetDescription	عنوان دارایی	text	نه	نه	نه
	serialNo	شماره سریال	int	نه	نه	نه
	dateAcquired	تاریخ دریافت	date	نه	نه	نه
	purchasePrice	استان مشتری	int	نه	نه	نه
	currentValue	ارزش فعلی	int	نه	نه	نه
	dateSold	تاریخ فروش	date	نه	بله	نه
	nextMaintenanceDate	تاریخ تعمیر بعدی	date	نه	بله	نه
	employeeNo	کد کارمند	int	نه	نه	بله
	assetCategoryNo	شماره طبقه بندی دارایی	int	نه	نه	بله
Employee (کارمند)	statusNo	شماره وضعیت	int	نه	نه	بله
	EmployeeNo	کد کارمند	int	بله	نه	نه
	title	عنوان کارمند	text	نه	نه	نه
	firstName	نام کارمند	text	نه	نه	نه
	lastName	نام خانوادگی کارمند	text	نه	نه	نه
	address	آدرس کارمند	text	نه	نه	نه
	workTelExt	تلفن محل کار کارمند	text	نه	نه	نه
	homeTellNo	تلفن منزل کارمند	text	نه	نه	نه
	employeeEmailAddress	پست الکترونیک کارمند	text	نه	نه	نه
	socialSecurityNumber	شماره تأمین اجتماعی	text	نه	نه	نه
	DOB	تاریخ تولد		نه	نه	نه
	Sex	جنسیت کارمند		bit	نه	نه
	Salary	حقوق کارمند		int		
	dateStarted	تاریخ شروع به کار کارمند		date	نه	نه
Valuations (ارزیابی)	valuationNo	کد ارزیابی	int	بله	نه	نه
	valuationDate	تاریخ ارزیابی	date	نه	نه	نه
	valuationPrice	قیمت ارزیابی	int	نه	نه	نه
	assetNo	کد دارایی	Int	نه	نه	بله
	employeeNo	کد کارمند	int	نه	نه	بله
Maintenance (تعمیر)	maintenanceNo	کد تعمیر	int	بله	نه	نه
	maintenanceDescription	توضیحات تعمیر	text	بله	نه	نه
	maintenanceDate	تاریخ تعمیر	date	نه	بله	نه
	maintenanceCost	هزینه تعمیر	int	نه	بله	نه

بله	نه	نه	int	کد دارایی	assetNo	
بله	نه	نه	int	کد نمایندگی	agentNo	
بله	نه	نه	int	کد کارمند	employeeNo	
نه	نه	بله	Int	کد طبقه بندی دارایی	assetcategoryNo	AssetCategory
نه	نه	نه	text	توصیف طبقه بندی دارایی	assetcategoryDescription	(دسته دارایی)
نه	نه	بله	int	کد نمایندگی	agentNo	ServiceAgent (نمایندگی سرویس)
نه	نه	نه	text	نام نمایندگی	agentName	
نه	نه	نه	text	خیابان نمایندگی	agentStreet	
نه	نه	نه	text	شهر نمایندگی	agentCity	
نه	نه	نه	text	استان نمایندگی	agentState	
نه	نه	نه	text	کد پستی نمایندگی	agentZipCode	
نه	نه	نه	text	تلفن نمایندگی	agentTelNo	
نه	بله	نه	text	پست الکترونیک نمایندگی	agentEmailAddress	
نه	بله	نه	text	وبسایت نمایندگی	agentWebAddress	
نه	نه	نه	text	نام نماینده	contactName	
نه	نه	نه	text	تلفن نماینده	contactTelNo	
نه	بله	نه	text	فکس نماینده	contactFaxNo	
نه	بله	نه	text	پست الکترونیکی نماینده	contactEmailAddress	
نه	نه	بله	int	کد وضعیت	statusNo	Status(وضعیت)
نه	نه	نه	text	توصیف وضعیت	statusDescription	

۶. جداول Project و Client از طریق فیلد clientNo با یکدیگر ارتباط دارند، به طوری که فیلد clientNo در جدول Project کلید خارجی است. اما، در جدول Client کلید اصلی است.

۷. جداول Project و Employee از طریق فیلد employeeNo و manageEmployeeNo با یکدیگر ارتباط دارند، به طوری که فیلد manageEmployeeNo در جدول Project کلید خارجی است. اما، فیلد employeeNo در جدول Employee کلید اصلی است.

