



دولت جمهوري اسلامي افغانستان  
اداره تعليمات تخنيكي و مسلکي  
معاونيت امور اکادميک  
رياست نصاب و تربيه معلم

## مدیریت دیتا

رشته: کمپیوتر ساینس - دیپارتمنت: دیتابیس  
صنف ۱۴ - سمستر اول

سال: ۱۳۹۹ هجری شمسی



## شناسنامه کتاب

نام کتاب: مدیریت دیتا  
رشته: کامپیوتر ساینس  
تدوین کننده: پوهندوی شعیب زرین خیل  
همکار تدوین کننده: پوهنمل اسدالله محمدی

- کمیته نظارت: ندیمه سحر رئیس اداره تعلیمات تخنیکي و مسلکی
- عبدالحمید اکبر معاون امور اکادمیک اداره تعلیمات تخنیکي و مسلکی
- حبیب الله فلاح رئیس نصاب و تربیه معلم
- عبدالمتمین شریفی آمر انکشاف نصاب تعلیمی، ریاست نصاب و تربیه معلم
- روح الله هوتک آمر طبع و نشر کتب درسی، ریاست نصاب و تربیه معلم
- احمد بشیر هیله من مسؤل انکشاف نصاب، پروژه انکشاف مهارت های افغانستان
- محمد زمان پویا کارشناس انکشاف نصاب، پروژه انکشاف مهارت های افغانستان
- علی خیبر یعقوبی سرپرست مدیریت عمومی تألیف کتب درسی، ریاست نصاب و تربیه معلم

- کمیته تصحیح: دوکتور محمد یونس طغیان ساکایی
- محمد علی شهاب
- محمد امان هوشمند مدیر عمومی بورد تصحیح کتب درسی و آثار علمی

دیزاین: صمد صبا و سید کاظم کاظمی  
سال چاپ: ۱۳۹۹ هجری شمسی  
تیراژ: ۱۰۰۰  
چاپ: اول  
وبسایت: [www.tveta.gov.af](http://www.tveta.gov.af)  
ایمیل: [info@tveta.gov.af](mailto:info@tveta.gov.af)

حق چاپ برای اداره تعلیمات تخنیکي و مسلکی محفوظ است.



## سرود ملی

دا وطن افغانستان دی	دا عزت د هر افغان دی
کور د سولې کور د تورې	هر بچی یې قهرمان دی
دا وطن د ټولو کور دی	د بلوڅو، د ازبکو
د پښتون او هزاره وو	د ترکمنو، د تاجکو
ورسره عرب، ګوجر دي	پامیریان، نورستانیان
براهوي دي، قزلباش دي	هم ایماق، هم پشه یان
دا هیواد به تل ځلېږي	لکه لمر پر شنه آسمان
په سینه کې د آسیا به	لکه زړه وی جاویدان
نوم د حق مو دی رهبر	وايو الله اکبر وايو الله اکبر



## پیام اداره تعلیمات تخنیکي و مسلکي

استادان نهایت گرامی و محصلان ارجمند!

تربیت نیروی بشری ماهر، متخصص و کارآمد از عوامل کلیدی و انکارناپذیر در توسعه اقتصادی و اجتماعی هر کشور محسوب می‌گردد و هر نوع سرمایه‌گذاری بزرگ در بخش‌های مختلف اقتصادی نیازمند به پلان‌گذاری و سرمایه‌گذاری در بخش نیروی بشری و توسعه منابع این نیرو می‌باشد. بر مبنای این اصل و بر اساس فرمان شماره ۱۱ مقام عالی ریاست جمهوری اسلامی افغانستان به تاریخ ۱۳۹۷/۲/۱ اداره تعلیمات تخنیکي و مسلکي از بدنه وزارت معارف مجزا و فصل جدیدی در بخش عرضه خدمات آموزشی در کشور گشوده شد. اداره تعلیمات تخنیکي و مسلکي به‌عنوان متولی و مجری آموزش‌های تخنیکي و مسلکي در کشور محسوب می‌شود که در چارچوب استراتژی ۵ ساله خویش دارای چهار اولویت مهم که عبارت‌اند از افزایش دسترسی عادلانه و مساویانه فراگیران آموزش‌های تخنیکي و مسلکي در سطح کشور، بهبود کیفیت در ارائه خدمات آموزشی، یادگیری مادام‌العمر و پیوسته و ارائه آموزش نظری و عملی مهارت‌ها به‌طور شفاف، کم‌هزینه و مؤثر که بتواند نیاز بازار کار و محصلان را در سطح محلی، ملی و بین‌المللی برآورده کند، می‌باشد. این اداره که فراگیرترین نظام تعلیمی کشور در بخش تعلیمات تخنیکي و مسلکي است، تلاش می‌کند تا در حیطه وظایف و صلاحیت خود زمینه دستیابی به هدف‌های تعیین‌شده را ممکن سازد و جهت رفع نیاز بازار کار، فعالیت‌های خویش را توسعه دهد.

نظام اجتماعی و طرز زندگی در افغانستان مطابق به احکام دین مقدس اسلام و رعایت تمامی قوانین مشروع و معقول انسانی عیار است. اداره تعلیمات تخنیکي و مسلکي جمهوری اسلامی افغانستان نیز با ایجاد زمینه‌های لازم برای تعلیم و تربیت جوانان و نوجوانان مستعد و علاقه‌مند به حرفه‌آموزی، ارتقای مهارت‌های شغلی در سطوح مختلف مهارتی، تربیت کادرهای مسلکي و حرفوی و ظرفیت‌سازی تخصصی از طریق انکشاف و ایجاد مکاتب و انستیتوت‌های تخنیکي و مسلکي در سطح کشور با رویکرد ارزش‌های اسلامی و اخلاقی فعالیت می‌نماید.

فلذا جهت نیل به اهداف عالی این اداره که همانا تربیه افراد ماهر و توسعه نیروی بشری در کشور می‌باشد؛ داشتن نصاب تعلیمی بر وفق نیاز بازار کار امر حتمی و ضروری بوده و کتاب درسی یکی از ارکان مهم فرایند آموزش‌های تخنیکي و مسلکي محسوب می‌شود، پس باید همگام با تحولات و پیشرفت‌های علمی نوین و مطابق نیازمندی‌های جامعه و بازار کار تألیف و تدوین گردد و دارای چنان ظرافتی باشد که بتواند آموزه‌های دینی و اخلاقی را توأم با دست‌آوردهای علوم جدید با روش‌های نوین به محصلان انتقال دهد. کتابی را که اکنون در اختیاردارید، بر اساس همین ویژگی‌ها تهیه و تدوین گردیده است.

بدین‌وسیله، صمیمانه آرزومندیم که آموزگاران خوب، متعهد و دلسوز کشور با خلوص نیت، رسالت اسلامی و ملی خویش را ادا نموده و نوجوانان و جوانان کشور را به‌سوی قله‌های رفیع دانش و مهارت‌های مسلکي رهنمایی نمایند و از محصلان گرامی نیز می‌خواهیم که از این کتاب به‌درستی استفاده نموده، در حفظ و نگهداشت آن سعی بلیغ به خرج دهند. همچنان از مؤلفان، استادان، محصلان و اولیای محترم محصلان تقاضا می‌شود نظریات و پیشنهادات خود را در مورد این کتاب از نظر محتوا، ویرایش، چاپ، اشتباهات املائی، انشایی و تایپی عنوانی اداره تعلیمات تخنیکي و مسلکي کتباً ارسال نموده، امتنان بخشد.

در پایان لازم می‌دانیم در جنب امتنان از مؤلفان، تدوین‌کنندگان، مترجمان، مصححان و تدقیق‌کنندگان نصاب تعلیمات تخنیکي و مسلکي از تمامی نهادهای ملی و بین‌المللی که در تهیه، تدوین، طبع و توزیع کتب درسی زحمت‌کشیده و همکاری نموده‌اند، قدردانی و تشکر نمایم.

ندیمه سحر

رئیس اداره تعلیمات تخنیکي و مسلکي جمهوری اسلامی افغانستان

ح	.....	مقدمه
---	-------	-------

## فصل اول: اساسات سیستم مدیریت دیتابیس ارتباط (Fundamentals of Relational DBMS) ..... ۱

۲	.....	اهمیت دیتا	۱.۱
۳	.....	مدیریت دیتا	۱.۲
۴	.....	انواع دیتا	۱.۳
۵	.....	مدل های دیتا	۱.۴

## فصل دوم: استفاده از سیستم مدیریت دیتابیس ارتباط ..... ۱۸

۲۱	.....	ایجاد دیتابیس در سیکویل	۲.۱
۲۴	.....	ایجاد جدول ها و ساختمان های دیگر در سیکویل	۲.۲
۲۷	.....	ایجاد رابطه ها بین جدول ها در سیکویل	۲.۳
۲۹	.....	داخل کردن، تغییر آوردن و پاک کردن دیتا در سیکویل	۲.۴
۲۹	.....	داخل کردن دیتا به دیتابیس	۲.۴.۱
۳۱	.....	پاک کردن دیتا از جدول ها در سیکویل	۲.۴.۲

## فصل سوم: ترانزکشن ها (Transactions) در دیتابیس ..... ۳۷

۳۹	.....	ترانزکشن چیست؟ (What is a Transaction?)	۳.۱
۴۰	.....	حالت های یک ترانزکشن	۳.۱.۱
۴۲	.....	خصوصیات یک ترانزکشن	۳.۱.۲
۴۳	.....	شریک ساختن دیتا و کنترل ترانزکشن دیتا با چندین استفاده کننده	۳.۲
۴۳	.....	استفاده از ترانزکشن (Use of Transaction)	۳.۳
۴۶	.....	پایان ترانزکشن	۳.۳.۱
۴۶	.....	نقص ها یا خرابی ها در ترانزکشن (Transaction Failure Types)	۳.۴
۴۸	.....	سیتهای عملیه های شامل ترانزکشن	۳.۵
۴۹	.....	مراحل انزواء و ناهنجاری های ممکنه (Different Levels of Isolation and Possible Anomalies)	۳.۶
۵۰	.....	استفاده از ترانزکشن در سیکویل	۳.۷

## فصل چهارم: مسایل پیشرفته در سیستم های مدیریت دیتابیس های ارتباط ..... ۵۶

۵۷	.....	اندکس ها (Indexes)	۴.۱
۵۸	.....	اندکس اولیه (Primary Index)	۴.۱.۱
۵۹	.....	اندکس دومی (Secondary Index)	۴.۱.۲
۵۹	.....	اندکس کلستری (Clustering Index)	۴.۱.۳
۶۰	.....	استفاده از اندکس ها در تنظیم دیتا	۴.۲



استفاده از طرز العمل اندکس در سیکویل .....	۴.۳
امنیت و کنترل دسترسی به سیستم (Security and Access Control) .....	۴.۴
انواع ایمنی (Security) در سیستم‌های مدیریت دیتابیس .....	۴.۴.۱
تهدیدها به دیتابیس‌ها (Threats to Databases) .....	۴.۵
کنترل دسترسی (Access Control) .....	۴.۶
کنترل‌های برداشت و نتیجه‌گیری (Inference Control)، جریان و رمزگذاری دیتا .....	۴.۷
ایجاد حساب‌های استفاده‌کنندگان .....	۴.۸
ایجاد حساب مدیران دیتابیس .....	۴.۸.۱
ایجاد و حذف حساب‌های استفاده‌کنندگان عادی .....	۴.۸.۲

## فصل پنجم: مدل‌های دیتا (Data Models) ..... ۷۸

فارمت XML (eXtensible Markup Language) .....	۵.۱
مفهوم کلمه Markup چه است؟ .....	۵.۱.۱
نحوه نوشتن کد XML – XML Syntax) .....	۵.۲
اعلامیه XML – XML Declaration) .....	۵.۲.۱
قواعد نحوی برای نوشتن اعلامیه XML .....	۵.۲.۲
تگ‌ها و عناصر XML .....	۵.۲.۳
قواعد نحوی برای تگ‌ها و عناصر XML .....	۵.۲.۴
مشخصه‌های XML – XML Attributes) .....	۵.۲.۵
قواعد نحوی برای مشخصه‌های XML .....	۵.۲.۶
ریفرنس‌های XML – XML References) .....	۵.۲.۷
سندهای XML – XML Documents) .....	۵.۳
متن (Text) .....	۵.۴
انواع دیتای متنی BINARY و VARBINARY .....	۵.۴.۱
انواع دیتای متنی BLOB و TEXT .....	۵.۴.۲
نوع دیتای متنی ENUM .....	۵.۴.۳
مولتی‌مدیا (Multimedia) .....	۵.۵
مولتی‌مدیای خطی و غیر خطی (Linear and Nonlinear Multimedia) .....	۵.۶
عناصر مولتی‌مدیا (تصویرها) .....	۵.۶.۱
عناصر مولتی‌مدیا (صدا و ویدیو) .....	۵.۶.۲

## فصل ششم: دیتای روی وب (Data on the Web) ..... ۱۰۸

ترکیب دیتا (Data Integration) .....	۶.۱
سطوح ترکیب دیتا (Data Integration Levels) .....	۶.۱.۱
مفاهیم کلیدی در قسمت ترکیب دیتا .....	۶.۲
ترکیب دیتا به اهداف تهیه معلومات احصائیوی و مدیریتی .....	۶.۲.۱
پیوند دقیق و پیوند احتمالی .....	۶.۲.۲
ارزیابی کیفیت ترکیب دیتا .....	۶.۲.۳

۱۱۳.....	سناریوهای ترکیب دیتا.....	۶.۲.۴
۱۱۵.....	ترکیب فیزیکی و ترکیب مجازی دیتا.....	۶.۳
۱۱۸.....	استفاده از نماها (Views).....	۶.۴
۱۲۱.....	بازیافتن معلومات (Information Retrieval).....	۶.۵
۱۲۱.....	یک سیستم بازیافتن معلومات.....	۶.۵.۱
۱۲۲.....	بازیافتن معلومات و یا بازیافتن دیتا.....	۶.۶
۱۲۵.....	کیوری‌های ساختاری روی وب (Structured Queries over the Web).....	۶.۷
۱۲۵.....	مدل استخراج سکیم (The Schema Extraction Model).....	۶.۷.۱
۱۲۷.....	مدل کیوری متنی (The Text Query Model).....	۶.۷.۲
۱۲۸.....	مدل استخراج دیتا برای دیتابیس (The Extraction Database Model).....	۶.۷.۳

### فصل هفتم: انبار کردن دیتا (Data Warehousing) و بهره‌برداری از دیتا (Data Mining)..... ۱۳۳

۱۳۴.....	انبار کردن دیتا (Data Warehousing).....	۷.۱
۱۳۵.....	پروژه‌های دیزاین انبار دیتا (Data Warehouse Design Process).....	۷.۱.۱
۱۳۷.....	ساختمان‌های انبار دیتا (Data Warehouse Architectures).....	۷.۱.۲
۱۳۸.....	ساختمان سه طبقه‌یی انبار دیتا.....	۷.۱.۳
۱۴۱.....	بهره‌برداری از دیتا (Data Mining).....	۷.۲
۱۴۱.....	هدف بهره‌برداری از دیتا (دیتامیننگ).....	۷.۲.۱
۱۴۲.....	وظایف دیتامیننگ.....	۷.۲.۲
۱۴۳.....	ساختار دیتامیننگ (Architecture of Data Mining).....	۷.۳
۱۴۵.....	طبقه‌بندی سیستم‌های دیتامیننگ.....	۷.۴
۱۴۵.....	تقسیم‌بندی سیستم‌های دیتامیننگ به اساس نوع دیتابیس‌ها.....	۷.۴.۱
۱۴۶.....	تقسیم‌بندی سیستم‌های دیتامیننگ به اساس نوع معلومات.....	۷.۴.۲
۱۴۶.....	تقسیم‌بندی سیستم‌های دیتامیننگ به اساس تکنیک‌ها.....	۷.۴.۳
۱۴۶.....	تقسیم‌بندی سیستم‌های دیتامیننگ به اساس نوع اپلیکیشن‌های استفاده شده.....	۷.۴.۴
۱۴۶.....	مقایسهٔ انبار دیتا با بهره‌برداری از دیتا.....	۷.۵

### مأخذ (References)..... ۱۵۱

امروزه تنظیم، ذخیره و استفاده از دیتا و معلومات به کمک کمپیوتر، یکی از اصول اساسی و مهم در پیشبرد امور زندگی انسان‌ها به شمار می‌رود. معلومات در امور تعلیمی، کارهای تحقیقی، تجارت، ارتباطات و غیره به مثابه اولین معیار کار، دارای اهمیت به‌سزایی است. به هر اندازه‌یی که معلومات مهم است به همان اندازه تنظیم و مدیریت دیتا و معلومات دارای اهمیت خاصی می‌باشد. بر اساس معلومات نادرست و ناتکمیل، امکان اشتباه در نتایج متوقعه بیش‌تر می‌گردد. با پیشرفت تکنالوژی و فناوری، معلومات به کمک کمپیوترها و برنامه‌های کمپیوتری به نام دیتابیس‌ها به صورت اساسی تحلیل، ذخیره و مدیریت می‌شود.

جمع‌آوری، ذخیره و استفاده معقول از معلومات یگانه راز موفقیت در کارها و تصمیم‌گیری‌ها در موارد مختلف شمرده می‌شود. معلومات در موارد خاص، نیاز به تحلیل دارد. زمانی که معلومات تحلیل و به اجزای کوچک‌تر دسته‌بندی شود، همین دسته‌های اجزای تحلیل‌شده معلومات به نام دیتا یا عناصر دیتا یاد می‌شوند. عناصر دیتا با در نظر داشت بعضی قواعد استندرد، دوباره ترکیب شده و راپورهای معلومات را تشکیل می‌دهند.

عناصر دیتا و معلومات در منابع گوناگون به طریقه‌های مختلف ذخیره شده و قابل دسترس می‌باشد. دیتای ذخیره‌شده در منابع مختلف دوباره قابل استفاده بوده و از آن راپورهای غنی تهیه می‌گردد. این کتاب تحت عنوان مدیریت دیتا (Data Management) تهیه گردیده است. موضوعات مهم تخنیکی از مراجع معتبر و جدید اکادمیک جمع‌آوری و در تهیه مواد کتاب از آن استفاده گردیده است. کتاب دارای هفت فصل بوده و در هر فصل یکی از موضوعات تخنیکی به ترتیب گنجانیده شده است. در شروع فصل‌ها، هدف عمومی و اهداف آموزشی در ارتباط با موضوع فصل گنجانیده شده تا یک راهنمای خوب برای استفاده‌کنندگان و محصلان گرامی باشد. سؤالات تخنیکی و فعالیت‌های لازم برای محصلان نیز به ترتیب در اخیر هر فصل، گنجانیده شده است.

در فصل اول مسایل عمومی چون سیستم‌های مدیریت دیتابیس ارتباطی توضیح داده شده است. در این فصل موضوعاتی چون اهمیت دیتا، مدیریت دیتا، انواع دیتا و مودل‌های دیتا به صورت عمومی بررسی شده است. این موضوعات بیش‌تر شکل تکراری داشته و به خاطر ارتباط با موضوع مورد نظر در آغاز فصل کتاب گنجانیده شده است. تفصیل بیش‌تر موضوعات ذکر شده در این فصل، در یکی دیگر از مضامین رشته دیتابیس تحت عنوان "اساسات دیتابیس" در صنوف پایین‌تر توسط محصلان عزیز فرا گرفته شده است. آخرین مبحث در فصل اول عبارت از سیستم مدیریت دیتابیس ارتباطی است. مثال‌های لازم در محتویات فصل اول برای آموزش بیش‌تر نیز در نظر گرفته شده است.

در فصل دوم به چگونگی استفاده از سیستم مدیریت دیتابیس ارتباطی پرداخته شده است. در این فصل نحوه استفاده عملی دستورهای سیکویل شرح شده و سپس محصلان با دستورالعمل‌های ابتدایی هم‌چون



کارکردن با دیتابیس آشنایی پیدا می‌کنند. قسمت دوم فصل، کارکردن با دیتا در دیتابیس‌های ارتباطی را در بر داشته و در اخیر فصل دوم نحوه حذف جدول‌ها و دیتابیس‌ها توضیح داده شده است.

در فصل سوم کتاب موضوع ترانزکشن‌ها (Transactions) در دیتابیس مورد بحث و بررسی قرار گرفته است. توضیح و حالت‌های یک ترانزکشن، خصوصیات ترانزکشن، شریک‌ساختن دیتا و کنترل ترانزکشن با چندین استفاده‌کننده از جمله عناوین قسمت‌های اول، فصل سوم است. نتیجه عملیه‌های شامل یک ترانزکشن در دو حالت تأیید و یا رد به کمک پایان ترانزکشن، موضوع بعدی در سلسله عناوین فصل بوده، نقیصه‌ها و خرابی‌های ممکنه در یک ترانزکشن و ست‌های عملیه‌های شامل ترانزکشن نیز به ترتیب در محتوای فصل سوم گنجانیده شده است. بعضی مسایل تخنیکی از قبیل مراحل انزواء و ناهنجاری‌های ممکنه در ترانزکشن توضیح گردیده است؛ به علاوه در قسمت‌های پایانی فصل، استفاده عملی از ترانزکشن در سیکویل با مثال‌های عملی نشان داده شده است.

محتویات فصل چهارم را سیستم‌های مدیریت دیتابیس‌های ارتباطی، تشکیل می‌دهد. اندکس‌ها و چگونگی استفاده از اندکس‌ها در دیتابیس‌ها و بیرون از دیتابیس‌ها با مثال‌های مناسب شرح شده است. دستور ایجاد و استفاده از اندکس در سیکویل توضیح و به صورت عملی نشان داده شده است. امنیت و کنترل دسترسی به سیستم، مسایل امنیتی دیتابیس‌ها، تهدیدهای ممکنه در دیتابیس‌ها، کنترل دسترسی به سیستم و غیره توضیح گردیده است. کنترل حساب‌های استفاده‌کنندگان دیتابیس از قبیل ایجاد حساب‌های مدیران دیتابیس، حساب‌های استفاده‌کنندگان عادی، تنظیم صلاحیت‌ها به انواع مختلف حساب‌ها و همچنان حذف حساب‌های موجود از جمله عناوین این فصل کتاب است. مثال‌های تطبیقی در دستورهای کنترولی نیز به صورت عملی کار شده و نتایج آن‌ها نشان داده شده است.

موضوع مورد بحث فصل پنجم عبارت از مدل‌های دیتا است که قابلیت ذخیره و استفاده در دیتابیس‌ها را دارند. فارمت XML من حیث یکی از فارمت‌هایی که در کمپیوترساینس بیش‌تر مورد استفاده قرار می‌گیرد، به بحث گرفته شده و توضیحات لازم در زمینه، ارائه شده است. نوع دیگری از مدل دیتا که در دیتابیس‌ها بیش‌تر استفاده می‌شود عبارت از نوع متن (Text) می‌باشد. جزئیات لازم در قسمت ذخیره و استفاده از متن در دیتابیس‌ها توضیح و با مثال‌های مناسب روشن گردیده است. مولتی‌میدیا عبارت از نوع دیگری از مدل‌های دیتای قابل استفاده در دیتابیس بوده که در ادامه فصل عناوین مربوطه آن توضیح گردیده است.

در فصل ششم کتاب موضوع جالب دیگری، تحت عنوان دیتا روی وب توضیح و ارائه می‌شود. عناوین مهمی که در این فصل گنجانیده شده است، به سه قسمت تقسیم می‌شود. در قسمت اول عناوین فرعی عبارت از سطوح ترکیب دیتا، مفاهیم کلیدی در ترکیب دیتا، ترکیب دیتا به اهداف توسعه معلومات، انواع پیوندها در ترکیب دیتا، ارزیابی کیفیت ترکیب دیتا و سناریوهای ترکیب دیتا می‌باشد. در قسمت دوم فصل موضوعاتی چون بازیافتن معلومات روی سیستم، تفکیک اصطلاح بازیافتن معلومات در تناسب با اصطلاح بازیافتن دیتا توضیح داده شده است؛ و بالاخره در قسمت سوم این فصل عناوینی در چوکات کیوری‌های ساختاری روی وب

گنجانیده شده است. همچنین در این بخش مدل‌های استخراج سکیم‌ها، کیوری متنی و استخراج (دیتا) برای دیتابیس نیز توضیح شده است.

بلاخره در فصل هفتم به حیث فصل پایانی، بر دو موضوع تمرکز صورت گرفته است. یکی انبار دیتا (Data Warehousing) و دیگری بهره‌برداری از دیتا (Data Mining) می‌باشد. عناوین فرعی مربوط به انبار دیتا عبارت از پروسه‌های دیزاین انبار دیتا و ساختمان‌های انبار دیتا می‌باشد. در بخش بهره‌برداری از دیتا یا دیتاماینینگ، عناوین فرعی شامل هدف بهره‌برداری از دیتا، وظایف دیتاماینینگ، ساختار دیتاماینینگ و طبقه‌بندی سیستم‌های دیتاماینینگ می‌شوند. برای وضاحت بیش‌تر موضوعات یاد شده، هر دو عنوان عمومی (انبار دیتا و بهره‌برداری از دیتا) با هم مقایسه و فوایدشان توضیح داده شده است.



## هدف کلی کتاب

آشنایی با ترانزکشن ها، مدل های دیتا در وب، انبار کردن دیتا و استخراج دیتا.

# فصل اول

## اساسات سیستم مدیریت دیتابیس ارتباط (Fundamentals of Relational DBMS)



**هدف کلی:** محصلان با اساسات سیستم مدیریت دیتابیس ارتباط (Relational Database Management System) به اختصار RDBMS آشنا شوند.

**اهداف آموزشی:** در پایان این فصل محصلان قادر خواهند شد تا:

- اهمیت دیتا، مدیریت دیتا و سیستم‌های مدیریت دیتا را تشریح نمایند.
- انواع مختلف دیتا، مدل‌های دیتا و سیستم‌های مدیریت دیتا را توضیح دهند.
- اساسات Relational DBMS مانند ساختار DBMS را شرح دهند.

دیتابیس عبارت از ثبت و استفادهٔ معلومات بر اساس قواعد استاندارد در ساختمان‌هایی به نام ارتباط است. هر رابطه در دیتابیس به شکل دیتای دو بُعدی تنظیم و به صورت جدول نمایش داده می‌شود. بخش‌های مختلف دیتابیس با هم در ارتباط بوده و به اساس قوانین استاندارد عمل می‌کند. قوانین استاندارد در قسمت دیتابیس‌ها تحت نام مدل ارتباط (The Relational Model) در دههٔ ۸۰ قرن نوزدهم میلادی مطرح شده و تا فعلاً که در سال ۲۰۲۰ قرار داریم، قابل تطبیق و مورد قبول است. ایجاد، انکشاف و استفادهٔ دیتابیس‌ها توسط برنامه‌یی به نام (Database Management System) یا سیستم مدیریت دیتابیس صورت می‌گیرد.

سیستم مدیریت دیتابیس ارتباطات (RDBMS) مجموعه‌یی از برنامه‌های کمپیوتری بوده که این برنامه‌ها، استفاده‌کننده را قادر می‌سازد تا دیتابیس را ایجاد و استفاده نماید. سیستم‌های مدیریت دیتابیس برای استفاده‌کنندگان مختلف، زمینهٔ انجام پروسه‌های تعریف، انجام‌دادن و شریک‌ساختن دیتابیس‌ها را فراهم نموده و کار با اپلیکیشن‌های مورد ضرورت را زمینه‌سازی می‌کند. تعریف دیتابیس شامل مشخصات انواع دیتا، ساختمان‌ها و محدودیت‌های دیتای قابل ذخیره در دیتابیس است. معلومات تفصیلی مربوط بخش‌های سیستم و ساختمان یک دیتابیس نیز توسط RDBMS در کتلاک دیتابیس که به نام دیکشنری نیز یاد می‌شود، ذخیره می‌گردد. این نوع معلومات به نام میتادیتا (Metadata) یاد شده و معمولاً از دید استفاده‌کنندگان عادی پنهان است.

## ۱.۱ اهمیت دیتا

دیتا عبارت از مجموعهٔ حقایق و ارقام پیرامون یک موضوع است. توسط همین مجموعه یا با ترکیب چندین بخش از مجموعهٔ دیتا، معلومات در مورد موضوع مشخص تهیه می‌شود. به گونهٔ مثال در یک جدول نام اشخاص، مشخصات تذکره و وظیفه آن‌ها ذکر شده است. در جدول دیگر، نوع وظیفه و مقدار معاش تعیین شده برای هر وظیفه مشخص شده است. در جدول دومی، فرضاً برای معلم ۱۲۰۰۰ افغانی، برای مدیر ۸۰۰۰ افغانی و برای مأمور ۶۰۰۰ افغانی ثبت شده است. به کمک این دو جدول معلومات، مقدار معاش اشخاص معلوم می‌گردد. جدول ۱ دیتای اشخاص را نشان می‌دهد، جدول ۲ دیتای معاشات را در بر دارد و جدول ۳ معلومات معاش هر شخص را به صورت دقیق نشان می‌دهد.

شماره	نام اشخاص	نمبر تذکره	وظیفه
1	احمد	123	معلم
2	محمود	234	مدیر
3	جواد	567	معلم
4	نجیم	678	مأمور
5	حامد	345	مأمور
6	اسد	456	معلم

جدول ۱-۱ دیتا مربوط اشخاص و وظیفه

شماره	وظیفه	معاش
1	معلم	12000
2	مدیر	1000
3	مامور	8000

جدول ۱-۲ دیتا مربوط نوع وظیفه و مقدار معاش آن

شماره	نام اشخاص	معاش
1	احمد	12000
2	محمود	1000
3	جواد	12000
4	نجیم	8000
5	حامد	8000
6	اسد	12000

جدول ۱-۳ معلومات اشخاص و مقدار معاش آن‌ها گرفته شده از دیتای جدول‌های شماره ۱ و ۲

طوری که در مثال بالا دیده می‌شود، با استفاده از دیتای جدول‌های شماره ۱ و ۲ معلومات در جدول شماره ۳ تهیه شده است. در صورت اشتباه در ترکیب مجموعه دیتا، امکان استخراج معلومات نادرست به وجود می‌آید. هم‌چنان اگر دیتا نادرست باشد، امکان نشان دادن معلومات نادرست در راپور نهایی بیش‌تر می‌شود. پس از این‌جا گفته می‌توانیم که دیتای درست و دقیق دارای اهمیت به‌سزایی است. راپورهایی که به اساس دیتا به شکل خودکار تهیه شده و در بخش‌های مهم استفاده می‌شوند، در چنین راپورها معلومات نظر به درجه مهم بودن موضوع تنظیم و دسته‌بندی می‌شود. به طور مثال معلومات در مورد پرواز طیاره‌ها در یک میدان هوایی اگر به اساس دیتای درست تنظیم نشود، امکان سقوط یا تصادم طیاره‌ها با هم بسیار بالا می‌باشد. پس گفته می‌توانیم که معلومات به خاطر دقت و درست بودن آن از اهمیت به‌سزایی برخوردار است. اشتباه در دیتا باعث اشتباه در معلومات شده و همین اشتباهات امکان تلفات جانی، مالی، حقوقی و غیره را به دنبال خواهد داشت.

## ۱.۲ مدیریت دیتا

طوری که در عنوان قبلی روی اهمیت دیتا بحث صورت گرفت، اینک روی مدیریت دیتا توضیحاتی ارائه می‌شود. مدیریت دقیق دیتا باعث پیش‌برد اهداف به سمت درست شده و معلومات مؤثر و کارا را در اختیار استفاده‌کنندگان قرار می‌دهد. در سال‌های پیش، قبل از انکشاف کمپیوترهای الکترونیکی، دیتا و معلومات به شکل فایل‌های چاپ‌شده و نوشته‌شده ذخیره می‌شد. چاپ اسناد به کمک ماشین‌های میخانیکی تایپنگ به نام "Typewriter" صورت می‌گرفت. بعد از به میان آمدن کمپیوترهای رقمی و برقی، قابلیت ذخیره و پروسس دیتا و معلومات به شکل فایل‌های رقمی (دیجیتال) به وجود آمد. البته این سهولت تکنالوژی من حیث راکار مترادف به خاطر ذخیره دیتا و معلومات به کار گرفته می‌شد. دیتای رقمی به نام کاپی نرم



(Soft Copy) و دیتای چاپی و نوشته شده به نام (Hard Copy) کاپی سخت یاد می شود. مدیریت دیتا در هر دو شکل کاپی سخت و کاپی نرم سهولت استفاده مؤثر دیتا و معلومات را فراهم می سازد.

مدیریت دیتای سخت عبارت از تنظیم کتابخانه های سنتی (Traditional)، ترتیب معلومات در کتاب ها، مجلات و دیگر آثار چاپ شده می باشد. معلومات کتاب ها به اساس فهرست عناوین، فهرست جدول ها، فهرست اشکال، صفحه گذاری، ذکر مآخذ و غیره را می توان یکی از مثال های خوب تنظیم دیتا نامید. به عین ترتیب آثار چاپ شده با معیارهای مشخص و ترتیب محتوای آنها نیز مستقیماً متأثر از مدیریت دیتا است. ترتیب و تنظیم کتاب ها در کتابخانه های سنتی به اساس بخش های علوم، اتاق ها، الماری های و ردیف ها همه و همه با در نظر گرفتن ترتیب حروف الفبا، مثال های دیگر مدیریت دیتای سخت بوده می تواند. به کمک مدیریت دیتا از بین ده ها هزار جلد کتاب یک کتابخانه، پیدا کردن یک کتاب مشخص کاری بسا آسان بوده که در چند دقیقه محدود یک استفاده کننده می تواند کتاب یا کتاب های مورد نظرش را پیدا نماید.

مدیریت دیتای نرم (Soft) مسئله مهم دیگری بوده که ذخیره، تنظیم و استفاده دیتا در کمپیوتر را شامل می شود. بعد از به میان آمدن کمپیوترهای برقی، دیتا و معلومات به صورت فارمت فایل ها مورد استفاده قرار گرفتند. ذخیره، تغیر، پاک کردن و استفاده مجدد از فایل ها در عصر تکنالوژی یکی از گام های بزرگ ترقی به شمار می رود. بعد از این انکشافات، نظریه ها و تجربه های جدیدی در زمینه طرح و انکشاف دیتا به عمل آمد. انکشاف دیتابیس ها و سیستم های مدیریت دیتا، وسایل مهم در این عرصه به شمار رفته و تنظیم دیتا از طریق آنها با سهولت های مهمی همراه است.

### ۱.۳ انواع دیتا

دیتا عبارت از معلومات تحلیل شده به اساس مفاهیم مشخص است. معلومات در مورد یک موضوع به صورت همه جانبه روی کاغذ آورده شده و بعداً بخش های معین آن تحلیل و دسته بندی می شود. در ساحة دیتابیس و مدیریت دیتا، شناختن و توضیح انواع دیتا حائز اهمیت است. به گونه مثال، معلومات در مورد یک مکتب اگر در نظر گرفته شود، به خاطر شناسایی و استخراج انواع دیتا، همین معلومات باید تحلیل و حلاجی گردد. مثال یاد شده بر اساس معلومات پایین عمل خواهد کرد.

مکتب الف شاگردان را از صنف یک الی صنف دوازده تعلیم می دهد. معلمین از هر دو طبقه ذکور و اناث در مکتب الف تدریس می نمایند. سن شاگردان از ۵ سال الی ۲۱ سال بوده می تواند. آدرس بودوباش شاگردان باید معلوم و معلومات اقارب شان موجود باشد. معلمین در مقابل تدریس معاش و سایر امتیازات اخذ می کنند. جملات این پاراگراف معلومات را در مورد مکتب الف در بر دارد. اگر این معلومات تحلیل شود از آن دیتا به دست می آید. دیتای به دست آمده می تواند برای ایجاد و انکشاف یک دیتابیس به کار رود؛ یعنی به خاطر ایجاد یک دیتابیس معلومات باید موجود بوده و همین معلومات تحلیل شود.

به خاطر تحلیل معلومات در مورد مکتب الف، یک تحلیل کننده ضرورت به دانستن طریقه های معیاری تحلیل دیتا دارد. در طریقه های معیاری هر دسته دیتا باید دارای یک نوعیت باشد. نوعیت دیتا زمینه ذخیره و استفاده

آن را در دیتابیس از طریق سیستم مدیریت دیتابیس مهیا می‌سازد. انواع دیتایی که در دیتابیس از طریق کمپیوتر ذخیره و استفاده می‌شود عبارت از متن، نمبر، واحد پولی، تاریخ و غیره بوده می‌تواند. دیتایی که از معلومات مکتب الف استخراج شود شامل انواع ذیل خواهد بود:

- شاگرد (نام شاگرد از نوع متن، نام پدر شاگرد از نوع متن، شماره شناسایی از نوع نمبر، آدرس از نوع متن و نمبر، فیس تعلیم از نوع واحد پولی، پول ترانسپورت از نوع واحد پولی، پول البسه از نوع واحد پولی، پول قرطاسیه از نوع واحد پولی، شماره تماس از نوع نمبر و سمبول، سن از نوع نمبر، ...)
- معلم (نام معلم از نوع متن، آدرس از نوع متن و نمبر، جنس از نوع ذکور یا اناث، نمبر تذکره از نوع نمبر، ایمیل از نوع متن و سمبول، صنوف تدریس از نوع متن و نمبر، معاش از نوع واحد پولی، ...)
- صنف (سال از نوع سال، شماره از نوع نمبر، نگران از نوع متن، فعال از نوع بلی/نخیر، ...)
- مضمون (نام از نوع متن، سال از نوع سال، معلم از نوع متن، ...).

به این ترتیب معلومات بر اساس جزئیات آن مورد تحلیل قرار گرفته و به هر اندازه‌یی که تفصیل بیش‌تر در معلومات باشد به همان اندازه تحلیل با جزئیات صورت می‌گیرد. معلومات تحلیل شده به نام دیتا یاد می‌شود. هر دسته‌یی از دیتا دارای یک نوع مشخص است. همین نوعیت دیتا، افراد تخنیکی را اجازه می‌دهد تا دیتابیس‌ها را ایجاد نموده و در اختیار استفاده‌کنندگان قرار دهند. به صورت نمونه، انواع دیتایی که قابل استفاده در دیتابیس باشند، در شکل پایین نشان داده شده است.

Field Name	Data Type
ID	AutoNumber
FirstName	Text
LastName	Text
Email	Text
Zip	Text
Username	Memo
Newsletter	Number
	Date/Time
	Currency
	AutoNumber
	Yes/No
	OLE Object
	Hyperlink
	Attachment
	Calculated
	Lookup Wizard...

شکل ۱-۱ انواع دیتا (Data types) قابل استفاده در دیتابیس مایکروسافت اکسس

## ۱.۴ مدل‌های دیتا

مدل‌های دیتا به منظور نشان دادن ساختمان منطقی دیتابیس‌ها استفاده می‌شوند. مدل‌های دیتا عبارت از معلومات اساسی به خاطر معرفی ساختمان عمومی دیتابیس‌ها بوده که این معلومات، ساختار کلی دیتابیس را با استفاده از سمبول‌های معیاری به شکل خلاصه نشان می‌دهد. بر علاوه، مدل‌های دیتا ارتباط بین بخش‌های یک دیتابیس را نشان داده و نیز شیوه عمل و ذخیره دیتا را در داخل سیستم نشان می‌دهد. مدل هموار (Flat) دیتا را می‌توانیم ابتدایی‌ترین مدل دیتا بنامیم. در هم‌چو مدل‌ها تمام بخش‌های دیتا روی یک سطح نشان داده می‌شود. مدل‌های سابقه دیتا، بیش‌تر ساده بوده و مسایل و ارتباطات پیچیده را به صورت واضح بیان



از یک نوع باشند، یعنی دارای یک دومین باشند. مشخصه‌ها دارای نوع دیتا (Data-Type) و Property های مشخص است. نوع دیتا و Property یک مشخصه، در مرحله اول دیزاین دیتابیس تعیین می‌شود. مثال‌های این نوع دیتا برای مشخصه‌ها عبارت از Date, Numeric, Character, Currency ... اند. Property شرایطی را برای داخل کردن دیتا به مشخصه‌ها وضع و تعیین می‌نماید تا یک مشخصه به صورت خودکار (Default) کدام دیتا را داشته باشد؟ آیا قابلیت گرفتن عناصر نامعلوم (Null) را داشته باشد و یا خیر؟

طوری که گفته شد مشخصه‌ها دارای یک دومین‌اند. دومین عبارت از ست عناصر ممکنه برای یک مشخصه است؛ مثلاً برای یک مشخصه‌یی به نام Gender که جنس را تعیین می‌نماید، صرف دو حرف M و F و یا هم دو کلمه ذکور و اناث تعیین می‌شود و عبارت دومین همان مشخصه است؛ و یا هم برای مشخصه‌یی به نام Title، کلماتی از قبیل Mrs. Mr. و Mss. عبارت از دومین است. یک دومین توسط چندین مشخصه، شریک یا Share شده می‌تواند. به طور مثال، آدرس‌های Student و Professor، می‌توانند از عین دومین یا گروه آدرس‌های ممکنه، استفاده نمایند.

مشخصه‌ها نظر به ساختمان‌شان به دو نوع‌اند:

- مشخصه ترکیبی (Composite Attribute)

- مشخصه ساده (Simple Attribute)

یک مشخصه ترکیبی عبارت از مشخصه‌یی است که بتواند به چندین مشخصه دیگر تقسیم و یا تجزیه شود؛ مثلاً «تحصیل» می‌تواند به مشخصه‌های دوره ابتدایه، لیسه، لیسانس، ماستری و دکتری تجزیه شود.

یک مشخصه ساده نمی‌تواند به اجزای کوچک‌تر تجزیه شود. مثال‌های این نوع مشخصه‌ها عبارت‌اند از: حالت مدنی، جنسیت، سن و غیره.

مشخصه‌ها نظر به داشتن محتوا به دو نوع‌اند:

- یک قیمته (Single Value)

- چندین قیمته (Multi Value)

مشخصه‌های یک قیمته دارای دیتای معین برای هر جزء می‌باشند؛ یعنی برای هر Entity، صرف یک جزء دیتا موجود است؛ مانند نمبر تذکره اشخاص، نمبر متوالی اجناس و غیره. مشخصه‌های چندین قیمته می‌توانند دارای دیتای مختلف باشند؛ مثلاً یک شخص می‌تواند چندین نمبر تلفون با فارمت‌های مختلف (موبایل، دفتر، خانه، ...) داشته باشد و یا تاریخ‌های مختلف از قبیل شمسی، قمری و یا میلادی در یک مشخصه استفاده شود.