۸. جداول TimeBooked و workPackage از طریق فیلد workPackageNo با یکدیگر ارتباط دارند، به طوری که فیلد workPackageNo در جدول TimeBooked کلید خارجی است. اما، در جدول workPackage کلید اصلی است.

۹. جداول TimeBooked و Employee از طریق فیلد employeeNo با یکدیگر ارتباط دارند، به طوری که فیلد employeeNo در جدول TimeBooked کلید خارجی است. اما، در جدول Employee کلید اصلی است.

۱۰. جداول Project و WorkPackage از طریق فیلد projectNo با یکدیگر ارتباط دارند، به طوری که فیلد projectNo در جدول WorkPackage کلید خارجی است. اما، در جدول Project کلید اصلی است.

۵-۴-۱. مدل داده مدیریت آموزش

فرض کنید یک شرکت آموزشی می خواهد یک بانک اطلاعاتی داشته باشد که اطلاعات آموزشی را ثبت می نماید. این شرکت چندین سمینار و دوره های آموزشی برگزار می کند. هر آموزش توسط یک عضو پرسنل در بعضی مناطق (مانند سمینار داخلی اتاق ۱۹، هتل و غیره) ارائه می شود. شهریه آموزش ها بر اساس تعداد

نماینده‌هایی که می‌فرستد متغیر است. به‌عنوان مثال، اگر شرکت یک نفر بفرستد، شهریه ۱۰۰۰ دلار و اگر شرکت دو نفر را بفرستد، شهریه اولی ممکن است ۱۰۰۰ دلار باشد. اما، شهریه دومی شاید ۷۵۰ دلار باشد. آموزش می‌تواند با حضور تعدادی از نماینده‌ها باشد که ناظر بر کیفیت آموزش باشد. نماینده می‌تواند به‌عنوان فرد یا از طریق شرکت او ثبت شود. نام کارمندی که نماینده را ثبت می‌کند نیز ثبت می‌گردد. صورتحساب یا از طریق نماینده یا از طریق فرد ارسال می‌شود. جداول مربوط به مدل داده مدیریت آموزش در جدول ۶-۱ آمده است.

جدول ۵-۱ جداول مدیریت پروژه.						
نام جدول	نام فیلد	هدف	نوع	کلید اولیه	تهی	کلید خارجی
Client (مشتری)	clientNo	کد مشتری	int	بله	نه	نه
	clientName	نام مشتری	text	نه	نه	نه
	clientStreet	خیابان مشتری	text	نه	نه	نه
	clientCity	شهر مشتری	text	نه	نه	نه
	clientState	استان مشتری	text	نه	نه	نه
	clientZipCode	کد پستی مشتری	text	نه	نه	نه
	clientTelNo	تلفن مشتری	text	نه	نه	نه
	clientFaxNo	فکس مشتری	text	نه	بله	نه
	clientWebAddress	وبسایت مشتری	text	نه	بله	نه
	contactName	نام نماینده	text	نه	نه	نه
	contactTelNo	تلفن نماینده	text	نه	نه	نه
	contactFaxNo	فکس نماینده	text	نه	بله	نه
	contactMailAddress	پست الکترونیک نماینده	text	نه	بله	نه
	Employee (کارمند)	employeeNo	کد کارمند	int	بله	نه
title		عنوان کارمند	text	نه	نه	نه
firstName		نام کارمند	text	نه	نه	نه
lastName		نام خانوادگی کارمند	text	نه	نه	نه
address		آدرس کارمند	text	نه	نه	نه
workTelExt		تلفن محل کار کارمند	text	نه	نه	نه
homeTelNo		تلفن منزل کارمند	text	نه	بله	نه
employeeEmailAddress		پست الکترونیک کارمند	text	نه	بله	نه
socialSecurityNumber		شماره تأمین اجتماعی	text	نه	نه	نه
DOB		تاریخ تولد		نه	نه	نه
sex		جنسیت کارمند	Bit	نه	نه	نه
salary		حقوق کارمند	int	نه	نه	نه
dateStarted		تاریخ شروع به کار کارمند	date	نه	نه	نه
Document (سند)		documentNo	شماره سند	int	بله	نه
	documentTitle	عنوان سند	text	نه	نه	نه
	documentDate	تاریخ سند	date	نه	نه	نه