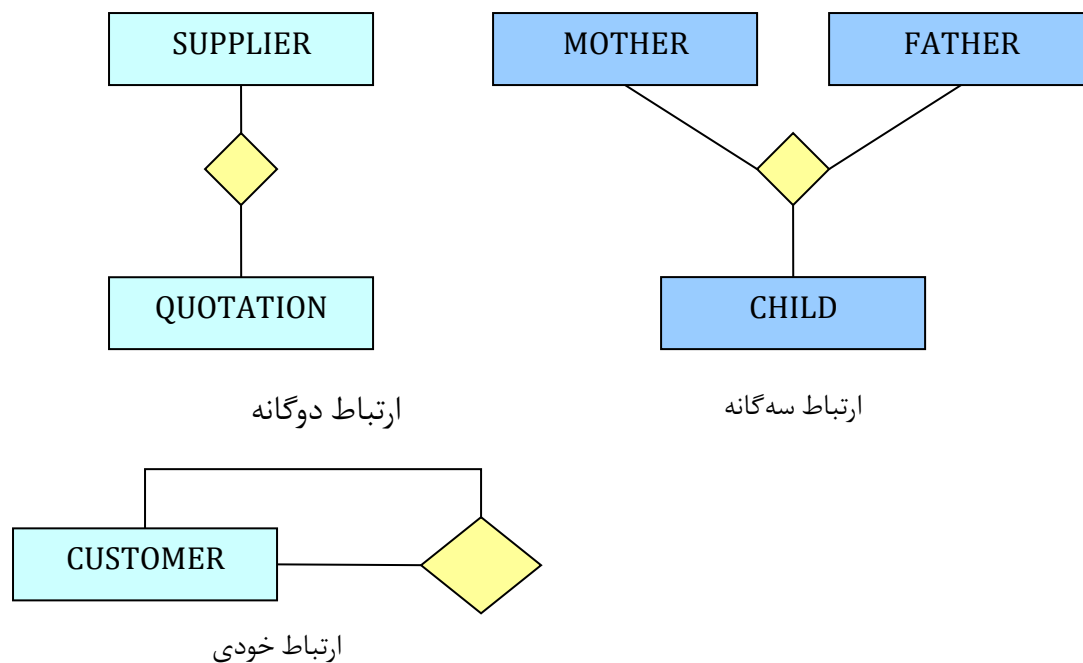
- **شناسه‌ها (Identifiers):** Entity-Class ها دارای شناسه یا Identifier اند. شناسه‌ها عبارت از مشخصه‌هایی‌اند که یک Entity-Instance را شناسایی یا Identify می‌کنند. طور مثال Entity-Instance به نام EMPLOYEE می‌تواند توسط Social-Security-Number و یا توسط EmployeeNumber و یا هم توسط EmployeeName شناسایی شود.

شناسه اول Entity-Instance، امکان دارد از یک مشخصه تشکیل شود و یا هم از تعداد بیش‌تر مشخصه‌ها ترکیب شود. شناسه‌یی که از دو و یا اضافه از دو مشخصه به صورت ترکیبی تشکیل شده باشد، به نام شناسه ترکیبی (Composite-Identifier) یاد می‌شود. مثال‌های شناسه ترکیبی عبارت‌اند از (AreaCode, LocalNumber)، (ProjectName, TaskName)، (FirstName, LastName, Phone) و غیره.

یک شناسه می‌تواند دیتای تکراری و یا هم غیر تکراری را قبول نماید. اگر امکان تکرار دیتا در یک شناسه نباشد (دارای دیتای Unique باشد)، محتوای آن تنها یک Entity-Instance را شناسایی می‌نماید. در صورت قبول دیتای تکراری توسط شناسه، گروهی از Entity-Instance ها توسط آن شناسایی می‌شوند؛ مثلاً EmployeeNumber، معمولاً یک شناسه از نوع یک (با دیتای غیر تکراری) بوده و EmployeeName با وجود بودن شناسه، می‌تواند دیتای تکراری را هم قبول نماید.

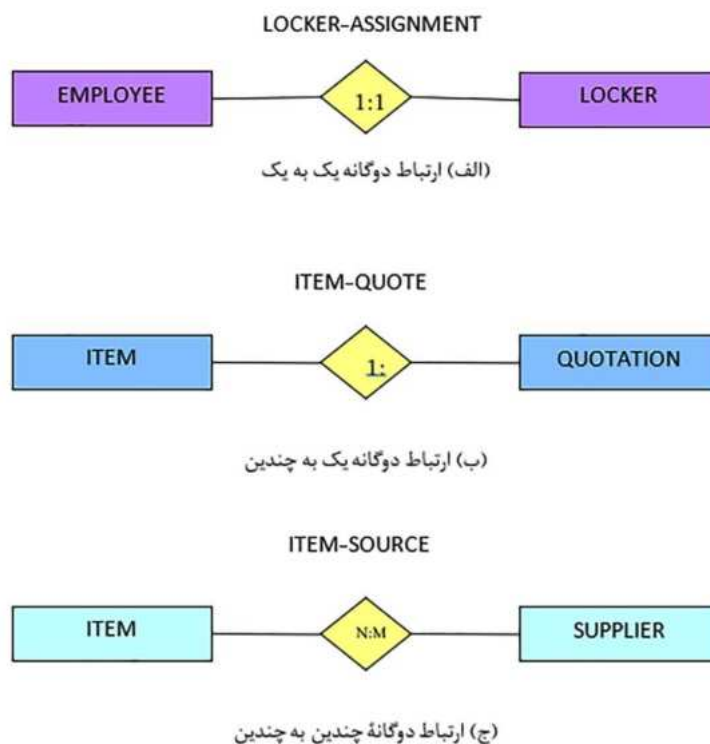
با در نظر گرفتن توضیحات بالا، می‌توان گفت که یک شناسه در دیتامدل E-R مشابه به کلید اولیه (Primary-Key) در مدل ارتباط (Relational-Model) است. البته بعضی تفاوت‌ها بین آن‌ها وجود دارد، از آن جمله می‌توان به برخی موارد اشاره کرد. اول به طور کلی شناسه، به مفهوم منطقی یا Logic آن در مدل E-R استفاده می‌شود؛ یعنی نام مشخصه یا مشخصه‌هایی که به حیث شناسه استفاده می‌شوند، مفهوم مکمل یک Entity را برای استفاده‌کنندگان افاده می‌نماید. دوم این‌که کلیدهای اولیه و کلیدهای کاندیدی (Candidate-Keys) حتماً باید دارای دیتای غیر تکراری (Unique) باشند. در حالی که شناسه‌ها هر دو امکانات را دارا می‌باشند، یعنی قابلیت گرفتن دیتای تکراری را نیز دارا می‌باشند.

- **ارتباط‌ها (Relationships):** Entity ها را می‌توان با ایجاد ارتباط‌ها یا Relationship ها، با یک‌دیگر در ارتباط قرارداد. در مدل E-R دو نوع ارتباط وجود دارد. یکی Relationship-Class و دیگر Relationship-Instance. اولی به خاطر ارتباط Entity-Class ها و دومی به خاطر ارتباط Entity-Instance ها استفاده می‌شود. در Entity-Class امکان ارتباط چندین Entity-Class به یک‌دیگر وجود دارد؛ یعنی اضافه از دو Entity هم می‌توانند در یک ارتباط سهیم باشند. تعداد Entity-Class ها در یک ارتباط عبارت از درجه ارتباط است. شکل پایین ارتباط‌ها را از درجه 2، درجه 3 و ارتباط خودی (Recursive Relationship) نشان می‌دهد.



شکل ۳-۱ مثال‌های از ارتباط‌ها

ارتباط‌های دوگانه می‌توانند به سه شکل ایجاد شوند: ارتباط یک با یک (1:1)، ارتباط یک با چندین (1:N) و ارتباط چندین با چندین (N:M). شکل پایین انواع ارتباط‌های دوگانه را به گونه مثال‌ها نشان می‌دهد.



شکل ۴-۱ سه مثال از ارتباط‌های دوگانه



در جزء (الف) شکل بالا، یعنی ارتباط یک به یک (1:1)، Entity-Instance از یک نوع به Entity-Instance از نوع دیگر ارتباط دارد.

در ارتباط شکل (الف) (LOCKER-ASSIGNMENT) یک EMPLOYEE به یک LOCKER و برعکس، ارتباط برقرار می‌کند. پس هیچ EMPLOYEE نمی‌تواند اضافه از یک LOCKER داشته باشد و هیچ LOCKER نمی‌تواند با اضافه از یک EMPLOYEE ارتباط بگیرد.

جزء (ب) شکل بالا، نوع دوم، ارتباط دوگانه را نشان می‌دهد. این نوع ارتباط عبارت از یک به چندین (1:N) است. در این ارتباط یعنی (ITEM-QUOTE)، یک عنصر از ITEM به چندین QUOTATION ارتباط داشته، یعنی دارای چندین QUOTATION بوده، اما یک QUOTATION تنها می‌تواند به یک عنصر از ITEM ارتباط بگیرد.

جزء (ج) شکل بالا نوع سوم، ارتباط دوگانه را نشان می‌دهد. در این حالت (چندین به چندین یا (N:M)، چندین عنصر از هر دو جانب در ارتباط سهم گرفته می‌توانند. در مثال ذکر شده (ITEM-SOURCE)، یک ITEM می‌تواند توسط چندین شخص Supply شود و یک Supplier می‌تواند چندین ITEM را تهیه نماید.

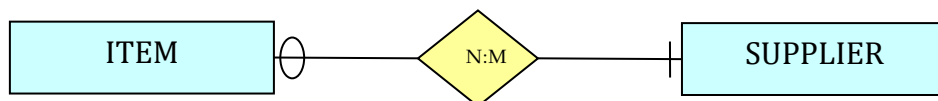
وقتی به شکل بالا دقت شود، دیده می‌شود که سمبول 1 و N که به جانب Entity قرار دارند، نشان دهنده سهم گرفتن Entity در ارتباط است؛ مثلاً در جزء (ب) شکل بالا، سمبول عدد "1" به سمت ITEM نوشته شده است و نشان می‌دهد که یک عنصر از ITEM در ارتباط سهم می‌گیرد و سمبول "N" نشان دهنده سهم گرفتن چندین QUOTATION در ارتباط است. اگر موقعیت سمبول‌های "1" با "N" عوض شود، در نتیجه چندین ITEM دارای یک QUOTATION و یک QUOTATION دارای چندین ITEM خواهد بود. پس موقعیت سمبول‌ها در ارائه مدل E-R مهم است.

- **کاردینالی (Cardinality):** کاردینالی حد اصغری و حد اعظمی را برای یک Entity به خاطر سهم گرفتن در یک ارتباط تعیین می‌نماید؛ به عبارت دیگر، هر تناسبی که یک Entity بتواند در ارتباط سهیم شود، عبارت از کاردینالی همان عنصر در ارتباط یا Relationship است. کمترین کاردینالی یا حد اصغری کاردینالی به اشکال مختلف نشان داده می‌شود. از جمله برای نمایش حد اقل یک Entity یک خط عمودی در کنار Entity به روی خط ارتباط ترسیم می‌شود و برای نمایش سهم گرفتن حد اقل صفر Entity، یعنی موجودیت Entity در ارتباط اختیاری باشد، یک بیضوی کوچک به شکل عمودی در کنار Entity به روی خط ارتباط دیاگرام آن ترسیم می‌شود.

به شکل پایین دقت شود، یک ITEM باید با حد اقل یک SUPPLIER ارتباط یا Relationship داشته باشد، اما یک SUPPLIER مجبور نیست که حتماً با یک ITEM ارتباط برقرار نماید؛ به عبارت

دیگر، یک ITEM کم‌ترین کاردینالی یک و بیش‌ترین کاردینالی چندین را با SUPPLIER دارا است. یک SUPPLIER کم‌ترین کاردینالی صفر و بیش‌ترین کاردینالی چندین را با ITEM دارد.

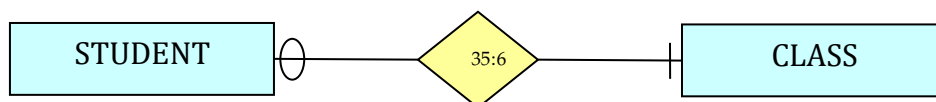
#### ITEM-SOURCE



شکل ۵-۱ ارتباط‌ها با کم‌ترین کاردینالی

- در مثال دیگر فرض شود، یک محصل بتواند 1 الی 6 صنف را در جریان یک سمسטר تعقیب نماید و تعداد محصلان در یک صنف از صفر الی 35 نفر بوده می‌توانند. شکل پایین دیاگرام مربوطه را نشان می‌دهد. کاردینالی اصغری "1" و اعظمی "6" برای محصل و به عین ترتیب کاردینالی اصغری «صفر» و اعظمی "35" برای صنف تعیین است.

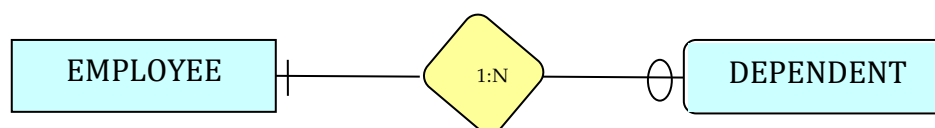
#### STUDENT-CLASS



شکل ۶-۱ مثال نمایش کاردینالی در یک ارتباط

- **وابستگی موجودیت (Existence Dependency):** وابستگی موجودیت یا Existence-Dependency در حالتی مهم است که برای ایجاد یک Entity-Instance ضرورت باشد تا یک یا اضافه از یک Entity-Instance در جدول یا جدول‌های دیگر از قبل موجود باشد. در کدام زمان چنین اتفاقی روی می‌دهد؟ وقتی ارتباط یک به چندین (1:N) بین دو Entity موجود باشد، به صورت قانونی باید کاپی شناسه یکی از Entity ها در دیگر آن اضافه شود، در این صورت Entity-Instance اولی باید موجود باشد تا Entity-Instance دومی اضافه گردیده بتواند. بناءً ترتیب داخل کردن Entity-Instance ها در ایجاد دیتابیس بسیار مهم است.
- **Entity های ضعیف (Weak Entities):** مدل E-R دارای یک نوع مخصوص Entity به نام Weak-Entity یا Entity ضعیف است. چنین Entity ها وابسته Entity های دیگر در دیتابیس است. بدین مفهوم که وجود Entity دیگر ضروری بوده، اگر آن Entity (که Strong Entity به آن گفته می‌شود) وجود نداشته باشد، این نوع Entity نمی‌تواند وجود داشته باشد. در شکل پایین دیتای مربوط

DEPENDENT، متکی به دیتای EMPLOYEE است. اگر EMPLOYEE وجود نداشته باشد، دیتای مربوطه، یعنی دیتای DEPENDENT، نمی‌تواند وجود داشته باشد.



### Weak Entity Example

شکل ۷-۱ مثالی از Entity ضعیف

طوری که در شکل بالا دیده می‌شود، Weak-Entity ها در چوکات با کناره‌های گول نشان داده می‌شود، هم‌چنان چوکات مربوط به چنین ارتباط نیز به عین شکل (با کناره‌های گول) نشان داده می‌شود. باید یادآوری گردد این که در بعضی کتاب‌ها، برای نشان دادن Weak-Entity ها در مدل E-R چوکات Entity و هم چوکات Relationship مربوطه به صورت دو خطه (Double-Line) نشان داده می‌شوند.

- **Entity های ترکیبی (Composite Entities):** برای اجراء گذاشتن ارتباط چندین به چندین (N:M) بین دو Entity، باید کاپی شناسه‌های (Identifiers) هر دو Entity گرفته شده و در یک Entity جدید، جایگزین شود. برای به اجراء گذاشتن این عمل، دو ارتباط جدید از نوع یک به چندین (1:N) یا یک به یک (1:1) ایجاد می‌شوند. به این مفهوم که یک ارتباط چندین به چندین باید به دو ارتباط یک به چندین و یا یک به یک تقسیم شود. چون در ارتباط نوع 1:N، کاپی شناسه جناح یک Entity یا یک در جناح چندین یا Entity دو جایگزین می‌شود و همین کار به ترتیب بین Entity سه و دو نیز باید اجراء گردد. همین Entity جدیداً تشکیل شده به نام Entity ترکیبی یا Composite-Entity یاد می‌شود.

### سیستم مدیریت دیتابیس ارتباط (Relational DBMS)

تنظیم دیتا در فایل‌ها یکی از مسایل عمده برای تغییر به سمت دیجیتلی شدن دنیا بود. ذخیره و استفاده دیتا و نیز تنظیم دیتا به شکلی که دوباره به آسانی مورد دسترسی قرار گیرد یکی از ضرورت‌های اصلی در دنیای استفاده از کمپیوترها به شمار می‌رود. برنامه‌های کمپیوتر که به این منظور انکشاف داده شده‌اند، یکی از مفاهیم جدید را در ساحة دیتابیس در بر می‌گیرد.

این برنامه‌ها دسته‌هایی را شامل می‌شوند که به نام سیستم‌های مدیریت دیتابیس‌های ارتباطی (Relational Database Management Systems – RDBMS) یاد می‌شوند. تحقیقات و کارهای عملی جهت انکشاف و شکل‌دهی هم‌چون سیستم‌ها در جریان سال‌ها متمادی صورت گرفته است. هدف از این تحقیقات نیز

ساده‌سازی و عمومی‌سازی موارد تنظیم دیتا به شکل کمپیوتری نسبت به اشکال سنتی تنظیم دیتا و فایل‌ها گفته شده است. یک سیستم مدیریت دیتابیس مدرن (DBMS) بخش‌های ذیل را دارا است:

- Entity های جهان واقعی (Real-World Entity): یک DBMS مدرن بیش‌تر حقیقی جلوه نموده و دیتاهای جهان به شکل حقیقی به خاطر دیزاین ساختمان‌اش استفاده می‌شود. در این نوع سیستم‌ها مشخصه‌ها و رُل آن‌ها در نظر گرفته می‌شود. به طور مثال در یک دیتابیس مکتب، شاگردان به حیث یک Entity استفاده شده و سن شاگردان به حیث یک مشخصه همین Entity تنظیم می‌شود.
- جدول‌ها بر اساس رابطه‌ها (Relation-based Tables): در این سیستم Entity ها و روابط بین آن‌ها باعث تشکیل جدول‌ها در دیتابیس می‌شود. یک استفاده‌کننده در این نوع سیستم‌های مدیریت دیتا می‌تواند با دیدن نام جدول‌های یک دیتابیس، ساختمان آن را بفهمد.
- جدا بودن دیتا و اپلیکیشن (Isolation of Data and Application): یک سیستم دیتابیس به صورت کل از دیتای آن مستقل و جدا است؛ یعنی بخش‌های دیتای سیستم و دیتای استفاده‌کننده از هم جدا و متمایز است. دیتابیس یک Entity فعال محسوب می‌شود، در حالی که دیتا از نظر نحوه فعالیت دیتابیس یک Entity غیرفعال به حساب می‌آید. به عین ترتیب DBMS میتادیتا (Metadata) را نیز ذخیره می‌نماید. میتادیتا عبارت از معلومات بخش سیستم در مورد ذخیره، پروسس و استفاده از دیتای استفاده‌کنندگان است. میتادیتا به خاطر سهولت پروسس دیتا در DBMS استفاده می‌شود.
- کمتر شدن دیتای زائد (Less Redundancy): سیستم‌های مدرن مدیریت دیتا قواعد نورمال‌سازی (Normalization) را دنبال می‌نماید. بر اساس شکل‌های نورمال یک جدول به جدول‌های دومی تجزیه شده و این تجزیه در صورت وجود دیتای تکراری به میان می‌آید. به طور مثال در یک جدول، معلومات کارمندان شرکتی با جزئیات وظیفه آن‌ها ذخیره می‌شود. در صورتی که چندین کارمند (مثلاً 100 نفر) عین وظیفه را داشته باشند، جزئیات وظیفه‌شان به صورت تکراری خواهد بود. بر اساس نورمال‌سازی اگر جزئیات وظیفه کارمندان به جدول دومی انتقال داده شود، از تکرار عین معلومات جلوگیری شده و اصل دیتای زائد مراعات خواهد شد.
- سازگاری دیتا (Consistency): بر اساس این اصل، هر رابطه یا جدولی در یک دیتابیس به شکل سازگار با دیگر بخش‌ها تنظیم می‌شود. در DBMS ها میتود و تکنیک‌هایی موجود است که توسط آن‌ها عدم سازگاری جدول‌ها شناسایی و اصلاح می‌شود. یک DBMS در مقایسه با سیستم‌های سنتی در ذخیره فایل‌ها و دیتاها، به مراتب دارای بخش‌های کامل‌تر و بهتر در تنظیم دیتا است.
- لسان کیوری (Query Language): سیستم مدیریت دیتابیس به لسان کیوری مجهز است. لسان کیوری این نوع سیستم‌ها را در زمینه استفاده و ترکیب دیتا بیش‌تر کارآمد می‌سازد. استفاده‌کنندگان هر تعداد فیلتر را که ضرورت داشته باشند، می‌توانند بالای دیتا وضع نموده تا ست دلخواه معلومات‌شان را از دیتابیس به دست بیاورند. در سیستم‌های سنتی ذخیره و استفاده از فایل‌ها قابل تطبیق نیست.

- خصوصیات ACID: سیستم‌های مدیریت دیتابیس‌ها مفاهیم تحت عنوان ACID را پیروی می‌نمایند. اصطلاح ACID از چهار کلمه (Atomicity, Consistency, Isolation و Durability) گرفته شده است. این چهار کلمه صفات DBMS ها را نشان می‌دهد. توضیح مختصر کلمات یاد شده طور ذیل است:

- Atomicity: تمام کارهای پلان شده در یک فعالیت باید تکمیل شوند، در غیر این صورت تمام فعالیت (transaction) حذف شود.
- Consistency: یک فعالیت دیتابیس را در حالت غیر سازگار (inconsistent) رها نکند.
- Isolation: هر فعالیت دیتابیس به صورت مستقل صورت گرفته و یکی به دیگری تداخل ننماید.
- Durability: فعالیت‌های تکمیل‌شده به حال خود باقی بماند، حتی اگر کمپیوترهای سرور خاموش و دوباره روشن شوند.

طوری که در بالا توضیح داده شد، مفاهیم فوق بر ترانزکشن‌ها قابل تطبیق است. ترانزکشن‌ها در دیتابیس به خاطر کار بر دیتا استفاده می‌شوند. فعالیت‌های ACID در یک دیتابیس که چندین استفاده‌کننده در آن کار می‌نماید به خاطر حفظ سلامت دیتا و اطمینان از دقت کار، یکی از صفات اساسی سیستم مدیریت دیتابیس به شمار می‌رود.

- چندین استفاده‌کننده و دسترسی هم‌زمان (Multiuser and Concurrent Access): سیستم مدیریت دیتابیس، محیط چندین استفاده‌کننده را حمایت کرده و چندین استفاده‌کننده را اجازه می‌دهد تا دسترسی به دیتا و نیز کارکردن با دیتا را به شکل موازی و در یک‌زمان داشته باشند. اگرچه در قسمت داشتن اجازه کار روی دیتاهای مختلف محدودیت‌هایی برنامه‌ریزی شده، ولی این‌گونه محدودیت‌ها معمولاً از نظر استفاده‌کنندگان پنهان است.

- چندین نماها (Multiple Views): سیستم‌های مدیریت دیتابیس‌ها، نماهای مختلفی را برای استفاده‌کنندگان مهیا می‌سازد. یک استفاده‌کننده که در بخش فروشات یک شرکت است، نمای متفاوتی از دیتابیس شرکت را نظر به استفاده‌کننده بخش تولید می‌بیند؛ یعنی کنترل‌هایی که برای فروشات است در بخش تولید ضرورت نبوده و به عین شکل کنترل‌های روی سیستم که در بخش تولید ضروری است برای بخش فروشات شاید زائد باشد. طرح و دیزاین بخش‌های مختلف برای استفاده‌کنندگان مختلف در DBMS ها کنترل و استفاده شده می‌تواند. توسط این کار نماهایی که به استفاده‌کنندگان مشخص دیزاین می‌شوند، باعث تمرکز بیش‌تر استفاده‌کنندگان خواهد بود که این امر مستقیماً به سیستم مدیریت دیتابیس و امکانات این سیستم‌ها وابسته است.

- امنیت (Security): مسئله استفاده از نماهای مشخص برای استفاده‌کنندگان مشخص، یکی از نکات مهم در قسمت امنیت دیتا به شمار می‌رود. در این‌گونه موارد، استفاده‌کنندگان یک بخش نمی‌توانند به دیتا و کنترل‌های بخش دیگر دسترسی داشته باشند. سیستم‌های مدیریت دیتابیس‌ها میتودهایی را پیشنهاد می‌نماید که از طریق آن‌ها محدودیت‌هایی بر دیتا، در زمان داخل کردن به دیتابیس و

گرفتن همان دیتا از دیتابیس بعد از ثبت شدن را وضع نماید. DBMS ها وضع درجه‌های مختلف امنیت بالای دیتا را اجازه می‌دهد. این درجه‌های امنیتی صلاحیت دسترسی در دیتا را به استفاده‌کنندگان می‌دهد. به طور مثال یک استفاده‌کننده از بخش فروشات یک شرکت، اجازه دیدن دیتای بخش خریداری را ندارد. در حالی که هر دو بخش دیتا، در یک دیتابیس ذخیره می‌شود.

در نهایت گفته می‌توانیم که سیستم‌های مدرنِ تنظیم و مدیریت دیتابیس‌های ارتباطی که به اختصار RDBMS نامیده می‌شوند، دارای امکانات وسیع در قسمتِ داخل کردن دیتا، ذخیره و کنترل دیتا و نیز امکان استفاده دوباره از دیتا را دارند. به کمک این سیستم‌ها دیتای استفاده‌کنندگان، ذخیره دیتا، کنترل دسترسی استفاده‌کنندگان به دیتا و غیره موارد مهم، به کمک مدیریت دیتا و دیتابیس به صورت دقیق و مناسب در مقایسه با سیستم‌های سنتی در تنظیم دیتا و معلومات انجام می‌گیرد.





در فصل اول اساسات مدیریت دیتابیس‌های ارتباطی، موضوعات تخیکی به ترتیب توضیح و تشریح شده است. در مقدمه این فصل دیتای معلومات و موارد استفاده از دیتا توضیح داده شد. معلومات مختصر در مورد دیتابیس و اهمیت آن و نیز برنامه‌های موجود به خاطر ایجاد، انکشاف و استفاده دیتابیس‌ها ارائه شد. اهمیت دیتا در زندگی روزمره و به خاطر تصمیم‌گیری درست در موضوعات مختلف توضیح داده شد. انواع دیتا و مشخص کردن آن‌ها در ساحة مدیریت معلومات، با ذکر مثالی ارائه گردید. مدل‌های دیتا، نحوه کارشان و نیز استندهای موجود در عنوان جداگانه مورد بحث و ارزیابی قرار گرفت. در این مبحث مثال‌های روشن جهت وضاحت موضوع به کار گرفته شده است؛ و بالاخره در بخش اخیر روی سیستم مدیریت دیتابیس‌های ارتباطی توضیحات نسبتاً تفصیلی داشته‌ایم. در این توضیحات مسایل تخیکی مربوط به یک سیستم مدیریت دیتابیس ارتباط (RDBMS) به اساس استندهای موجود بین‌المللی در نظر گرفته شده و به محصلان و خوانندگان گرامی این کتاب ارائه شد.



## سوالات فصل اول

۱. فرق بین دیتا و معلومات را مختصراً بیان کنید.
۲. دیتابیس چه است؟
۳. نظریه مدل ارتباط (The Relational Model) در کدام سال ها مطرح شده است؟
۴. DBMS اختصار کدام کلمات است؟
۵. سیستم مدیریت دیتابیس ارتباطی را توضیح دهید.
۶. میتادیتا (Metadata) چه است؟
۷. قبل از به میان آمدن کمپیوترهای برقی یا الکترونیکی، دیتا به کدام شکل تهیه و ذخیره می شد؟
۸. سن شاگردان در مکتب الف در کدام کتگوری باید باشد؟
۹. انواع دیتای قابل استفاده در دیتابیس ها را صرف نام بگیرید.
۱۰. مدل های دیتا به خاطر چه و به کدام هدف در دیتابیس ها استفاده می شوند؟
۱۱. دیتامدل ER بیش تر در کدام بخش ها استفاده می شود؟
۱۲. کلمه Null در دیتابیس به چه مفهوم است.
۱۳. مشخصه های (Attributes) نظر به ساختمان شان و نیز نظر به داشتن محتوا به چند نوع است؟ هر کدام را نام بگیرید.
۱۴. شناسه ترکیبی (Composite Identifier) را توضیح دهید.
۱۵. تفاوت بین Relationship-Class و Relationship-Instance را در یک مثال بیان کنید.

## فصل دوم

### استفاده از سیستم مدیریت دیتابیس ارتباط



**هدف کلی:** محصلان با شیوه استفاده از سیستم مدیریت دیتابیس ارتباط (Relational Database Management System) به اختصار RDBMS آشنا شوند.

**اهداف آموزشی:** در پایان این فصل محصلان قادر خواهند شد تا:

۱. سیستم مدیریت دیتابیس ارتباط شیوه ایجاد دیتابیس‌ها، جدول‌ها و ساختمان‌های دیگر در سیکویل را شرح نمایند.
۲. طُرُق مختلف داخل کردن دیتا به جدول‌ها، تغییر آوردن و پاک کردن دیتا را توضیح نمایند.
۳. شیوه‌های پاک کردن دیتا از جدول‌ها، حذف جدول‌ها و دیتابیس‌ها را بیان نمایند.

عبارت از مجموعه برنامه‌هایی است که جهت ذخیره، تنظیم و استفاده دیتا به اساس قواعد مدل ارتباط (The Relational Model) کار می‌نماید. استفاده‌کنندگان و انکشاف‌دهندگان دیتابیس با آگاهی از قواعد مدل ارتباط، دیتا و معلومات تحلیل‌شده را به کمک سیستم‌های مدیریت دیتابیس آماده ساخته و در اختیار استفاده‌کنندگان نهایی (End Users) قرار می‌دهد. استفاده و به کارگیری سیستم‌های مدیریت دیتابیس ضرورت به آموزش و تجربه کافی در این ساحه را ایجاب می‌نماید. درمجموع، سیستم‌ها در مدیریت دیتابیس مجهز به یکی از لسان‌های کیوری دیتا است. لسان‌های کیوری دیتا نه تنها در قسمت ذخیره دیتا به کار گرفته می‌شود بلکه در قسمت ایجاد دیتابیس و ساختمان‌های داخلی آن‌ها و نیز در قسمت استخراج معلومات از دیتابیس به گونه وسیع استفاده می‌شود. سیکویل SQL اختصار کلمات (Structured Query Language) بوده و به حیث اساسی‌ترین لسان کیوری در سیستم‌های مدیریت دیتابیس استفاده می‌شود.

سیکویل یک لسان متنی بوده و درمجموع طرز‌العمل‌های آن به سه شکل نوشته می‌شوند:

- طرز‌العمل‌های Data Manipulation Language (DML): از این نوع طرز‌العمل‌ها به خاطر پاک کردن، وارد کردن و تغییر دیتا استفاده می‌شوند.
- طرز‌العمل‌های Data Definition Language (DDL): از این نوع طرز‌العمل‌ها جهت ایجاد دیتابیس‌ها، جدول‌ها، نماها، ارتباط‌ها و ساختمان‌های دیگر داخل دیتابیس یا میتادیتا استفاده می‌شوند.
- طرز‌العمل‌های Data Control Language (DCL): از این نوع طرز‌العمل‌ها، جهت کنترل دیتا در قسمت مدیریت دیتابیس استفاده می‌شوند.

طوری که گفته شد سیکویل یک لسان متنی یا 'Text-Oriented Language' است و مدت‌ها قبل‌تر از سیستم‌های عامل با گرافیک بالا دیزاین شده است. پس به خاطر استفاده از لسان سیکویل یک صفحه ساده متنی کفایت می‌کند.

اکثر سیستم‌های مدیریت دیتا مانند MS Access، MS SQL Server، Oracle و... وسایل گرافیکی را در دسترس استفاده‌کنندگان قرار می‌دهند، در حالی که تمام این‌ها کود سیکویل را استفاده نموده و در Backend طرز‌العمل‌ها، توسط سیکویل راه‌اندازی می‌شوند. به خاطر اهمیت موضوع، بحث بیش‌تر بالای استفاده از طرز‌العمل‌های لسان سیکویل جهت ایجاد، انکشاف و استفاده دیتابیس و ساختمان‌های داخلی آن صورت گرفته و موضوع با مثالی از دیتابیس بیش‌تر توضیح می‌گردد. در ادامه بعد از ارائه مثال، طریقه‌های ایجاد دیتابیس و جدول‌ها، ایجاد ارتباط‌ها، داخل کردن دیتا، پاک کردن دیتا و حذف جدول‌ها و دیتابیس‌ها توضیح داده خواهد شد.

## مثالی از دیتابیس:

یک دیتابیس مکتب در نظر گرفته شود که دارای جدول‌های ذیل باشد:

PROJECT (ProjectID, Name, Department, MaxHours)

EMPLOYEE (EmployeeNumber, Name, Phone, Department)

ASSIGNMENT (ProjectID, EmployeeNum, HoursWorked)

دیتای نمونه برای سه جدول یاد شده در شکل پایین نشان داده شده است:

PROJECT			
ProjectID	Name	Department	MaxHours
1000	03 Portfolio Analysis	Finance	75
1200	03 Tax Prep	Accounting	145
1400	04 Product Plan	Marketing	138
1500	04 Portfolio Analysis	Finance	110
EMPLOYEE			
Employee Number	Name	Phone	Department
100	Mary Jacobs	285-8879	Accounting
200	Keni Numoto	287-0098	Marketing
300	Heather Jones	287-9981	Finance
400	Rosalie Jackson	285-1273	Accounting
500	James Nestor		Info Systems
600	Richard Wu	287-0123	Info Systems
700	Kim Sung	287-3222	Marketing
ASSIGNMENT			
ProjectID	EmployeeNum	HoursWorked	
1000	100	17.5	
1000	300	12.5	
1000	400	8	
1000	500	20.25	
1200	100	45.75	
1200	400	70.5	
1200	600	40.5	
1400	200	75	
1400	700	20.25	
1400	500	25.25	

شکل ۱-۲ نمونه دیتا برای جدول‌های پروژه (Project)، کارمند (Employee) و مأموریت (Assignment)

کلید اولیه (Primary Key) جدول پروژه، عبارت از فیلد ProjectID است و کلید اولیه جدول کارمند، ستون EmployeeNumber است. به عین ترتیب کلید اولیه جدول مأموریت به شکل ترکیبی عبارت از فیلدهای (ProjectID, EmployeeNum) است. این دو فیلد عبارت از کلیدهای خارجی (Foreign Keys) جدول‌های پروژه و کارمند نیز می‌باشند.

در این دیتابیس هر سطر جدول پروژه می‌تواند به چندین سطر جدول مأموریت ارتباط داشته باشد و نیز هر سطر جدول کارمند نیز با چندین سطر جدول مأموریت می‌تواند ارتباط داشته باشد.

قواعد RIC<sup>1</sup> (Referential Integrity Constraint) و قواعد تجاری (Business Rules)<sup>2</sup> برای این جدول‌ها و روابط بین‌شان طور ذیل است:

- عناصر ستون ProjectID در جدول مأموریت باید شامل ستون ProjectID در جدول پروژه باشند.
- عناصر ستون EmployeeNum در جدول مأموریت باید شامل ستون EmployeeNumber در جدول کارمند باشند.
- اگر سطری از جدول پروژه پاک شود، تمام سطرهای جدول مأموریت که با سطر متذکره ارتباط داشته باشد، پاک می‌شوند.
- اگر سطری از جدول کارمند پاک شود و سطر متذکره با سطر و یا سطرهایی در جدول مأموریت ارتباط داشته باشد، این چنین عملیه اجازه داده نمی‌شود.

## ۲.۱ ایجاد دیتابیس در سیکویل

دیتابیس‌ها در سیکویل به کمک طرزالعمل‌های مشخص ایجاد می‌شوند. اساسی‌ترین واحدی که در آن دیتا ذخیره می‌شود به نام رابطه یاد شده و در اصطلاح جدول (table) نامیده می‌شود. جدول‌های دیتابیس در سیکویل تا زمانی که دیتابیزی موجود نباشد، ایجاد شده نمی‌تواند؛ یعنی در سیستم‌های مدیریت دیتابیس نخستین قدم ایجاد دیتابیس است. یک دیتابیس زمانی که ایجاد می‌شود، باید دارای نام و تعریف مشخص باشد. در سیکویل طرزالعملی که توسط آن یک دیتابیس ایجاد می‌شود، مشخص کردن نام فایل (دیتابیس) شرط حتمی است. بعد از ایجاد دیتابیس با استفاده از همان نام، دیتابیس مشخص شده قابلیت استفاده را پیدا می‌کند؛ یعنی بعد از ایجاد، یک دیتابیس باید فعال (قابل استفاده) شود. اضافه کردن ساختمان‌های دیگر به دیتابیس صرف بعد از انجام این دو عملیه صورت می‌گیرد.

---

<sup>1</sup> قاعده RIC یا Referential Integrity Constraint طریقه‌یی است که به اساس آن محتوای کلید خارجی، باید ست فرعی محتوای کلید اولیه در جدول بالایی باشد؛ یعنی عناصر دیتای جدول اول باید در ستون کلید اولیه موجود باشد تا در ستون کلید خارجی در جدول پایین اضافه شده بتواند.

<sup>2</sup> قواعد تجاری (Business Rules) مسایل و محدودیت‌های مربوط یک واحد را در بر می‌گیرد تا به اساس همان محدودیت‌ها دیتابیس مربوطه دیزاین و قابل استفاده ساخته شود.



جهت فعال ساختن کوماندلاین (Command Line) یا یک سیستم مدیریت دیتابیس و استفاده عملی از سیکویل باید مراحل ذیل طی شوند:

- به خاطر استفاده از سیستم مدیریت دیتابیس (مثلاً MySQL) اولاً باید در دایرکتوری C:\...\MySQL\BIN رفته و بعد طرزالعمل ذیل تایپ شود:

```
>>mysql -u root
```

- در صورت موجود بودن رمز عبور (پاسورد) در سیستم مدیریت دیتابیس طرزالعمل ذیل استفاده شود:

```
>>mysql -p -u root
```

- جهت ایجاد یک فایل متنی (Logfile) با اکستنشن.txt و ثبت نمودن تمام طرزالعمل‌ها و خروجی‌ها در آن فایل، طرزالعمل‌های ذیل استفاده شده می‌تواند:

```
>>\T path_file_name
```

و یا

```
>>tee path_file_name
```

- طرزالعمل‌های ذیل، فایل متنی از قبل ایجاد شده (لوگ فایل) را غیر فعال می‌سازد؛ یعنی دیگر دیتا به فایل متذکره ثبت نمی‌شود.

```
>>\t path_file_name
```

و یا

```
>>notee path_file_name
```

نوت: path\_file\_name عبارت از نام و موقعیت فایل متنی با اکستنشن.txt است. به طور مثال چنین نوشته‌شده می‌تواند:

```
>>\T c:\kabul\today.txt
```

در طرزالعمل بالا '\T' نام طرزالعمل، 'c' نام درایو، 'kabul' نام دایرکتوری در درایو c و 'today.txt' عبارت از نام لوگ فایل است که سیستم مدیریت دیتابیس آن را ایجاد می‌نماید. اگر فایل از قبل موجود باشد، سیستم مدیریت دیتابیس آن را ویرایش نموده دیتا به آن علاوه یا (Append) می‌شود.

- طرزالعمل ذیل دیتابیسی به نام مکتب را ایجاد می‌کند:

```
>>CREATE DATABASE SCHOOL;
```

- به خاطر دیدن دیتابیس‌ها طرزالعمل ذیل استفاده می‌شود:

>>SHOW DATABASES;

- جهت استفاده و فعال نمودن دیتابیس موجود به نام مکتب، به طور ذیل تایپ می‌شود:

>>USE SCHOOL;

- جدول‌های موجود در دیتابیس فعال (تعیین شده) توسط طرزالعمل ذیل مشاهده می‌شود:

>>SHOW TABLES;

نتیجه استفاده طرزالعمل‌های ابتدایی از یک سیستم مدیریت دیتابیس در شکل پایین نشان داده شده است.

```

C:\Program Files (x86)\MySQL\MySQL Server 5.5\bin\mysql.exe
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.5.16 MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol      |
| db         |
| mysql      |
| performance_schema |
| phpmyadmin |
| test       |
| webauth    |
+-----+
8 rows in set (0.08 sec)

mysql> create database school;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol      |
| db         |
| mysql      |
| performance_schema |
| phpmyadmin |
| school     |
| test       |
| webauth    |
+-----+
9 rows in set (0.00 sec)

mysql> use school
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql> _

```

شکل ۲-۲ صفحه سیستم مدیریت دیتابیس با طرزالعمل‌های اولیه استفاده از آن

طوری که در شکل بالا دیده می‌شود، در اولین قسمت صفحه، دیتابیس از آدرس مشخص آن در کمپیوتر باز شده است. طرزالعمل مشخص در دیتابیس‌ها به راه انداخته شده و در فهرست تمام دیتابیس‌های مربوط به

سیستم که به شکل خودکار موجود می‌باشند، نشان داده شده است. طرزالعمل بعدی، دیتابیس را به نام مکتب ایجاد نموده است. زمانی که طرزالعمل قبلی نشان‌دهنده دیتابیس‌ها برای بار دوم در این قسمت به راه انداخته شده و دیتابیس تازه ایجاد شده مکتب نیز شامل فهرست گردیده است. طرزالعمل استفاده از دیتابیس مکتب، این دیتابیس را فعال ساخته و طرزالعمل نشان‌دهنده جدول‌ها (Show Tables) با نشان‌دهنده ست خالی در نتیجه، همانا جدول‌های دیتابیس فعال (دیتابیس مکتب) را نشان می‌دهد؛ یعنی دیتابیس به نام مکتب تا این مرحله صرف ایجاد شده است، ولی ساختمان‌های دیگر مانند جدول‌ها در آن موجود نیستند. بعد از ایجاد دیتابیس، استفاده‌کننده می‌تواند جدول‌ها را در آن علاوه نماید.

## ۲.۲ ایجاد جدول‌ها و ساختمان‌های دیگر در سیکویل

به خاطر ایجاد جدول‌ها در دیتابیس بازهم از طرزالعمل ایجاد به نام (create table) استفاده می‌شود. از طرزالعمل ایجاد جدول جدید در سیکویل به طور ذیل استفاده می‌شود:

```
CREATE TABLE newtablename (  
    three-part column description,  
    three-part column description,  
    three-part column description,  
    optional table constraints  
    ...  
);
```

در پایین دستورهای سیکویل برای ایجاد جدول‌های پروژه، کارمند و ماموریت نشان داده شده است.

```
CREATE TABLE PROJECT (
```

```
    ProjectID Integer Primary Key,  
    Name Char(25) Not Null,  
    Department VarChar(100) Null,  
    MaxHours Numeric(5,2) Default 100);
```

```
CREATE TABLE EMPLOYEE (
```

```
    EmployeeNumber Integer Primary Key,  
    Name Char(25) Not Null);
```

```

Phone Char(8),
Department VarChar(100));

CREATE TABLE ASSIGNMENT (
ProjectID Integer Not Null,
EmployeeNum Integer Not Null,
HoursWorkedNumeric(5,2) Default 10);

```

قسمت‌هایی که توسط آن‌ها یک ستون (Column) تعریف می‌گردد، دارای بخش‌های ذیل است:

- نام ستون؛
- نوع دیتای ستون؛
- یک قید (Constraint) اختیاری برای ستون.