نه	نه	نه	text	شماره نسخه سند	versionNo		
بله	نه	نه	int	شماره ماژول کاری	workPackageNo		
بله	نه	نه	int	کد کارمند تصویب کننده	approvedByEmployeeNo		
بله	نه	بله	int	کد پروژه	projectNo	Project (پروژه)	
نه	نه	نه	text	نام پروژه	projectName		
نه	نه	نه	date	تاریخ شروع طراحی	plannedStartDate		
نه	نه	نه	date	تاریخ پایان طراحی	plannedEndDate		
نه	نه	نه	date	تاریخ شروع فعالیت	actionStartDate		
نه	نه	نه	date	تاریخ پایان واقعی	actualEndDate		
نه	نه	نه	int	هزینه پیش بینی شده پروژه	projectedCost		
نه	نه	نه	int	هزینه واقعی پروژه	actualCost		
بله	نه	نه	int	کد مشتری	clientNo		
بله	نه	نه	int	کد مدیر پروژه	manageEmployeeNo		
نه	نه	بله	int	شماره ماژول کاری	workPackageNo		WorkPackage (ماژول کاری)
نه	نه	نه	date	تاریخ شروع برنامه ریزی شده	plannedStartDate		
نه	نه	نه	date	تاریخ پایان برنامه ریزی شده	plannedStopDate		
نه	بله	نه	date	تاریخ شروع واقعی	actualStartDate		
نه	بله	نه	date	تاریخ پایان واقعی	actualEndDate		
نه	نه	نه	int	هزینه پیش بینی شده پروژه	projectedCost		
نه	بله	نه	int	هزینه واقعی پروژه	actualCost		
بله	نه	نه	int	کد پروژه	projectNo		
بله	بله	بله	int	کد نوع هزینه	expenseTypeNo	ExpenseType (نوع هزینه)	
نه	نه	نه	text	نام نوع هزینه	expenseTypeDescription		
نه	نه	بله	int	کد سند	documentNo	DocumentAuthor (مؤلف سند)	
بله	نه	بله	int	کد کارمند	employeeNo		
نه	نه	بله	int	کد ماژول کاری	workPackageNo	TimeBooked	
بله	نه	نه	int	کد کارمند	employeeNo		
نه	نه	نه	date	تاریخ شروع کار	dateStartWork		
نه	نه	نه	date	تاریخ پایان کار	dateStopWork		
نه	نه	نه	int	مدت زمان کار شده	timeWorked		
نه	بله	بله	int	کد نقش	roleNo	Role (نقش)	
نه	نه	نه	text	نام نقش	roleDescription		
نه	نه	نه	int	نرخ صورت حساب	billingRate		

ارتباطات بین جداول در زیر آمده است:

۱. جداول Delegate و Client از طریق فیلد clientNo با یکدیگر ارتباط دارند، به طوری که فیلد clientNo در جدول Delegate کلید خارجی است. اما، در جدول Client کلید اصلی است.
۲. جداول Booking و Location از طریق فیلد locationNo با یکدیگر ارتباط دارند، به طوری که فیلد locationNo در جدول Booking کلید خارجی است. اما، در جدول Location کلید اصلی است.

۳. جداول Booking و Course از طریق فیلد courseNo با یکدیگر ارتباط دارند، به طوری که فیلد courseNo در جدول Booking کلید خارجی است. اما، در جدول Course کلید اصلی است.
۴. جداول Booking و Employee از طریق فیلدهای employeeNo و bookingEmployeeNo با یکدیگر ارتباط دارند، به طوری که فیلد bookingEmployeeNo در جدول Booking کلید خارجی است. اما، فیلد employeeNo در جدول Employee کلید اصلی است.
۵. جداول Course و Employee از طریق فیلد employeeNo با یکدیگر ارتباط دارند، به طوری که فیلد deliverEmployeeNo در جدول Course کلید خارجی است. اما، فیلد employeeNo در جدول Employee کلید اصلی است.
۶. جداول Course و CourseType از طریق فیلد courseTypeNo با یکدیگر ارتباط دارند، به طوری که فیلد courseTypeNo در جدول Course کلید خارجی است. اما، در جدول CourseType کلید اصلی است.
۷. جداول Invoice و Registration از طریق فیلد registrationNo با یکدیگر ارتباط دارند، به طوری که فیلد registrationNo در جدول Invoice کلید خارجی است. اما، در جدول Registration کلید اصلی است.
۸. جداول Invoice و PaymentMethod از طریق فیلد pMethodNo با یکدیگر ارتباط دارند، به طوری که فیلد pMethodNo در جدول Invoice کلید خارجی است. اما، در جدول PaymentMethod کلید اصلی است.
۹. جداول Location و Delegate از طریق فیلد delegateNo با یکدیگر ارتباط دارند، به طوری که فیلد delegateNo در جدول Location کلید خارجی است. اما، در جدول Delegate کلید اصلی است.
۱۰. جداول Location و CourseFee از طریق فیلد courseFeeNo با یکدیگر ارتباط دارند، به طوری که فیلد courseFeeNo در جدول Location کلید خارجی است. اما، در جدول CourseFee کلید اصلی است.
۱۱. جداول Location و Employee از طریق فیلدهای employeeNo و registerEmployeeNo با یکدیگر ارتباط دارند، به طوری که فیلد employeeNo در جدول Location کلید خارجی است. اما، فیلد employeeNo در جدول Employee کلید اصلی است.
۱۲. جداول Location و Course از طریق فیلد courseNo با یکدیگر ارتباط دارند، به طوری که فیلد courseNo در جدول Location کلید خارجی است. اما، در جدول Course کلید اصلی است.