قیدهای ستون شامل کلید اولیه ناشناخته (Null)، غیر ناشناخته (Not Null) و خودکار (Default) بوده می‌توانند.

در مثال بالا ستون ProjectID از نوع عدد صحیح (Integer) بوده و در قسمت قید آن کلید اولیه تعیین شده است، ستون بعدی (Name) از نوع کرکتر است که به حروف کلیدی Char نشان داده شده است و دارای 25 کرکتر طول است. قید Not Null نشان دهنده آن است که یک ریکورد باید در این فیلد دیتا داشته باشد تا این که کرسر به ستون بعدی برود.

ستون سه (Department) از نوع VarChar(100) بوده و قسمت قید آن عبارت از Null است. کلمه VarChar به مفهوم دیتا با طول متغیر (Variable-Length Character) است. پس ستون دیپارتمنت می‌تواند دیتا را با طول‌های متفاوت، در سطرها مختلف داشته باشد و طول اعظمی در این ستون عبارت از ۱۰۰ کرکتر است. اگر یکی از سطرها دارای ۴ کرکتر دیتا باشد، در حافظه تنها همان قسمت ۴ بیت ریزرف و استفاده می‌گردد؛ و قید Null نشان دهنده آن است که ستون متذکره می‌تواند بدون دیتا نیز باشد.

طوری که ذکر شد، متحول Char همیشه با طول ثابت است. Char(25) برای ستون نام، طول این فیلد را تعیین می‌نماید و در حافظه، همان ۲۵ کرکتر برای هر Cell در ستون متذکره ریزرف است؛ به عبارت دیگر، اگر فیلد نام برای یک ریکورد صرف ۸ کرکتر و یا ۲۵ کرکتر باشد، عین جای را در قسمت حافظه احتواء می‌نماید.

در این جا سؤالی خلق می‌شود که مزیت‌های متحول Varchar نظر به متحول Char در قسمت اشغال نمودن حافظه بیش‌تر است، پس چرا به عوض متحول Char از متحول Varchar استفاده نمی‌شود؟ دلیل این است

که ستون‌هایی با نوع دیتا Varchar در زمان اجراء به پروسه‌های بیش‌تری نیاز دارد. با تقلیل آوردن در قسمت حافظه، سیستم مدیریت دیتابیس وقت بیش‌تری را با اجراء پروسه‌های اضافی برای Varchar تلف می‌نماید. تولیدکنندگان سیستم‌های مدیریت دیتابیس معمولاً رهنمایی‌هایی جهت استفادهٔ متحولین برای دیتابیس‌هایی که تهیه می‌شود، ارائه می‌دارند. به این منظور اسناد یا Documentation معنای سیستم‌های مدیریت دیتابیس باید دیده شوند. نوع دیتای ستون چهارم جدول پروژه عبارت از Numeric(5,2) است. هدف از این نوع دیتا این است که عناصر ستون MaxHours متشکل از 5 رقم بوده می‌توانند، طوری که دو رقم آن به سمت راست، بعد از علامهٔ اعشاری قرار بگیرند. (برای علامهٔ اعشاری موقعیت در نظر گرفته نمی‌شود) طور مثال نمبر ۱۲۳۴۵ توسط سیستم مدیریت دیتابیس به شکل ۱۲۳.۴۵ نشان داده می‌شود. اصطلاح ۱۰۰ Default نشان می‌دهد که در ریکورد جدید، ستون MaxHours به صورت خودکار عدد ۱۰۰ را نشان بدهد طوری که اگر استفاده‌کننده بخواهد در آن تغییر آورده و یا همان عدد را حفظ کند.

نوت: به اساس قاعدهٔ استاندارد در سیکویل، در پایان هر طرزالعمل باید علامهٔ سیمی‌کولن (Semicolon) یا (;) گذاشته شود. در مثال بالا مجموعاً سه طرزالعمل نشان داده شده که به اساس هر کدام آن یک جدول خالی ایجاد می‌شود و در ختم هر کدام از طرزالعمل‌ها علامهٔ (;) گذاشته شده است. بعضی سیستم‌های مدیریت دیتابیس‌ها از این محدوده خارج بوده و موجودیت علامهٔ سیمی‌کولن در اخیر طرزالعمل ضروری پنداشته نشده و بدون آن هم طرزالعمل اجراء می‌گردد. مثال‌های ایجاد جدول‌های پروژه، کارمند و مأموریت در داخل دیتابیس مکتب در سیکویل تطبیق شده و در شکل پایین نتیجه را نشان می‌دهد.

```

C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin\mysql.exe

mysql> show tables;
Empty set (0.00 sec)

mysql> CREATE TABLE PROJECT (
  -> ProjectID Integer Primary Key,
  -> Name Char(25) Not Null,
  -> Department VarChar(100) Null,
  -> MaxHours Numeric(5,2) Default 100
  -> );
Query OK, 0 rows affected (0.15 sec)

mysql> CREATE TABLE EMPLOYEE (
  -> EmployeeNumber Integer Primary Key,
  -> Name Char(25) Not Null,
  -> Phone Char(8),
  -> Department VarChar(100)
  -> );
Query OK, 0 rows affected (0.12 sec)

mysql> CREATE TABLE ASSIGNMENT (
  -> ProjectID Integer Not Null,
  -> EmployeeNum Integer Not Null,
  -> HoursWorked Numeric(5,2) Default 10
  -> );
Query OK, 0 rows affected (0.17 sec)

mysql> show tables;
+-----+
| Tables_in_school |
+-----+
| assignment        |
| employee          |
| project           |
+-----+
3 rows in set (0.00 sec)

mysql>

```

شکل ۳-۲ اجراء سه طرزالعمل اجراء شده در سیکویل به خاطر ایجاد جدول‌ها

طوری که در شکل بالا دیده می‌شود، در مجموع دو طرز العمل سیکویل اجراء شده و نتایج آن نشان داده شده است. طرز العمل نشان‌کرده جدول در دو حالت قبل از ایجاد جدول‌ها و نیز بعد از ایجاد جدول‌های دیتابیس اجراء شده است. در حالت اول نتیجه ست خالی است، یعنی در دیتابیس جدولی اضافه نشده است. در حالت دوم جدول‌های ایجاد شده به ترتیب الفبا فهرست شده‌اند. در کنار استفاده از طرز العمل نشان‌کرده جدول، طرز العمل دیگری برای ایجاد جدول برای سه بار یعنی یک بار برای هر جدول استفاده شده است. نتیجه هر بار مثبت و در نهایت سه جدول به دیتابیس مکتب اضافه شده است.

### ۲.۳ ایجاد رابطه‌ها بین جدول‌ها در سیکویل

معلومات در مورد یک موضوع جهت تطبیق و به کارگیری مؤثر به اساس قواعد مدل ارتباط، به اجزای کوچک‌تر تحلیل و تقسیم شده که همین اجزای کوچک تقسیم‌شده به نام دیتا یاد می‌شود. به خاطر استفاده و به کارگیری دیتا، هر قطعه آن در ساختمانی به نام جدول در دیتابیس ارتباط ذخیره می‌شود. برای تفهیم بیش‌تر، مثالی در ایجاد جدول‌ها در بحث قبلی به تفصیل توضیح و به صورت عملی نشان داده شد. یک جدول به تنهایی شاید معلومات کافی در مورد موضوع مورد بحث را ارائه ننماید. طوری که در مثال مکتب دیده می‌شود یک جدول، دیتا را در مورد پروژه ذخیره می‌نماید در جدول بعدی، دیتا در مورد کارمندان را در بر می‌گیرد و جدول سومی، دیتا در مورد تعداد ساعات کار هر کارمند روی یک پروژه را مشخص می‌سازد. اگر توجه شود دیده می‌شود که هیچ‌کدام از جدول‌ها، معلومات مکمل را در بر ندارد. معلومات مکمل تنها زمانی از این دیتابیس گرفته شده می‌تواند که این سه جدول با هم دیگر ارتباط پیدا کنند. اضافه کردن ارتباط (Relationship) بین جدول‌ها، قدرت دیتابیس‌ها را به نمایش گذاشته و دیتابیس‌ها ارتباط مؤثریت خود را نمایان می‌سازد.

جهت دستیابی به موضوع بالا، دو ارتباط بین جدول‌های دیتابیس مکتب باید ایجاد گردد. به خاطر ایجاد ارتباط بین جدول‌ها، برای هر ارتباط حد اقل دو فیلد لازم است: فیلد کلید اولیه در جدول اول یا بالایی و فیلد کلید خارجی در جدول دوم یا پایینی. نوع دیتای فیلدهای کلید اولیه و کلید خارجی که در یک ارتباط استفاده می‌شوند باید یک نوع باشد؛ به این معنی که اگر نوع دیتای کلید اولیه نمبر باشد، نوع دیتای کلید خارجی نیز باید نمبر باشد تا ارتباط برقرار شده بتواند. در صورت اختلاف نوع دیتا، اشتباه در کود به میان آمده و ارتباط برقرار نمی‌شود. مسئله دیگر در مورد ارتباط دیتا عبارت از ست دیتای قابل ذخیره در هر دو فیلد است. ست عناصر کلید خارجی در جدول پایین همیشه یک ست فرعی عناصر دیتای کلید اولیه در جدول بالا است.

کودی که در پایین ذکر شده باید در جدول مأموریت علاوه گردد تا ارتباط اول بین جدول مأموریت و جدول پروژه و نیز ارتباط دوم بین جدول مأموریت و جدول کارمند ایجاد شود. در هر دو ارتباط، جدول مأموریت به حیث جدول دوم کلیدهای خارجی (کاپی کلیدهای اولیه) و جدول‌های اول هر کدام جدول پروژه و جدول کارمند را در بر دارد.

CREATE TABLE ASSIGNMENT (

...

HoursWorked Numeric(5,2) Default 10,

CONSTRAINT ProjectFK FOREIGN KEY (ProjectID) REFERENCES PROJECT  
(ProjectID) ON DELETE CASCADE,

CONSTRAINT EmployeeFK FOREIGN KEY (EmployeeNum) REFERENCES  
EMPLOYEE (EmployeeNumber) ON DELETE NO ACTION);

اجراء كود بالا در برنامه صورت گرفته و نتیجه آن در شكل پايين نشان داده شده است.

```

mysql> DROP TABLE ASSIGNMENT;
Query OK, 0 rows affected (0.09 sec)

mysql> CREATE TABLE ASSIGNMENT (
  -> ProjectID Integer Not Null,
  -> EmployeeNum Integer Not Null,
  -> HoursWorked Numeric(5,2) Default 10,
  -> CONSTRAINT ProjectFK FOREIGN KEY (ProjectID)
  -> REFERENCES PROJECT (ProjectID) ON DELETE CASCADE,
  -> CONSTRAINT EmployeeFK FOREIGN KEY (EmployeeNum)
  -> REFERENCES EMPLOYEE (EmployeeNumber) ON DELETE NO ACTION
  -> );
Query OK, 0 rows affected (0.34 sec)

mysql> DESCRIBE ASSIGNMENT;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ProjectID | int(11) | NO | MUL | NULL | |
| EmployeeNum | int(11) | NO | MUL | NULL | |
| HoursWorked | decimal(5,2) | YES | | 10.00 | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)

mysql> DESCRIBE PROJECT;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ProjectID | int(11) | NO | PRI | NULL | |
| Name | char(25) | NO | | NULL | |
| Department | varchar(100) | YES | | NULL | |
| MaxHours | decimal(5,2) | YES | | 100.00 | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)

mysql> DESCRIBE EMPLOYEE;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| EmployeeNumber | int(11) | NO | PRI | NULL | |
| Name | char(25) | NO | | NULL | |
| Phone | char(8) | YES | | NULL | |
| Department | varchar(100) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)

mysql>

```

شكل ۴-۲ اجراء طرز العمل هاي از بين بردن جدول مأموريت و ايجاد دوباره آن با اضافه شدن ارتباطها به دو جدول ديگر در همان ديتابيس



به خاطر ایجاد ارتباطها بین جدولهای دیتابیس مکتب، در قدم نخست جدول مأموریت که جدول پایین در ارتباطها محسوب می‌شود، با استفاده از طرزالعمل Drop Table از سیستم پاک شده و همین جدول با اضافه شدن قیدهای کلید خارجی با استفاده از طرزالعمل Create Table دوباره ایجاد شد. بعد از ایجاد دوباره جدول مأموریت، کلیدهای خارجی به دو جدول دیگر مشخص شد. با اجراء این حالت دیتایی که به ستونهای کلید خارجی در جدول مأموریت اضافه می‌شود، باید از قبل در ستونهای کلید اولیه در جدولهای بالایی (جدول پروژه و جدول کارمند) موجود باشد.

هر کدام از شرطهای ذکر شده در کد، مفهوم خود را دارد. با استفاده از شرط "ON DELETE CASCADE" پاک کردن ریکورد دیتا در حال ارتباط مجاز بوده در صورت پاک کردن دیتا در جدول بالایی (پروژه)، ریکوردهای مربوطه در جدول پایینی (مأموریت) نیز از بین می‌روند. به عین ترتیب با استفاده از شرط "ON DELETE NO ACTION" برای پاک کردن دیتا اجازه داده نمی‌شود؛ یعنی ریکوردهای جدول بالایی در صورت داشتن دیتا در جدول پایینی به اساس ارتباط تا زمانی پاک شده نمی‌تواند تا در جدول پایینی دیتای مربوطه پاک نشود.

## ۲.۴ داخل کردن، تغییر آوردن و پاک کردن دیتا در سیکویل

سه عملیه داخل کردن دیتا به جدولها (Insert)، تغییر آوردن در دیتای جدولها (Update) و پاک کردن دیتا جدولها (Delete) در سیکویل ممکن بوده و توسط استفاده کنندگان انجام داده می‌شود. عملیه‌های یاد شده با مثالهای مناسب در ذیل توضیح می‌گردند.

### ۲.۴.۱ داخل کردن دیتا به دیتابیس

یکی از راههای استندرد داخل کردن دیتا به جدولهای دیتابیس در سیکویل، استفاده از طرزالعمل INSERT است. تولیدکنندگان برنامه‌های سیستم مدیریت دیتابیس یا DBMS ها معمولاً برنامه‌های فرعی را جهت داخل نمودن دیتا به دیتابیس دیزاین می‌نمایند. در دیزاین چنین برنامه‌ها طرزالعمل INSERT استفاده می‌گردد. برنامه‌های یاد شده از طریق صفحه طرزالعمل (Command Prompt) و یا هم مستقیم با استفاده از فایل‌های کود به راه انداخته می‌شوند.

استفاده نمودن از طرزالعمل INSERT سخت نبوده، یعنی باعث ایجاد مشکلات در دیتابیس نمی‌گردد. طوری که این طرزالعمل یک ریکورد را به جدول داخل می‌نماید، در صورت درست نبودن آن با استفاده از طرزالعمل دیگر سیکویل یعنی طرزالعمل DELETE معلومات داخل شده دوباره پاک شده می‌توانند. طرزالعمل INSERT در تفاوت با طرزالعمل CREATE، با یوزر دیتا سروکار داشته در قسمت ساختمان خود دیتابیس و یا ساختمان جدولها و دیگر اجزای دیتابیس یعنی میتادیتا ارتباط ندارد.



بعضی اوقات طرزالعمل INSERT مشکلاتی در قسمت داخل نمودن ریکوردهای تکراری به یک جدول به بار می‌آورد. در حقیقت این نوع مشکلات برمی‌گردد به ساختمان اولیه دیتابیس و Business-Rule هایی که در وقت ایجاد دیتابیس یاد شده، استفاده شده است.

طوری که معلوم است در سیکویل، یک جدول می‌تواند بدون داشتن کلید اولیه وجود داشته باشد یعنی یگانه (Unique) بودن ریکوردها در سیکویل مطرح نیست. بناءً در هر دیتابیس سیکویل باید در وقت دیزاین آن مشکلات یاد شده در نظر گرفته شده و پروسیجرهایی استفاده شوند که در جدول‌های دیتابیس ریکوردهای تکراری دیتا را قبول نمایند. در صورت تطبیق این مسئله، طرزالعمل INSERT با اطمینان تطبیق شده و مشکلی ایجاد نمی‌نماید.

### شکل عمومی طرزالعمل INSERT

طرزالعمل INSERT به صورت عمومی به دو شکل استفاده می‌شود:

1. INSERT INTO MYTABLE (Columns) VALUES (Values);
2. INSERT INTO MYTABLE (Columns)

SELECT Columns FROM ANOTHERTABLE;

در طرزالعمل اول نام ستون‌ها به هر تعداد که باشند توسط علامه کامه جدا شده و محتواها (Values) نیز به عین تعداد و به عین ترتیب توسط علامه کامه جدا شده و در داخل قوس‌ها نوشته می‌شود. در طرزالعمل دوم نام ستون‌ها مانند طرزالعمل یک در داخل قوس‌ها جدا و با علامه کامه نوشته شده و جهت اضافه نمودن محتواها، طرزالعمل SELECT استفاده می‌شود. کیوری انتخاب می‌تواند از هر جدول موجود در دیتابیس معلومات را گرفته و آن را به جدول یاد شده داخل نماید. حتی با استفاده از کیوری انتخاب معلومات از دیتابیس‌های دیگر که در سیستم موجود است، نیز به جدول علاوه شوند. طرزالعمل INSERT معلومات داده شده را به ترتیبی که تایپ شده باشد اخذ می‌نماید؛ یعنی محتوای یک به ستون یک، محتوای دوم به ستون دوم و به همین ترتیب تا به آخرین محتوا و ستون اجراات صورت می‌گیرد. این میتود معمولاً باعث ایجاد دو نوع غلطی می‌گردد. اشکال اول تعداد ستون‌ها کم‌تر و یا بیش‌تر از تعداد محتواها است و اشکال دوم تعداد ستون‌ها بیش‌تر از تعداد محتواهای تولیدشده توسط کیوری انتخاب است. در هر دو حالت طرزالعمل INSERT دیتا را به جدول داخل نمی‌نماید. بعضاً میتود یاد شده شماره دوم در بالا برای چند مدتی درست کار نموده و بعد اشتباهاتی رخ می‌دهد. دلیل آن بیش‌تر این است که ساختمان جدول یا جدول‌های که کیوری به اساس آن‌ها ایجاد شده و در ارتباط می‌باشند، تغییر داده شده است. این تغییر می‌تواند با آگاهی نداشتن شخص تغییردهنده از ارتباط جدول‌ها با کیوری باشد.

## داخل کردن دیتا به چندین جدول

با استفاده از طرزالعمل INSERT به چندین جدول به طور همزمان دیتا وارد شده نمی‌تواند. جهت داخل کردن عین دیتا به چند جدول باید طرزالعمل INSERT برای هر جدول به صورت جداگانه استفاده شود؛ اما امکان آن وجود دارد تا توسط این طرزالعمل دیتا از چندین جدول مختلف گرفته شده و به یک جدول در دیتابیس داخل گردد. طور مثال:

```
INSERT INTO TABLE1 SELECT *
```

```
FROM TABLE2 JOIN TABLE3;
```

در قسمت کیوری انتخاب در طرزالعمل بالا تمام شرایط یک کیوری عادی اضافه شده می‌تواند. نکته قابل ذکر این است که دیتا از جدول‌های مختلف با شرایط مختلف گرفته شده می‌تواند، اما جدولی که دیتای یاد شده به آن داخل می‌شود صرف یک جدول در یک طرزالعمل است. در صورت ضرورت داخل کردن عین دیتا به چندین جدول، طرزالعمل طرح‌شده باید بار بار به راه انداخته شود و در هر بار نام یکی از جدول‌ها در قسمت اول طرزالعمل تایپ شود.

## ۲.۴.۲ پاک کردن دیتا از جدول‌ها در سیکویل

پاک کردن دیتا از دیتابیس یکی از نکات مهم است. دیتا به کمک طرزالعمل DELETE از جدول‌های دیتابیس پاک می‌شود. به صورت عمومی در قسمت استفاده از طرزالعمل DELETE باید احتیاط کامل به خرج داده شود. یک طرزالعمل ساده برای پاک کردن دیتا از جدول، متشکل از چند کلمه محدود در یک سطر، احتمال دارد زحمت هفته‌ها و یا حتی ماه‌های یک شخص را به هدر دهد؛ و یا یک طرزالعمل ساده DELETE احتمال دارد خساره بزرگ مالی را به یک شخص یا شرکتی وارد نماید.

شکل عمومی طرزالعمل DELETE طور ذیل است:

```
DELETE FROM TABLE WHERE criteria;
```

طرزالعمل DELETE در سیکویل بدون کدام مشکل تطبیق شده و تمام ریکوردهای مشخص شده در قسمت Criteria بعد از کلمه WHERE پاک می‌شود. در سیکویل طرزالعملی مقابل DELETE وجود ندارد؛ یعنی ریکوردهای پاک شده دوباره Undo شده نمی‌توانند، یعنی با تطبیق این طرزالعمل دیتا دوباره Restore شده نمی‌تواند.