۶-۴-۱. مدل داده مدیریت منابع انسانی

فرض کنید بخش HRM (Human Resource Management) می‌خواهد یک بانک اطلاعاتی را برای نظارت بر کارمندانش ایجاد کند. این شرکت به تعدادی از بخش‌ها تقسیم شده است و کارمندان به یک بخش تخصیص می‌یابند. هر بخش یک مدیر طراحی دارد که مسئولیت کلی بخش و کارمندان موجود در آن بخش بر عهده او می‌باشد. با این وجود، جهت کمک به مدیریت بخش تعدادی از کارمندان برای کمک به گروه‌های پرسنلی کاندید شده‌اند. وقتی که کارمند جدیدی به شرکت اضافه می‌شود، اطلاعات سابقه قبلی او و صلاحیتش مورد نیاز است. بر اساس قاعده هر کارمند نیازمند نظارت دارد که معمولاً توسط مدیر انجام می‌شود. اما شاید توسط نماینده صورت گیرد. این شرکت تعدادی از انواع شغل‌ها از قبیل مدیر، تحلیل‌گر تجارت، فروشنده دارد. جداول مربوط به مدل داده مدیریت منابع انسانی در جدول ۷-۱ آمده است ارتباطات بین جداول در زیر آمده است:

۱. جداول Department و Employee از طریق فیلدهای manageEmployeeNo و employeeNo با یکدیگر ارتباط دارند، به طوری که فیلد manageEmployeeNo در جدول Department کلید خارجی است. اما، فیلد employeeNo در جدول Employee کلید اصلی است.

جدول ۱-۶ جداول مدیریت آموزش.						
نام جدول	نام فیلد	هدف	نوع	کلید اولیه	تهی	کلید خارجی
Client (مشتری)	clientNo	کد مشتری	int	بله	نه	نه
	clientName	نام مشتری	text	نه	نه	نه
	CustomerStreet	خیابان مشتری	text	نه	نه	نه
	clientCity	شهر مشتری	text	نه	نه	نه
	clientState	استان مشتری	text	نه	نه	نه
	clientZipCode	کد پستی مشتری	text	نه	نه	نه
	clientTelNo	تلفن مشتری	text	نه	نه	نه
	custFaxNo	فکس مشتری	text	نه	بله	نه
	clientWebAddress	وبسایت مشتری	text	نه	بله	نه
	contactName	نام نماینده	text	نه	نه	نه
	contactTelNo	تلفن نماینده	text	نه	نه	نه
	contactFaxNo	فکس نماینده	text	نه	بله	نه
	contactMailAddress	پست الکترونیک نماینده	text	نه	نه	نه
	Delegate (نماینده)	delegateNo	شماره نماینده	Int	بله	نه
delegateTitle		عنوان نماینده	text	نه	نه	نه
delegateFName		نام نماینده	text	نه	نه	نه
delegateStreet		خیابان نماینده	text	نه	نه	نه
delegateCity		شهر نماینده	text	نه	نه	نه
delegateState		استان نماینده	text	نه	نه	نه
delegateZipCode		کد پستی نماینده	text	نه	نه	نه
attTelNo		تلفن نماینده	text	نه	نه	نه
attFaxNo		فکس نماینده	text	نه	بله	نه
attEmailAddress		پست الکترونیک نماینده	text	نه	بله	نه
Employee (کارمند)	clientNo	کد مشتری	int	نه	نه	بله
	EmployeeNo	کد کارمند	int	بله	نه	نه
	title	عنوان کارمند	text	نه	نه	نه
	firstName	نام کارمند	text	نه	نه	نه
	lastName	نام خانوادگی کارمند	text	نه	نه	نه
	address	آدرس کارمند	text	نه	نه	نه
	workTelExt	تلفن محل کار کارمند	text	نه	نه	نه
	homeTelNo	تلفن منزل کارمند	text	نه	نه	نه
	employeeEmailAddress	پست الکترونیک کارمند	text	نه	نه	نه
	socialSecurityNumber	شماره تأمین اجتماعی	text	نه	نه	نه

نه	نه	نه		تاریخ تولد	DOB		
نه	نه	نه	Bit	جنسیت کارمند	Sex		
			int	حقوق کارمند	Salary		
نه	نه	نه	date	تاریخ شروع به کار کارمند	dateStarted		
نه	نه	بله	int	شماره سفارش	InvoiceNo	Invoice (سفارش)	
نه	نه	نه	date	تاریخ ثبت سفارش	dateRaised		
نه	نه	نه	date	تاریخ پرداخت سفارش	datePaid		
نه	نه	نه	int	شماره اعتبار کارت	creditCardNo		
نه	نه	نه	text	نام مالک	holdersName		
نه	نه	نه	date	تاریخ انقضاء	expiryDate		
بله	نه	نه	int	کد ثبت	registrationNo		
بله	نه	نه	int	کد روش پرداخت	pMethodNo		
نه	نه	بله	int	کد قرارداد آموزشی	bookingNo	Booking (قرارداد آموزشی)	
نه	نه	بله	date	تاریخ قرارداد آموزشی	bookingDate		
بله	نه	بله	int	کد مکان	locationNo		
بله	نه	بله	int	کد درس	courseNo		
بله	نه	بله	text	کد کارمند ثبت کننده قرارداد	bookingEmployeeNo		
بله	نه	بله	int	کد درس	courseNo	Course (درس)	
بله	نه	بله	text	نام درس	courseName		
نه	بله	نه	text	توصیف درس	courseDescription		
نه	نه	نه	date	تاریخ شروع	startDate		
نه	نه	نه	time	زمان شروع	startTime		
نه	نه	نه	date	تاریخ پایان	endDate		
نه	بله	نه	time	زمان پایان	endTime		
نه	بله	نه	int	تعداد حداکثر نمایندگان	maxDelegates		
نه	بله	نه	bit	تائید شده	Confirmed		
بله	نه	نه	int	کد کارمند تحویل گیرنده	deliverEmployeeNo		
بله	نه	نه	int	کد نوع درس	courseTypeNo		
نه	نه	بله	int	شماره ثبت	registrationNo		Location (مکان)
نه	نه	نه	date	تاریخ ثبت	registrationDate		
بله	نه	نه	int	کد نماینده	delegateNo		
بله	نه	نه	Int	کد هزینه درس	courseFeeNo		
بله	نه	نه	int	کد کارمند ثبت کننده	registerEmployeeNo		
نه	نه	بله	int	کد هزینه درس	courseFeeNo	(هزینه درس) CourseFee	
نه	نه	نه	text	توصیف هزینه	feeDescription		
بله	بله	نه	int	کد درس هزینه	feeCourseNo		

۲. جداول Employee و Department از طریق فیلد departmentNo با یکدیگر ارتباط دارند، به طوری که فیلد departmentNo در جدول Employee کلید خارجی است. اما، فیلد Department در جدول Employee کلید اصلی است.