### استفاده از کلمات FROM - FROM در طرزالعمل DELETE

طرزالعمل DELETE تنها ریکوردهای یک جدول را در یک‌زمان پاک می‌نماید و این یکی از نکات مثبت طرزالعمل به شمار می‌آید. اگر ضرورت باشد تا شرط WHERE نیز در یک طرزالعمل DELETE تطبیق شود و توسط آن ریکوردهای جدول مشخص‌تر و محدودتر شوند، در چنین حالتی بعضی از سیستم‌های مدیریت

دیتابیس اجازه استفاده از کلمه کلیدی FROM را به شکل مکرر آن می‌دهند. طوری که در مثال پایین دیده می‌شود، در طرزالعمل DELETE کلمه کلیدی FROM دو بار استفاده شده است.

```
DELETEFROM TABLEA
```

```
FROM TABLEA JOIN TABLEBON TABLEA.ID = TABLEB.ID
```

```
WHERE TABLEB.Code = 23
```

با دقت به مثال بالا، کلمه FROM در عبارت اولی جدولی که دیتا از آن پاک می‌شود را مشخص نموده و FROM دومی به خاطر شناختن ریکوردها با وضع نمودن شرایط استفاده شده است. جدول‌های ذکر شده در FROM اولی باید در FROM دومی ذکر شده باشند. مثال دوم در نظر گرفته شود:

```
DELETE FROM ORDER
```

```
FROM ORDER
```

```
JOIN CUSTOMER
```

```
ON ORDER.CustomerID = CUSTOMER.CustomerID
```

```
WHERE CUSTOMER.Active = 0;
```

طوری که دیده می‌شود، جملات مثال بالا واضح بوده و ریکورد شخصی که فعال نیست پیدا شده و Order یا Order های مربوط از جدول ORDER پاک می‌شود. طرزالعمل‌های FROM-FROM در ابتداء یک اندازه، مبهم به نظر رسیده ولی با دقت نمودن به آن‌ها وضاحت موضوع نمایان می‌شود. با استفاده از این میتود، کیوری‌های قوی دیزاین و استفاده شده می‌توانند.

### حذف جدول‌ها و دیتابیس‌ها در سیکویل

یکی از طرزالعمل‌های مهم سیکویل عبارت از طرزالعمل 'DROP' است. توسط این طرزالعمل، یک دیتابیس و یا یک جدول در داخل دیتابیس کاملاً حذف شده می‌تواند، یعنی هم قسمت یوزر دیتا و هم قسمت میتادیتا یا ساختمان جدول‌ها، به صورت مکمل پاک می‌شود. پس در استفاده از این طرزالعمل باید بسیار محتاطانه عمل شود.

طور مثال، طرزالعمل یک دیتابیس به نام 'ICE2' را حذف و طرزالعمل دو جدولی با نام CUSTOMER را به اضافه تمام دیتای آن از بین می‌برد.

```
DROP DATABASE ICE2;
```

```
DROP TABLE CUSTOMER;
```

طرز العمل بالا همیشه موقّقانه اجراء نمی‌شود. اگر جدول CUSTOMER، روابطی از قبیل Referential Integrity Constraint با دیگر جدول‌ها داشته باشد که وجود آن‌ها مهم باشد، در این صورت طرز العمل بالا اجراء نشده و سیستم مدیریت دیتابیس پیام غلطی (error) خواهد داد.

در مثال دیتابیس مکتب اگر خواسته شود تا جدول کارمند از بین برده شود، این طرز العمل عملی شده نمی‌تواند. راه حل این است که یا جدول کارمند و جدول مأموریت، هر دو به ترتیب از بین برده شوند و یا هم در قسمت قاعده Referential Integrity Constraint بین این دو جدول، تغییر آورده شود. در صورت به اجراء درآوردن حالت دوم از طرز العمل ذیل استفاده می‌شود.

```
ALTER TABLE ASSIGNMENT DROP FOREIGN KEY EmployeeNum;
```

این طرز العمل، ارتباط جدول مأموریت را با جدول کارمند قطع می‌نماید. حال می‌شود طرز العمل DROP بالای جدول کارمند عملی گردیده و جدول متذکره حذف گردد؛ یعنی

```
DROP TABLE EMPLOYEE;
```

پاک کردن دیتا از جدولی در دیتابیس با استفاده از طرز العمل DELETE صورت می‌گیرد در حالی که حذف کردن جدول از دیتابیس با استفاده از طرز العمل DROP به اجراء گذاشته می‌شود. این دو عملیه از نظر مفهومی با هم تفاوت دارد. اولی آن مربوط بخش DML سیکویل بوده و دومی آن مربوط بخش DDL سیکویل می‌شود. در طرز العمل پاک کردن دیتا، جدول پاک شده ولی ساختمان و خود جدول در دیتابیس باقی می‌ماند. در طرز العمل حذف (DROP) دیتا و ساختمان جدول هر دو از بین رفته و بعد از اجراء آن بالای جدول الف، این جدول دیگر در دیتابیس نیست.



این فصل با بحث روی موضوعاتی مربوط به سیستم‌های مدیریت دیتابیس آغاز گردید. در مقدمه اهمیت لسان کیوری (سیکوئل) در سیستم‌های مدیریت دیتابیس توضیح داده شد. بخش‌بندی طرز‌العمل‌های لسان سیکوئل در سه کتگوری (طرز‌العمل‌های DML، طرز‌العمل‌های DDL و طرز‌العمل‌های DCL) با ذکر اهداف هر بخش آذین‌بخش قسمت‌های ابتدایی فصل گردیده است. مثالی از دیتابیس با داشتن سه جدول و دو ارتباط در بین جدول‌های آن در نظر گرفته شده و طرز‌العمل‌های استفاده از سیکوئل، ایجاد دیتابیس، فعال‌سازی دیتابیس و دیدن دیتابیس‌ها به ترتیب با مثال توضیح گردید.

ایجاد جدول‌ها به حیث ساختمان‌های اصلی یک دیتابیس جهت ذخیره دیتا، تعیین نوع دیتا برای فیلدهای جدول‌ها و تعریف کلیدهای اولیه از جمله بخش‌های مهم طرز‌العمل‌های DDL سیکوئل بوده که به طور مفصل توضیح و به صورت عملی نشان داده شد. ایجاد روابط بین جدول‌ها با اضافه قیدهای کلیدهای خارجی در یکی از جدول‌های پایینی نیز به شکل عملی آن نشان داده و به متن فصل اضافه شدند.

در ادامه فصل، طرز‌العمل‌های کار با دیتا یعنی DML سیکوئل با داخل کردن دیتای نمونه به جدول‌ها توضیح داده شد. پاک کردن دیتا از جدول‌های دیتابیس به کمک طرز‌العمل مشخص آن با ذکر مثال و نیز با استفاده از شرایط مختلف به توضیح گرفته شد. حذف جدول‌ها و نیز دیتابیس‌ها به حیث عناوین تکمیل‌کننده فصل، در اخیر به بحث گرفته شد. تفاوت بین پاک کردن دیتا از جدولی در دیتابیس و نیز پاک کردن ساختمان جدول از دیتابیس عبارت از دو موضوع جداگانه بوده که در قسمت اخیر فصل به آن پرداخته شده است.



## سوالات فصل دوم

۱. سیکویل (SQL) اختصار کدام کلمات بوده و در کجا استفاده می‌شود؟
۲. طرزالعمل‌های DML سیکویل را با مثال توضیح دهید.
۳. طرزالعمل‌های DDL سیکویل را با مثال توضیح دهید.
۴. طرزالعمل‌های DCL سیکویل را با مثال توضیح دهید.
۵. سیستم‌های مدیریتی دیتابیس که از لسان سیکویل استفاده می‌نمایند را نام بگیرید.
۶. اساسی‌ترین واحد در دیتابیس که در آن دیتا ذخیره می‌شود به چه نام یاد می‌شود؟
۷. جهت ایجاد نمودن یک فایل متنی (Logfile) و ثبت طرزالعمل‌های انجام آن در کوماندلاین سیکویل از کدام طرزالعمل استفاده می‌شود؟
۸. برای ایجاد دیتابیس و جدول در آن، از کدام طرزالعمل‌ها استفاده می‌شود؟
۹. آیا استفاده از نام دیتابیس در طرزالعملِ فعال‌سازی دیتابیس ضروری است؟ اگر باشد در مثال نشان دهید.
۱۰. قسمت‌هایی که توسط آن‌ها یک ستون (Column) در جدول دیتابیس تعریف می‌گردد را نام بگیرید.
۱۱. استفاده از قیدهای ناشناخته (Null) و غیر ناشناخته (Not Null) چه تفاوت دارد؟
۱۲. تفاوت انواع دیتای Char و VarChar در چه است؟
۱۳. جهت ایجاد ارتباط بین دو جدول حد اقل چند فیلد و کدام فیلدها ضرورت است؟ با مثال توضیح دهید.
۱۴. طرزالعمل Drop Table چه زمانی استفاده می‌شود؟
۱۵. آیا به کمک طرزالعمل INSERT بخش یوزردیتا در سیکویل تغییر داده می‌شود؟ اگر می‌شود، توضیح دهید.
۱۶. دیتا که با استفاده از طرزالعمل DELETE از یک جدول دیتابیس پاک می‌شود، توسط کدام



۱. برنامهٔ سیکویل را در کوماندا لاین اجرا کنید.
  ۲. دیتابییسی تحت عنوان فروشگاه ایجاد نموده و فعال سازید.
  ۳. جدول های موجود در دیتابیس فروشگاه را نشان دهید.
  ۴. در دیتابیس فروشگاه دو جدول به نام های مشتری (Customer) و فروشات (Sales) با بیش تر از سه فیلد ایجاد کنید.
  ۵. دیتای نمونه حد اقل 4 ریکورد را به هر جدول اضافه کنید.
  ۶. ارتباط بین هر دو جدول را طوری ایجاد کنید که یک مشتری بتواند چندین خریداری انجام دهد (شما می توانید کلید اولیه جدول بالایی را در جدول پایینی اضافه کنید).
- نوت:** در فعالیت های فوق شما می توانید از طرز العمل های ذیل استفاده کنید:

- Create database ... •
- Create table ... •
- Show databases •
- Show tables •
- Use database ... •
- Alter table ... •
- Insert into ... •
- Delete from ... •
- Drop table ... •
- Drop database ... •

## فصل سوم

### ترانزکشن‌ها (Transactions) در دیتابیس



**هدف کلی:** محصلان با مفهوم و استفاده عملی ترانزکشن (Transaction) آشنا شوند.

**اهداف آموزشی:** در پایان این فصل محصلان قادر خواهند شد تا:

۱. ترانزکشن را تعریف نموده، خصوصیات ACID و حالت‌های یک ترانزکشن را بفهمند.
۲. شیوه استفاده از ترانزکشن و دانستن حالت‌های یک ترانزکشن را بیان نمایند.
۳. مراحل مختلف انزواء و ناهنجاری‌های ممکنه آن را توضیح دهند
۴. ترانزکشن را به صورت عملی در سیستم مدیریت دیتابیس به کار گیرند.



دیتابیس‌ها عبارت از سیستم‌های ثبت معلومات به شکل کمپیوتری بوده و اصطلاح معلومات کمپیوتری به همان شکل دیجیتالی یا رقم معلومات اطلاق می‌گردد. به گونه‌ی مثال معلومات مالیات یک ساله یک مملکت در نظر گرفته شده و به صورتی تنظیم شود تا در یک کمپیوتر با استندردهایی ذخیره شود، سیستمی که این مجموعه معلومات را ذخیره و قابل استفاده کرده بتواند به نام دیتابیس یاد می‌شود. پس تعریف دیتابیس کمی بیش‌تر وسعت داده شده چنین نگاشته می‌شود: دیتابیس عبارت از مجموعه تعریف‌شده داخلی عناصر دیتا و اجزای مربوطه آن بوده طوری که هر جزء دیتا در ساختمانی به نام جدول ذخیره شده و دوباره قابل استفاده باشد.

جدول‌های دیتابیس‌های ارتباط به اساس مدل ارتباطی (The Relational Model) طوری دیزاین می‌شوند که هر ستون دارای یک نام و نوع مشخص دیتا (Data Type) بوده، هر ریکورد یک مثال از دیتای قابل ذخیره در جدول را در بر داشته باشد. خود جدول نیز به یک نام و مشخصات تعیین‌شده در داخل دیتابیس، ایجاد می‌شود. ساختمان‌های دیگری مانند ارتباط بین جدول‌ها، نماها (Views)، قیدها (Constraints)، کیوری‌ها و غیره در یک دیتابیس دیزاین و استفاده می‌شوند. نام دیتابیس، نام و مشخصات جدول‌ها و ساختمان‌های دیگر در داخل دیتابیس به نام میتادیتا (Metadata) یاد می‌شود. دیتای دومی به نام یوزردیتا (Userdata) یاد می‌شود. یوزردیتا عبارت از معلومات تحلیل‌شده یا همان عناصر دیتا است که توسط استفاده‌کننده تهیه و به دیتابیس ذخیره می‌شود. در یک جدول به نام شاگرد اگر معلومات دو نفر شاگرد به نام‌های احمد، صنف 3، چهارساله و محمود، صنف 4، پنج‌ساله درج باشد:

- قسمت میتادیتا عبارت از نام جدول (شاگرد)، نام فیلدها (نام، صنف، سن) و انواع دیتاهای قابل ذخیره در هر فیلد.
- قسمت یوزردیتا همان معلومات در مورد شاگردان (احمد، صنف 3، چهارساله و محمود، صنف 4، پنج‌ساله) است.

یوزردیتا به جدول‌های دیتابیس داخل شده می‌تواند و به عین شکل همین یوزردیتا از جدول‌های دیتابیس پاک شده می‌تواند؛ و بالاخره یوزردیتا در جدول‌های دیتابیس قابل تغییر و به‌روزرسانی خواهد بود. انجام عملیه‌های داخل کردن، پاک کردن و تغییردادن بالای یوزردیتا در دیتابیس‌ها توسط استفاده‌کنندگان صورت می‌گیرد. یک دیتابیس امکان دارد چندین استفاده‌کننده را در یک زمان داشته باشد. در دیتابیس‌های بزرگ امکان صدها و هزارها استفاده‌کننده در یک‌زمان امکان‌پذیر است. به خاطر کنترل استفاده هم‌زمان توسط استفاده‌کنندگان مختلف یک دیتابیس، عملیه‌هایی به نام ترانزکشن (Transaction) در سیستم‌های مدیریت دیتابیس (DBMS)‌ها در نظر گرفته شده است. مدیریت ترانزکشن باعث می‌شود تا کارهای داخل کردن، پاک کردن و تغییر آوردن در دیتا بین استفاده‌کنندگان تداخل ننماید.

در این فصل به ترانزکشن‌ها به حیث ساختمان‌های مهم مدیریت دیتا در سیستم‌های مدیریت دیتابیس پرداخته شده است. به خصوصیات ترانزکشن‌ها، موارد استفاده آن‌ها، مراحل پایان‌دادن ترانزکشن‌ها، مراحل انزواء و ناهنجاری‌های ممکنه، استفاده ترانزکشن‌ها در سیکویل با چند مثال عملی پرداخته شده است.

### ۳.۱ ترانزکشن چیست؟ (What is a Transaction?)

ترانزکشن عبارت از یک برنامه اجرایی یا پروسه‌یی است که توسط آن یک یا چندین نوع کار از قبیل خواندن، تغییرآوردن و یا پاک کردن دیتای موجود و یا هم داخل کردن دیتای جدید به دیتابیس صورت گرفته بتواند. به عبارت دیگر دسته‌یی از کارها و طرزالعمل‌هایی را که در یک جلسه (Session) سیکویل بالای یوزردیتا انجام می‌شوند به نام ترانزکشن یاد می‌نمایند. یک کار عبارت از آخرین تقسیم بخش کود توسط استفاده‌کننده بوده که آن قطعه کم‌ترین تعداد پروسه‌ها را در بر داشته باشد؛ یعنی یک ترانزکشن می‌تواند حد اقل یک طرزالعمل اجرایی را در بر داشته باشد. طرزالعمل داخل کردن دیتا به جدول و یا طرزالعمل پاک کردن دیتا از جدولی در یک دیتابیس را می‌توان کار در سیکویل نامید. دسته‌یی از طرزالعمل‌ها تحت نام ترانزکشن توسط استفاده‌کننده، کنترل شده و به گونه مثبت استفاده می‌شوند. با استفاده ترانزکشن، یک استفاده‌کننده می‌تواند کار بالای دیتابیس را بدون از دست دادن دیتا و حفظ کردن تغییرات کنترل شده، بالای دیتابیس انجام دهد.

در ادامه یک ترانزکشن عبارت از قطعه‌یی از کار روی دیتا است که این قطعه کار یا باید به صورت کامل انجام شود و یا هم باید به صورت کامل از تطبیق آن جلوگیری شده و دیتابیس به حالت اولی خود گذاشته شود؛ یعنی دسته‌یی از طرزالعمل‌ها که در یک ترانزکشن دسته‌بندی شده باشند، نباید به طور قسمی اجراء شوند، بلکه مجموعه طرزالعمل‌ها به شکل یک قطعه مکمل مدیریت شود. به خاطر اهداف بازیابی (Recovery)، سیستم مدیریت دیتابیس ضرورت به حفظ مسیرهای کار روی دیتا را دارد. یک سیستم جزئیات کار روی دیتا توسط استفاده‌کنندگان را ذخیره می‌نماید. در زمان تنظیم و اجرای ترانزکشن مسایلی که توسط سیستم ذخیره می‌شوند عبارت از آغاز ترانزکشن و پایان ترانزکشن بوده و شامل تطبیق مؤفقانه ترانزکشن و یا کنسل شدن ترانزکشن‌اند. پس گفته می‌شود که بخش مدیریت بازیابی سیستم مدیریت دیتابیس (DBMS) ضرورت به ثبت عملیات‌های زیر است:

- **آغاز ترانزکشن:** این طرزالعمل آغازکننده ترانزکشن بوده و با طرزالعمل (Start Transaction) اجراء می‌شود.
- **خواندن و یا نوشتن:** عملیات خواندن و نوشتن دیتا (کیوری و داخل کردن دیتا) در دیتابیس اجراء می‌شوند. این طرزالعمل‌ها هدف اساسی بخش اجرایی یک ترانزکشن را در بر دارند.
- **ختم ترانزکشن:** طرزالعمل‌های پایانی ترانزکشن در این کتگوری اضافه می‌شوند. طرزالعمل‌های پایانی در حقیقت پایان طرزالعمل‌های اصلی خواندن و نوشتن دیتا را نشان می‌دهند. در همین قسمت تصمیم موفقیت و عدم موفقیت یک ترانزکشن گرفته می‌شود؛ یعنی نتیجه اجرایی طرزالعمل‌ها (تغییرات دیتا) در دیتابیس به صورت دائمی تطبیق شده و یا از تغییرات دیتا جلوگیری به عمل آید. در حالت یک تغییرات دائمی وقتی در دیتا اضافه شوند، این تغییرات قابل برگشت (Undo) نمی‌باشند. حالت‌های تأیید و رد تغییرات دیتا که به اساس ترانزکشن باید واقع شوند توسط طرزالعمل‌های پایانی Commit و Rollback طور ذیل به اجراء گذاشته می‌شوند:

- تسلیم‌شده (Commit): در صورت تأیید کارهای اجراء شده در ترانزکشن، استفاده‌کننده می‌تواند نتیجه ترانزکشن را تأیید نموده و با استفاده از کلمه Commit تغییرات دائمی و غیرقابل برگشت بسازد.
- رد شده (Rollback or Abort): استفاده از این طرزالعمل در صورتی قابل اجراء است که کدام مشکلی در قسمت تطبیق ترانزکشن باشد، استفاده‌کننده و یا در حالاتی سیستم این مسئله را تشخیص می‌دهد که تغییرات باید اجازه داده نشود. در صورت رد شدن نتیجه ترانزکشن، تمام تغییرات خواندن و نوشتن دیتابیس کنسل شده و دیتابیس حالت قبل از آغاز ترانزکشن را به خود می‌گیرد.

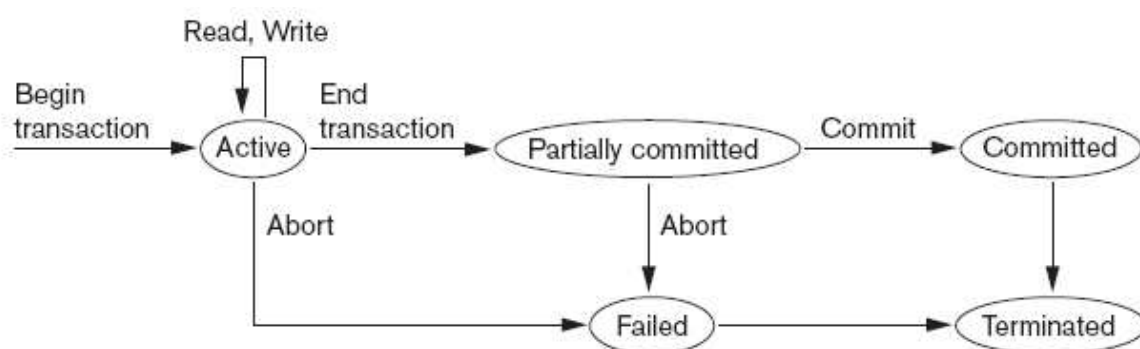
### ۳.۱.۱ حالت‌های یک ترانزکشن

یک ترانزکشن قابلیت تکمیل و یا رد شدن را دارد. به گونه‌یی که نتیجه اجرایی ترانزکشن باید یک حالت مشخص را داشته باشد. حالت‌های احتمالی تکمیل یک ترانزکشن به طور ذیل بوده می‌توانند:

- فعال (Active): در مرحله اول یک ترانزکشن و جریان اجرایی ترانزکشن، حالت فعال بودن آن را نشان می‌دهد.
- قسمی تسلیم‌شده (Partially Committed): بعد از اجرای آخرین طرزالعمل ترانزکشن (در حالی که اضافه از یک طرزالعمل شامل ترانزکشن باشد)، حالت نیمه تسلیم آن طرزالعمل‌ها را نشان می‌دهد.
- تسلیم‌شده (Committed): حالت بعد از اجرای مؤفقانه طرزالعمل و یا طرزالعمل‌های شامل ترانزکشن را نشان می‌دهد.
- نامؤفق‌شده (Failed): حالتی است که در آن بعد از دانستن حالتی که اجرای عادی ادامه نخواهد داشت، مشخص می‌شود.
- ناتمام‌مانده (Aborted): بعد از عقب‌کشیدن ترانزکشن و بازگرداندن (ریستورکردن) دیتابیس به حالت قبل از آغاز ترانزکشن گفته می‌شود. در این حالت از اختیار دوباره آغازکردن ترانزکشن و یا هم از بین‌بردن کامل آن استفاده شده می‌تواند.

شکل پایین یک دیاگرام انتقال حالت اول یک ترانزکشن در حال اجراء نشان می‌دهد. یک ترانزکشن به مجرد اجرای طرزالعمل اول آن یعنی (Start Transaction) به حالت فعال می‌رود. در حالت فعال طرزالعمل‌های خواندن و نوشتن دیتا تطبیق می‌شوند. با پایان‌یافتن ترانزکشن حالت تسلیم‌دهی قسمی (Partially Committed) واقع می‌شود. در این مرحله تعدادی از پروتوکول‌های بازیابی (ریکوری) سیستم مدیریت دیتابیس، تغییرات مؤقت دیتا را بررسی می‌کنند تا در صورت کنسل‌شدن ترانزکشن، تغییرات دائمی به دیتابیس وارد نشود. زمانی که این بررسی تکمیل شد و موفقیت عملیات‌های ترانزکشن تأیید شدند، حالت ترانزکشن به مرحله بعدی (Committed) پیش می‌رود. بعد از این مرحله ترانزکشن به مرحله نهایی تکمیل مؤفقانه خود رسیده و تغییرات جدید به شکل دائمی بالای دیتا تطبیق می‌شوند.

یک ترانزکشن امکان دارد بعد از مرحله تسلیم‌دهی قسمی، به حالت ناتکمیل شده برود (به خاطر ناتکمیل ماندن یک ترانزکشن تفصیلات بیش‌تر در عناوین بعدی موجوداند). در صورت بروز مشکل در یکی از بررسی‌ها به علت‌های مختلف و یا اگر یک ترانزکشن از طرف استفاده‌کننده، کنسل یا Abort شود ترانزکشن به حالت ناتمام یا Failed می‌رود. در این حالت ترانزکشن باید رد یا Rollback شود تا تغییرات مؤقت نوشتاری دیتا پس گرفته شود و دیتابیس به حالت اول خود نگه داشته شود. در نهایت حالت خاتمه‌دادن (Terminated) ترانزکشن حالتی است که جزئیات مربوط به کار ترانزکشن، از جدول‌های سیستم که معلومات یاد شده را به صورت مؤقت در خود ذخیره نموده بودند، پاک شود. ترانزکشن‌هایی که تکمیل نشده و یا رد شده‌اند شاید دوباره تطبیق شوند. تطبیق دوباره آن‌ها توسط سیستم، قابل انجام است و یا هم استفاده‌کننده ترانزکشن‌های مورد ضرورتش را دوباره به شکل ترانزکشن‌های جدید به اجرا درآورده می‌تواند.



شکل ۱-۳ دیاگرام انتقال حالت یک ترانزکشن در حال اجرا

به خاطر وضاحت بیش‌تر موضوع یک مثال از یک ترانزکشن ساده در نظر گرفته شود. یک کارمند بانک مبلغ 5000 افغانی را از حساب مشتری الف به حساب مشتری ب انتقال می‌دهد. این ترانزکشن ساده و کوچک شامل چندین کار در سیستم است. در جدول پایین کارهای اجراء شده در این ترانزکشن فهرست شده‌اند.

A's Account (حساب مشتری الف)	B's Account (حساب مشتری ب)
Open_Account (A)	Open_Account(B)
Old_Balance = A.balance	Old_Balance = B.balance
New_Balance = Old_Balance - 5000	New_Balance = Old_Balance + 5000
A.balance = New_Balance	B.balance = New_Balance
Close_Account(A)	Close_Account(B)

جدول ۱-۳ تفصیل طرزالعمل‌های اجراء شده در یک ترانزکشن ساده انتقال پول از حساب یک مشتری به حساب مشتری دیگر

در مثال ترانزکشن بالا عملیه‌های خواندن دیتا (Read) و نوشتن دیتا (Write) ضرورت به اجراء خواهند داشت. طرزالعمل‌های خواندن به خاطر گرفتن دیتا از دیتابیس معلومات شخص الف با استفاده از کیوری

انتخاب اجراء می‌شوند. به عین ترتیب طرزالعمل‌های خواندن دیتا به خاطر تغییر آوردن در دیتا عبارت از طرزالعمل‌های تغییر دیتابیس مانند داخل کردن دیتا، به‌روز کردن دیتا و پاک کردن دیتا خواهند بود. هر اجرای یک برنامه در سیکویل را می‌توان ترانزکشن نامید. هر ترانزکشن می‌تواند دارای پارامترهای مختلف باشد. پارامترهای ترانزکشن انتقال پول بین حساب‌های بانکی عبارت‌اند از: ۱- نمبر حساب مشتری الف ۲- حساب مشتری ب و پارامتر ۳- مقدار پولی که باید انتقال شود.

در سیکویل طرزالعمل‌های کنترل ترانزکشن موجود بوده، استفاده‌کننده دیتابیس می‌تواند با به کار گرفتن این نوع طرزالعمل‌ها ترانزکشن‌های کار روی یوزردیتا را به صورت درست آن استفاده نماید. طرزالعمل‌های کنترل ترانزکشن به خاطر مشخص ساختن بخش‌هایی از پروسس دیتا که برای کنترل و استفاده هم‌زمان دیتا و بازیابی (ریکوری) دیتا به کار می‌روند، استفاده می‌شوند.

### ۳.۱.۲ خصوصیات یک ترانزکشن

یک ترانزکشن باید خصوصیات استندردی را تحت نام مختصر **ACID** دارا باشد. اصطلاح **ACID** از چهار کلمه (**Atomicity**، **Consistency**، **Isolation** و **Durability**) گرفته شده است. این خصوصیات باید توسط سیستم مدیریت دیتابیس به خاطر کنترل دیتای هم‌زمان (**Concurrency**) و مسایل بازیابی دیتا قابل تطبیق باشند. توضیح مختصر کلمات یاد شده طور ذیل است:

- **Atomicity**: تمام کارهای پلان شده در یک ترانزکشن به شکل یک مجموعه باید تکمیل شوند، در غیر این صورت ترانزکشن به کلی حذف شود.
  - **Consistency**: یک ترانزکشن دیتابیس را در حالت غیر سازگار (**Inconsistent**) رها نکند؛ یعنی زمانی که یک ترانزکشن بدون کدام مشکل به صورت مؤفّقانه اجراء می‌شود، بعد از تکمیل آن، دیتابیس دوباره در حالت سازگار (**Consistent**) گذاشته شود.
  - **Isolation**: هر ترانزکشن دیتابیس را به صورت مستقل صورت گرفته و یکی به دیگری تداخل ندارد. این در حالی باید تطبیق گردد که امکان اجرای چندین ترانزکشن در یک‌زمان باشد. هر ترانزکشن باید کاملاً بدون تأثیر بالای ترانزکشن دیگری که در حال اجراء است تطبیق شده بتواند.
  - **Durability**: در صورت بروز هر نوع مشکل یا نقیصه (**Failure**)، ترانزکشن‌های تکمیل‌شده در یک دیتابیس باید بدون تغییر و به حال خود حفظ شوند تا زمانی که سیستم دوباره نورمال شود. بعد از فعال شدن سیستم، ترانزکشن‌های حفظ‌شده به اجراء گذاشته شوند. این امر باید قابل تطبیق باشد حتی اگر کمپیوترهای سرور خاموش و دوباره روشن شوند.
- طوری که دیده شد، خصوصیات فوق بالای ترانزکشن‌ها تطبیق می‌شوند. ترانزکشن‌ها در دیتابیس به خاطر کار روی دیتا استفاده می‌شوند. فعالیت‌های **ACID** در یک دیتابیس که چندین استفاده‌کننده، در یک‌زمان در آن کار می‌نماید به خاطر حفظ سلامت دیتا و اطمینان از دقت کار یکی از صفات اساسی سیستم مدیریت دیتابیس به شمار می‌رود.

### ۳.۲ شریک ساختن دیتا و کنترل ترانزکشن دیتا با چندین استفاده کننده

یک سیستم مدیریت دیتابیس (DBMS) با چندین استفاده کننده معمولاً به خاطر مدیریت استفاده کنندگان و دیتابیس های مربوطه آن به یک سلسله مسایل تخنیکی جهت کنترل استفاده همزمان دیتابیس ها ضرورت دارد. طوری که از نام سیستم های مدیریت دیتابیس چندین استفاده کننده پیداست، در چنین سیستم ها اضافه از یک نفر در یک زمان به دیتا و دیتابیس ها شریک ساخته شده، اجازه دسترسی داده می شود. تنظیم و ذخیره دیتا در یک دیتابیس به هدف استفاده در اپلیکیشن های مختلف در ساحة سیستم های معلوماتی و کمپیوترساینس یک امر ضروری پنداشته می شود. سیستم مدیریت دیتابیس باید نرم افزاری را جهت کنترل استفاده همزمان دیتابیس داشته باشد. داشتن چنین کارکردی در نرم افزار به خاطر اطمینان از صحت عین ست دیتا که توسط استفاده کنندگان مختلف در یک زمان استفاده شده و تغییراتی در آن وارد شود، حتمی است.

طوری که در مثال درس قبلی نیز دیده شد، زمانی که چندین نماینده یک شرکت هوایی می خواهند یک سیت را در یک پرواز به مشتری ریزرف نمایند، سیستم مدیریت دیتابیس باید کار را طوری تنظیم نماید تا یک سیت معین در یک پرواز معین تنها به یک مسافر داده شود. هم چون اپلیکیشن ها به اختصار به نام OLTP یا (Online Transaction Processing) یاد شده و به خاطر تنظیم مسایل مربوط به کار گرفته می شوند. دیتابیس های OLTP سیستم های وسیع چندین استفاده کننده ای اند که هزاران ترانزکشن را در یک زمان (مثلاً یک دقیقه) اجازه داده و بدون اشتباه کنترل می کنند. رُل اساسی سیستم های مدیریت دیتابیس چندین استفاده کننده در اطمینان از درست بودن و مؤثر بودن ترانزکشن های اجراء شده به صورت همزمان و موازی است. مفهوم ترانزکشن و کنترل آن از موضوعات مرکزی بیش تر اپلیکیشن های دیتابیس ها به شمار می رود.

### ۳.۳ استفاده از ترانزکشن (Use of Transaction)

طوری که گفته شد، یک ترانزکشن عبارت از سلسله اجرایی طرز العمل ها روی دیتا در دیتابیس است. ترانزکشن در محیطی قابلیت اجراء را دارد که در آن کمپیوتر موجود بوده و نرم افزار استفاده از طرز العمل های دیتابیس مانند دیدن دیتا، داخل کردن دیتا، پاک کردن دیتا و تغییر آوردن در دیتا موجود باشد. یک ترانزکشن می تواند به دو شکل برنامه ریزی و استفاده شود:

- به صورت مستقل: یک ترانزکشن می تواند توسط یک برنامه سطح بالا مانند سیکویل مستقیم نوشته شده و به اجراء درآید.
- به صورت ضمیمه: یک ترانزکشن می تواند در یک برنامه ضمیمه شود طوری که در زمان اجرای برنامه، ترانزکشن نیز در محدوده خود به اجراء درآورده می شود.

مثال ترانزکشن مستقل در عنوان قبلی در قسمت انتقال 5000 افغانی از حساب بانکی مشتری الف به حساب مشتری ب بود که با تفصیل ذکر گردید. اینک مثالی از یک ترانزکشن ضمیمه در بخشی از کود برنامه یی ارائه می شود. در این قطعه کود، ترانزکشن آغاز (begin) و انجام (end) داشته و از طریق آن مقدار 10 فیصد بودجه ریکوردهای پروژه یی به نام CAD/CAM به جدولی به نام پروژه اضافه می گردد.

Transaction BUDGET\_UPDATE

begin

EXEC SQL

UPDATE PROJECT

SET BUDGET = BUDGET \* 1.1

WHERE PNAME = "CAD/CAM"

end.

طوری که در مثال بالا نیز دیده می‌شود، مسئله دیگری در یک ترانزکشن عبارت از آغاز (Begin Transaction) و انجام ترانزکشن (End Transaction) بوده که سرحدات عملیه‌های داخل ترانزکشن را تعیین می‌نماید؛ یعنی با آغازشدن ترانزکشن، طرزاعمال‌های سیکویلی که اجراء می‌شوند شامل ترانزکشن شده تا زمانی که طرزاعمال بسته‌شدن ترانزکشن صادر نگردیده باشد، مجموعه همه طرزاعمال‌ها تحت همان ترانزکشن قابل قبول و یا رد می‌باشد. برنامه‌های اپلیکیشن‌ها که ترانزکشن‌های مختلف را استفاده می‌نمایند، می‌توانند هر ترانزکشن را به صورت جداگانه مورد پروسس قرار دهند. ولی جلسه‌های استفاده مستقیم سیکویل می‌تواند تنها ترانزکشن در حال اجراء را کنترل کند. کود ترانزکشن می‌تواند برای چندین بار در حلقه‌های (Loops) در برنامه‌ها به کار گرفته شود.

یک مثال دیگری در استفاده از ترانزکشن توسط دیتابیس شرکت هوایی در نظر گرفته شود. دیتابیس شرکت هوایی جدول‌های پرواز (FLIGHT)، مشتری (CUST) و حالت پرواز (FC) را با تفصیل ذیل دارا است.

FLIGHT (FNO, DATE, SRC, DEST, STSOLD, CAP)

CUST (CNAME, ADDR, BAL)

FC (FNO, DATE, CNAME, SPECIAL)

در این دیتابیس به خاطر ریزرف کردن یک تکت، نماینده شرکت هوایی معلومات ذیل را وارد می‌نماید: شماره پرواز (FNO)، تاریخ (DATE) و نام مشتری (CNAME)؛ و بعد از داخل نمودن معلومات، تقاضای ریزرف کردن یک سیت پرواز را می‌نماید. به خاطر انجام دادن این کارها ترانزکشن به شکل پایین دیزاین شده می‌تواند.

Begin transaction Reservation

begin

input (flight\_no, date, customer\_name);

EXEC SQL UPDATE FLIGHT

SET STSOLD = STSOLD + 1

WHERE FNO = flight\_no AND DATE = date;

```

EXEC SQL INSERT
      INTO FC (FNO, DATE, CNAME, SPECIAL);
      VALUES (flight_no, date, customer_name, null);
output ("reservation completed")
end.

```

در ادامه مثال، یک ترانزکشن باید نقطه انجام (نتیجه) داشته باشد؛ یعنی ترانزکشن بالا یا باید تکت ریزرف نماید و یا هم اگر سیت در پرواز به پایان رسیده باشد، دیتا را به سیستم داخل نموده و جواب رد بدهد. به این منظور ترانزکشن بالا با اضافه کردن بخش‌های ذیل آن را مکمل‌تر انجام دهد.

```

Begin transaction Reservation
begin
input (flight no, date, customer name);
EXEC SQL SELECT STSOLD, CAP
      INTO temp1, temp2
      FROM FLIGHT
      WHERE FNO = flight no AND DATE = date;
if temp1 = temp2 then
      output ("no free seats");
      Abort
else
      EXEC SQL UPDATE FLIGHT
      SET STSOLD = STSOLD + 1
      WHERE FNO = flight no AND DATE = date;
      EXEC SQL INSERT
      INTO FC (FNO, DATE, CNAME, SPECIAL);
      VALUES (flight no, date, customer name, null);
      Commit
      output ("reservation completed")
endif
end.

```

طوری که در قطعه کود بالا دیده می‌شود، در قسمت اول حالتی است که سیت خالی وجود ندارد و این را شرط IF در کود بالا پیدا می‌نماید و باعث می‌شود تا ترانزکشن عقب کشیده شده (Abort) و ختم گردد. در حالت دوم (اگر شرط IF نادرست باشد) جدول پرواز به‌روز (Update) شده و به جدول حالت پرواز نیز دیتا



داخل گردیده و ترانزکشن مؤفقانه تکمیل (Commit) می‌گردد؛ و در نهایت متن تکمیل ریزرف‌شدن سیت برای مشتری صادر می‌شود.

### ۳.۳.۱ پایان ترانزکشن

هر ترانزکشن باید یک پایان روشن و با نتیجه داشته باشد. هر زمانی که یک ترانزکشن در سیستم مدیریت دیتابیس به خاطر اجراءشدن از آن استفاده می‌شود، سیستم مدیریت دیتابیس مسئول تکمیل ساختن پروسه و رسیدن به نتیجه است. در صورت مؤفق‌بودن نتیجه، تمام طرزالعمل‌های داخل ترانزکشن، مثبت شده و تمام تغییرات وارده به دیتا در دیتابیس ثبت دائمی می‌گردند. به عین ترتیب در صورت عدم موفقیت تمام طرزالعمل‌ها یا قسمتی از طرزالعمل‌های داخل ترانزکشن، منفی شده و هیچ تغییری به دیتای دیتابیس وارد نمی‌شود.

در حالت دوم، دیتابیس به شکل اولی خود حفظ می‌گردد. نتیجه یک ترانزکشن چه منفی باشد و یا مثبت، خالی از دو حالت نیست: تکمیل‌شدن مؤفقانه و عدم تکمیل‌شدن مؤفقانه. در حالت اول اصطلاح Commit به آن استفاده شده و به حیث یک طرزالعمل عمل می‌نماید. در چنین حالتی تغییراتی که در جریان ترانزکشن به یوزردیتا در دیتابیس وارد شده، عملی شده و برای دائم در دیتابیس ذخیره می‌گردند. حالت دوم (عدم تکمیل مؤفقانه) به نام Abort یاد می‌شود. در این حالت ترانزکشن تکمیل نشده و تمام طرزالعمل‌های آن رد می‌شوند؛ یعنی تغییراتی در یوزردیتای دیتابیس وارد نمی‌شود.

اطمینان از اضافه‌شدن تغییرات به دیتابیس با تطبیق عملیة Commit و نیز اطمینان از بودن دیتابیس در حالت قبل از آغاز ترانزکشن با تطبیق عملیة Abort از شرط‌های استفاده از ترانزکشن به شمار می‌آید؛ یعنی یک سیستم مدیریت دیتابیس این شرایط را باید به صورت درست تطبیق نماید. یک ترانزکشن می‌تواند حد اقل یک طرزالعمل اجرایی در خود داشته باشد و حد اکثر تعداد طرزالعمل‌های شامل یک ترانزکشن تعیین نشده است. تمام طرزالعمل‌های داخل یک ترانزکشن به حیث یک ست یا مجموعه در نظر گرفته می‌شوند. نتیجه یک ترانزکشن به حیث یک واحد منفی و یا مثبت در کار، روی دیتا در دیتابیس‌ها در نظر گرفته می‌شود. با در نظر داشت این اصل، سیستم مدیریت دیتابیس باید بخش‌های فرعی داخل یک ترانزکشن را الی پایان ترانزکشن اجازه اجراء ندهد. این به خاطری است که اگر نتیجه یک ترانزکشن منفی باشد، دیتابیس باید به حالتی که قبل از ترانزکشن داشت، برگردانده شود. در صورت آوردن تغییرات قسمی به اساس بخش‌های ترانزکشن، این اصل زیر پا شده و باعث ایجاد مشکلات به استفاده‌کنندگان دیگر که در عین زمان مصروف استفاده از دیتابیس‌اند، خواهد گردید. پس یکی از عللی که نتیجه پایانی یک ترانزکشن صرف بعد از اجرای مؤفق آن باید تطبیق گردد، همین دلیل بوده می‌تواند.