۳. جداول Employee و Employee از طریق فیلدهای manageEmployeeNo و employeeNo با یکدیگر ارتباط دارند، به طوری که فیلد manageEmployeeNo در جدول Employee کلید خارجی است. اما، فیلد employeeNo در جدول Employee کلید اصلی است.
۴. جداول Grade و positionType از طریق فیلد positionTypeNo با یکدیگر ارتباط دارند، به طوری که فیلد positionTypeNo در جدول Grade کلید خارجی است. اما، در جدول positionType کلید اصلی است.
۵. جداول GradePost و Grade از طریق فیلدهای gradeNo و valiedFromDate با یکدیگر ارتباط دارند، به طوری که فیلدهای gradeNo و valiedFromDate در جدول GradePost کلید خارجی است. اما، فیلدهای gradeNo و valiedFromDate در جدول Grade کلید اصلی است.
۶. جداول GradePost و Post از طریق فیلدهای postNo و availableFromDate با یکدیگر ارتباط دارند، به طوری که فیلدهای postNo و availableFromDate در جدول GradePost کلید خارجی است. اما، فیلدهای postNo و availableFromDate در جدول Post کلید اصلی است.
۷. جداول Posiuiion و Employee از طریق فیلد employeeNo با یکدیگر ارتباط دارند، به طوری که فیلد employeeNo در جدول Posiuiion کلید خارجی است. اما، در جدول Employee کلید اصلی است.
۸. جداول Position و Post از طریق فیلدهای postNo و startDate با یکدیگر ارتباط دارند، به طوری که فیلدهای postNo و startDate در جدول Position کلید خارجی است. اما، فیلدهای postNo و startDate در جدول Post کلید اصلی است.
۹. جداول Post و Department از طریق فیلد departmentNo با یکدیگر ارتباط دارند، به طوری که فیلد employeeNo در جدول Post کلید خارجی است. اما، در جدول Department کلید اصلی است.
۱۰. جداول Qualification و Employee از طریق فیلد employeeNo با یکدیگر ارتباط دارند، به طوری که فیلد employeeNo در جدول Qualification کلید خارجی است. اما، در جدول Employee کلید اصلی است.
۱۱. جداول Qualification و institution از طریق فیلد institutionNo با یکدیگر ارتباط دارند، به طوری که فیلد institutionNo در جدول Qualification کلید خارجی است. اما، در جدول institution کلید اصلی است.
۱۲. جداول Review و Employee از طریق فیلدهای employeeNo و revieweeEmployeeNo با یکدیگر ارتباط دارند، به طوری که فیلد revieweeEmployeeNo در جدول Review کلید خارجی است. اما، فیلد employeeNo در جدول Employee کلید اصلی است.
۱۳. جداول Review و Employee از طریق فیلدهای employeeNo و reviewerEmployeeNo با یکدیگر ارتباط دارند، به طوری که فیلد reviewerEmployeeNo در جدول Review کلید خارجی است. اما، فیلد employeeNo در جدول Employee کلید اصلی است.

جدول ۱-۷ جداول مدیریت منابع انسانی.						
نام جدول	نام فیلد	هدف	نوع	کلید اولیه	تهی	کلید خارجی
Department (بخش)	departmentNo	کد بخش	int	بله	نه	نه
	departmentName	نام بخش	text	نه	نه	نه
	depLocation	مکان بخش	text	نه	نه	نه
	manageEmployeeNo	کد مدیر بخش	int	نه	نه	بله
Grade (نمره)	gradeNo	کد نمره	int	بله	نه	نه
	validFromDate	تاریخ شروع اعتبار نمره	date	بله	نه	نه
	validToDate	تاریخ پایان اعتبار نمره	date	بله	نه	نه
	gradeDescription	توصیف نمره	text	نه	نه	نه
	gradeSalary	حقوق نمره	int	نه	نه	نه
	noDayLeave	تعداد روز غیبت	text	نه	نه	نه
	positionTypeNo	کد نوع جایگاه	int	نه	نه	بله
	EmployeeNo	کد کارمند	int	بله	نه	نه
Employee (کارمند)	Title	عنوان کارمند	text	نه	نه	نه
	firstName	نام کارمند	text	نه	نه	نه
	lastName	نام خانوادگی کارمند	text	نه	نه	نه
	Address	آدرس کارمند	text	نه	نه	نه
	workTelExt	تلفن محل کار کارمند	text	نه	نه	نه
	homeTelNo	تلفن منزل کارمند	text	نه	بله	نه
	employeeEmailAddress	پست الکترونیکی کارمند	text	نه	بله	نه
	socialSecurityNumber	شماره تأمین اجتماعی	text	نه	نه	نه
	DOB	تاریخ تولد			نه	نه
	Sex	جنسیت کارمند	Bit	نه	نه	نه
	Salary	حقوق کارمند	int	نه	نه	نه
	departmentNo	کد بخش	int	نه	نه	بله
	dateStarted	تاریخ شروع به کار کارمند	date	نه	نه	نه
	dateLeft	تاریخ ترک به کار کارمند	date	نه	نه	بله
	supervisorEmployeeNo	کد کارمندی مدیر	int	نه	نه	نه
	GradePost (نمره پست)	gradeNo	کد نمره	int	بله	نه
validFromDate		تاریخ شروع اعتبار	date	بله	نه	نه
postNo		کد پست	int	بله	نه	نه
availableFromDate		تاریخ شروع دسترسی	date	بله	نه	نه
Institution (سازمان)	institutionNo	کد سازمان	int	بله	نه	نه
	institutionName	نام سازمان	text	نه	نه	نه
	instTelNo	تلفن سازمان	text	نه	نه	بله