### ۳.۴ نقیصه‌ها یا خرابی‌ها در ترانزکشن (Transaction Failure Types)

در ترانزکشن‌ها امکان عدم تکمیل پروسه‌ها و تطبیق کامل آن‌ها وجود دارد. دلایل عدم تکمیل یک ترانزکشن به نام نقیصه ترانزکشن (Transaction Failure) یاد می‌شود. نقیصه‌ها به صورت عمومی به سه بخش تقسیم

می‌شوند: نقیصه‌های ترانزکشن، نقیصه‌های سیستم و نقیصه‌های میدیا. دلایل احتمالی که باعث نقص در تکمیل‌شدن و اجراءشدن یک ترانزکشن در وسط اجرای آن می‌گردند عبارت‌اند از:

- نقص کمپیوتر (Computer Failure) یا خرابی سیستم: مشکل سخت‌افزار، نرم‌افزار و یا اشتباه شبکه در یک کمپیوتر در زمان اجرای یک ترانزکشن واقع شده می‌تواند. مشکل‌های سخت‌افزار در زمان‌های زیادی، برمی‌گردد به درست کارنکردن پرزه‌جات کمپیوتر مانند کارنکردن حافظه اصلی کمپیوتر در جریان یک ترانزکشن که در آن دسته‌یی از طرزالعمل‌ها باید اجراء شوند و یک نتیجه واحد داشته باشند.
- اشتباه در ترانزکشن و یا سیستم (Transaction or System Error): بعضی عملیات‌ها در داخل ترانزکشن امکان دارد باعث عدم تکمیل‌شدن ترانزکشن شود. سرریزشدن (Overflow) یک عدد و یا تقسیم‌شدن یک عدد بالای صفر می‌تواند ترانزکشن را به سوی عدم تکمیل‌شدن هدایت نماید. استفاده از پارامترهای نادرست در طرزالعمل‌ها و نیز اشتباهات منطقی برنامه‌نویسی، امکان پیداکردن نقیصه در تکمیل‌شدن ترانزکشن را به بار خواهد آورد. مداخله استفاده‌کننده نیز باعث قطع پروسس یک ترانزکشن شده که در نهایت باعث عدم تکمیل ترانزکشن خواهد شد.
- اشتباهات محلی (Local) یا حالات استثنایی (Local Errors or Exception Conditions): در جریان اجرای ترانزکشن، حالت‌هایی واقع می‌شوند که باعث کنسل‌شدن (Cancel) حتمی ترانزکشن می‌شوند. به طور مثال دیتای مورد ضرورت یک ترانزکشن از موقعیت تعیین‌شده در دیتابیس آن پیدا نمی‌شود، یا حالتی واقع شود که بیلانس پول موجود در حساب بانکی یک مشتری کم‌تر از مقدار درخواست‌شده به خاطر گرفتن پول توسط مشتری از حساب بانکی‌اش باشد. در چنین حالتی نیز ترانزکشن تکمیل‌نشده و در مقابل باید پیام معلوماتی به استفاده‌کننده ارائه شود. (حالت دوم امکان دارد در خود ترانزکشن توسط طرزالعملی کنترل شود. در هم‌چو حالت دلیل کنسل‌شدن ترانزکشن را غلطی (ERROR) نپنداشته، بلکه بخشی از پروسه‌های داخل ترانزکشن گفته می‌شود).
- کنترل پروسه‌های هم‌زمان (Concurrency Control Enforcement): پروسه‌های هم‌زمان که توسط استفاده‌کنندگان دیگر در عین قسمت دیتابیس انجام می‌شود، شاید از طرف سیستم مدیریت دیتابیس طوری تصمیم گرفته شود تا یکی از ترانزکشن‌های در حال اجراء توقف داده شده و Abort شود. این عمل به خاطری است که مشکل در تنظیم دیتا به وجود نیامده و پروسه‌ها تحت کنترل باشند. در هم‌چون حالت‌ها به یک یا چند ترانزکشن در حال اجراء، توقف اجباری داده شده و باعث از بین‌بردن بن‌بست‌ها (Deadlocks) بین ترانزکشن‌ها خواهد گردید. ترانزکشن‌هایی که از هم‌چون پروسه متأثر شده و توقف داده می‌شوند معمولاً بعدتر به صورت خودکار دوباره از سر گرفته شده و تطبیق می‌شوند.
- مشکل دیسک (Disk Failure): بعضی بلاک‌های ذخیره دیتا روی دیسک‌ها، شاید دیتای خود را به علت‌های مانند درست کارنکردن (Malfunction) خواندن یا نوشتن دیتا و یا تصادم‌هایی که وظیفه خواندن و نوشتن دیتا روی دیسک را دارد، از دست داده باشند. این مشکل امکان دارد در زمان خواندن و یا نوشتن عملیات ترانزکشن به روی دیسک واقع شده باشد.

- مشکلات فیزیکی و پایان ناگهانی ترانزکشن (Physical Problems and Catastrophes): این دسته می‌تواند یک فهرست طولانی مشکلات را در بر داشته باشد. این فهرست شامل مشکلات مختلفی مانند قطع جریان برق، حالات جوی نامناسب به خاطر کارکردن وسایل سیستم، آتش‌سوزی، دزدی، خراب‌کاری (Sabotage)، دوباره نوشتن اشتباهی روی دیسک‌ها و تیپ‌های دیتا و یا استفادهٔ دیسک نادرست از طرف استفاده‌کننده و غیره شده می‌تواند.
- چهار مشکل اول در فهرست بالا بیش‌تر عام بوده و دو مشکل آخری کم‌تر واقع می‌شود. در صورت واقع شدن یکی از چهار دلیل اول در قطع اجباری یک ترانزکشن، سیستم باید معلومات کافی به استفاده‌کننده ارائه کند تا به زودترین فرصت در قسمت حل مشکل اقدام صورت گیرد.

### ۳.۵ ست‌های عملیه‌های شامل ترانزکشن

طوری که قبلاً نیز در توضیح مثال ترانزکشن انتقال پول از حساب بانکی یک مشتری به حساب بانکی مشتری دیگر توضیح شد، ترانزکشن‌ها به اساس عملیه‌های خواندن (Read) و نوشتن (Write) از هم فرق می‌شوند.

- ست خواندن (Read Set): عبارت از دسته‌یی از عناصر دیتا است که در یک ترانزکشن خوانده می‌شوند و به اختصار آن را به حروف RS نشان می‌دهند.
  - ست نوشتن (Write Set): عبارت از دسته‌یی از عناصر دیتا است که در یک ترانزکشن نوشته می‌شوند و به اختصار آن را به حروف WS نشان می‌دهند.
  - ست عمومی (Base Set): عبارت از دسته‌یی از عناصر دیتا است که از اتحاد ست‌های خواندن و نوشتن به دست آمده و به اختصار آن را به حروف BS نشان می‌دهند.
- در مثال ریزرف کردن تکت پرواز اگر ست‌های خواندن، نوشتن و عمومی مشخص شوند، طور ذیل خواهند بود:

یعنی طرزالعمل‌های خواندن دیتا تنها دو فیلد را از جدول پرواز (FLIGHT) می‌خوانند. طرزالعمل‌های نوشتن بالای یک فیلد در جدول پرواز و تمام فیلدها در جدول حالت پرواز (FC) را در دیتا نوشته می‌نمایند؛ و ست عمومی تمام فیلدها را فهرست می‌کند.

$RS[Reservation] = \{FLIGHT.STSOLD, FLIGHT.CAP\}$

$WS[Reservation] = \{FLIGHT.STSOLD, FC.FNO, FC. DATE, FC. CNAME, FC. SPECIAL\}$

$BS[Reservation] = \{FLIGHT.STSOLD, FLIGHT.CAP, FC.FNO, FC. DATE, FC. CNAME, FC. SPECIAL\}$

یعنی طرزالعمل‌های خواندن دیتا، تنها دو فیلد را از جدول پرواز (FLIGHT) می‌خوانند. طرزالعمل‌های نوشتن بالای یک فیلد در جدول پرواز و تمام فیلدها در جدول حالت پرواز (FC) دیتا را نوشته می‌نمایند و ست عمومی (BS) تمام فیلدهای تأثیرشده را فهرست کرده است.

### ۳.۶ مراحل انزواء و ناهنجاری‌های ممکنه (Different Levels of Isolation and Possible Anomalies)

هر ترانزکشن دارای مشخصات معین و معلوم است. این مشخصات با طرزالعمل ست کردن ترانزکشن (SET TRANSACTION) در سیکویل تنظیم می‌شوند. نمونه‌های مشخصات ترانزکشن، روش دسترسی (Access Mode)، اندازه ساحت مشخصه (Diagnostic Area Size) و سطح انزواء (Isolation Level) را شامل می‌شوند. روش دسترسی به دیتا می‌تواند به شکل فقط خواندنی (Read Only) و یا هم به شکل خواندن-نوشتن (Read Write) تعیین شود. در شکل خودکار، روش دسترسی عبارت از حالت دوم یعنی خواندن - نوشتن است مگر این‌که از آدرس تنظیم سطح انزواء به شکل دیگری تنظیم شده باشد. روش و یا حالت خواندن-نوشتن اجازه می‌دهد تا در یک دیتابیس طرزالعمل‌های دیدن دیتا، داخل کردن دیتا، پاک کردن دیتا و ایجاد ست‌های دیتا به اجراء گذاشته شوند. حالت فقط خواندنی طوری که از نامش پیداست، تنها به خاطر دیدن دیتا تنظیم و استفاده می‌شود.

سطح انزواء (Isolation Level) هم به شکل محیطی (Local) تنظیم شده می‌تواند و هم به شکل کلی (Global) تعریف شده می‌تواند. در حالت اول، تنظیم سطح انزواء تنها در داخل یک ترانزکشن قابل اجراء بوده ولی در حالت دوم در تمام ترانزکشن‌ها به صورت عمومی قابلیت تطبیق را پیدا می‌نماید. شکل خودکار تنظیم سطح انزواء عبارت از محیط (Local) بوده و منحصر به یک جلسه است. در صورت تنظیم کردن عمومی از کلمه کلیدی GLOBAL در طرزالعمل تنظیم انزواء استفاده می‌شود تا بعد از اجراء این اختیار، سطح انزواء در تمام جلسات استفاده از دیتابیس مطابق اختیار داده شده تنظیم شود.

شکل عمومی طرزالعمل تنظیم سطح انزواء طور ذیل است:

```
SET [GLOBAL or SESSION] TRANSACTION ISOLATION LEVEL
{
    READ UNCOMMITTED
or READ COMMITTED
or REPEATABLE READ
or SERIALIZABLE
}
```

### ۳.۷ استفاده از ترانزکشن در سیکویل

ترانزکشن در سیکویل به مفهوم اصلی آن مانند سیستم‌های دیگر مدیریت دیتابیس استفاده می‌شود. به اساس توضیحات قبلی که داده شد، ترانزکشن عبارت از یک قسمت منطقی کار است که بسته‌بودن (Atomicity) آن ضمانت شده است. یک جمله سیکویل یا به عبارتی طرز‌العمل سیکویل همیشه یک کار را انجام می‌دهد؛ یعنی در هر حالت نتیجه اجرای یک طرز‌العمل همان کار است. یا به دیتابیس تغییر وارد می‌کند و یا هم در صورت عدم موفقیت طرز‌العمل، دیتابیس بدون تغییر در حالت اول خود می‌ماند.

در سیکویل، یک ترانزکشن در جریان یک جلسه استفاده و تنظیم شده می‌تواند. هر ترانزکشن آغاز شده در سیکویل بعد از اجرای طرز‌العمل‌های معین باید توسط طرز‌العملی پایان داده شود. طرز‌العمل آغاز ترانزکشن در سیکویل عبارت از (START TRANSACTION) و یا (BEGIN) بوده و طرز‌العمل‌های تکمیلی آن عبارت از (COMMIT) و یا (ROLLBACK) است. دو طرز‌العمل که برای آغاز ترانزکشن استفاده می‌شوند، معادل هم‌اند و در زمان استفاده، یکی از آن‌ها کفایت می‌کند. به عین شکل طرز‌العمل‌های پایان‌دهنده ترانزکشن نیز در صورت موافق‌بودن با نتیجه، تأیید و در صورت عدم موافق‌بودن استفاده‌کننده طرز‌العمل از رد تغییرات (ROLLBACK) استفاده به عمل می‌آید.

#### طرز‌العمل آغاز ترانزکشن، تأیید و یا رد (START TRANSACTION ... COMMIT ... ROLLBACK)

جهت اطمینان و کنترل کامل بالای دیتابیس و طرز‌العمل‌هایی که بالای دیتابیس اجراء می‌شوند، از طرز‌العمل آغاز ترانزکشن، تأیید و یا رد آن استفاده شده می‌تواند. طریقه کار طوری است که قبل از آغاز طرز‌العمل‌های که یک وظیفه معین را بالای دیتا در دیتابیس اجراء می‌کند، طرز‌العمل START TRANSACTION تایپ شود. به تعقیب این طرز‌العمل، استفاده‌کننده به صورت نورمال کار را با دیتابیس ادامه دهد. بعد از تکمیل کار، اگر کار اجراء شده مورد قبول استفاده‌کننده باشد کلمه تأیید یا COMMIT تایپ شده و اگر احیاناً کدام اشتباه رخ داده باشد و یا به اثر کدام علت دیگری طرز‌العمل‌های اجراء شده در این بلاک قابل قبول استفاده‌کننده نباشد، کلمه رد یا ROLLBACK تایپ شده و از تمام تغییراتی که بعد از طرز‌العمل آغاز ترانزکشن اجراء شده است، جلوگیری به عمل می‌آید. در مثال پایین، طرز‌العمل‌های تغییر دیتا در سیکویل از آدرس یک ترانزکشن تطبیق شده و نتیجه با استفاده از طرز‌العمل رد (حالت اول) قبول نشده و در جدول خالی نشان داده می‌شود.

۱. DELETE FROM PROJECT: (دیتای جدول پروژه پاک‌شده و جدول خالی می‌شود).

۲. SELECT \* FROM PROJECT: (محتوای جدول نشان داده می‌شود).

۳. START TRANSACTION: (ترانزکشن آغاز می‌شود) و

۴. INSERT INTO PROJECT VALUES ('1000', '03 Portfolio Analysis', 'Finance', 75):

(یک ریکورد دیتا به جدول پروژه اضافه می‌گردد).

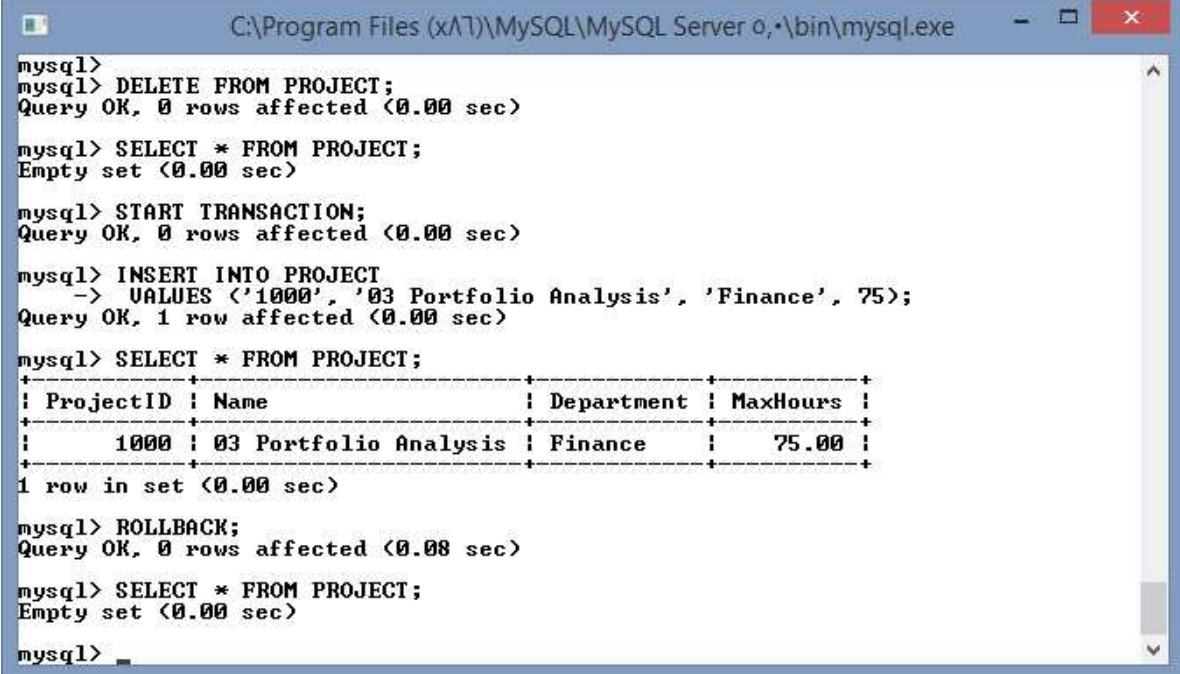
۵. SELECT \* FROM PROJECT: (محتوای جدول نشان داده می‌شود).

۶. ROLLBACK: (ترانزکشن با طرزالعمل رد پایان داده می‌شود).

or COMMIT (ترانزکشن با طرزالعمل تأیید پایان داده می‌شود).

۷. SELECT \* FROM PROJECT: (محتوای جدول نشان داده می‌شود).

شکل پایین، صفحهٔ سیکویل را با اجرای طرزالعمل‌های فوق با نتیجهٔ آن‌ها تحت کنترل ترانزکشن نشان می‌دهد.



```
mysql>
mysql> DELETE FROM PROJECT;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM PROJECT;
Empty set (0.00 sec)

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO PROJECT
-> VALUES ('1000', '03 Portfolio Analysis', 'Finance', 75);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM PROJECT;
+-----+-----+-----+-----+
| ProjectID | Name                | Department | MaxHours |
+-----+-----+-----+-----+
| 1000      | 03 Portfolio Analysis | Finance    | 75.00    |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> ROLLBACK;
Query OK, 0 rows affected (0.08 sec)

mysql> SELECT * FROM PROJECT;
Empty set (0.00 sec)

mysql>
```

شکل ۲-۳: تطبیق طرزالعمل‌های سیکویل و استفاده از ترانزکشن با طرزالعمل رد تغییرات در جدول پروژه

طوری که در صفحهٔ سیکویل شکل بالا دیده می‌شود، جدول پروژه در ابتداء خالی از دیتا گردیده و بعد از آن ترانزکشن آغاز گردیده است. در چوکات ترانزکشن یک ریکورد دیتا به جدول پروژه اضافه گردیده است. قبل از پایان ترانزکشن زمانی که کیوری انتخاب بالای جدول پروژهٔ تطبیق شده، ریکورد دیتا که به طور مؤقت در جدول ذخیره شده است را می‌دهد. بعد از پایان دادن ترانزکشن به کمک طرزالعمل رد (ROLLBACK) دیتای مؤقت به طور خودکار از جدول پروژه پاک می‌شود و زمانی که کیوری انتخاب دوباره به راه انداخته می‌شود، دیتابیس ست خالی را نشان می‌دهد.

عین مثال (کود) بالا با تغییر طرزالعمل رد به طرزالعمل تأیید (COMMIT) در سیکویل به راه انداخته شده، نتیجه کاملاً برعکس شده و طور شکل پایین تغییر می‌نماید.

```
C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin\mysql.exe
mysql>
mysql> DELETE FROM PROJECT;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM PROJECT;
Empty set (0.00 sec)

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO PROJECT
    -> VALUES ('1000', '03 Portfolio Analysis', 'Finance', 75);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM PROJECT;
+-----+-----+-----+-----+
| ProjectID | Name                | Department | MaxHours |
+-----+-----+-----+-----+
| 1000      | 03 Portfolio Analysis | Finance    | 75.00    |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM PROJECT;
+-----+-----+-----+-----+
| ProjectID | Name                | Department | MaxHours |
+-----+-----+-----+-----+
| 1000      | 03 Portfolio Analysis | Finance    | 75.00    |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

شکل ۳-۳ تطبیق طرزالعمل‌های سیکویل و استفاده از ترانزکشن با طرزالعمل تائید تغییرات در جدول پروژه

طوری که در مثال‌های بالا دیده می‌شود، تغییرات در یوزردیتا با استفاده از ترانزکشن کاملاً کنترل شده و در اختیار استفاده‌کننده است. هر زمانی که از ترانزکشن استفاده شده باشد، استفاده‌کننده می‌تواند تغییراتی از قبیل داخل کردن دیتا، پاک کردن دیتا، تغییر آوردن دیتا و به‌روز کردن دیتا را کاملاً در کنترل خود داشته باشد.





این فصل با توضیحات در مورد دیتا، دیتابیس و ساختمان‌های مربوطه آن در قسمت مقدمه آغاز گردید. مهم‌بودن دیتا در زندگی روزمره و مدیریت درست دیتا به حیث اصول مهم در ساحة دیتابیس‌ها بررسی شده و بحث روی کنترل کار روی تنظیم و استفاده از دیتا، متمرکز گردید. به خاطر کنترل دیتا در استفاده‌های هم‌زمان توسط استفاده‌کنندگان دیتابیس، به کارگیری ترانزکشن توسط سیستم‌های مدیریت دیتابیس‌های ارتباط به بحث گرفته شد. در مورد ترانزکشن، معلومات نسبتاً مختصر ارائه گردیده و خصوصیات ACID یک ترانزکشن و نیز حالت‌های یک ترانزکشن در عناوین مستقل توضیح شدند.

شریک‌ساختن دیتا و کنترل استفاده‌کنندگان در کار هم‌زمان بالای یک ست دیتا و به کارگیری ترانزکشن‌ها از جمله مسایل مورد بحث در این فصل بودند. پایان ترانزکشن و مسئله‌یی که هر ترانزکشن باید پایانی داشته باشد و پایان ترانزکشن می‌تواند مثبت ارزیابی شده و توسط استفاده‌کننده تأیید شود و یا هم برعکس پایان ترانزکشن با استفاده از طرزالعمل معین رد گردیده و دیتابیس به حالت اول خود حفظ شود نیز از موضوعات مهم بحث در این فصل به شمار می‌روند. نقیصه‌ها یا خرابی‌هایی که باعث مختل شدن کار ترانزکشن در دیتابیس می‌گردد و ست‌های عملیه‌های شامل ترانزکشن که همانا کار با یوزر دیتا را نشان می‌داد به بحث گرفته شدند. تنظیم مراحل انزواء و ناهنجاری‌های ممکنه در ترانزکشن‌ها و طریقه‌های کنترل و تنظیم سطوح مختلف انزواء (Isolation) با شکل عمومی کود سیکویل آن در یک عنوان مستقل گنجانیده شده است.

در درس استفاده از ترانزکشن در سیکویل طرزالعمل‌های عملی استفاده از ترانزکشن در یک جلسه سیکویل نشان داده شدند. طرزالعمل‌های عملی تنظیم و به کارگیری ترانزکشن‌ها با مثال‌های داخل کردن دیتا به یک جدول در یک دیتابیس معین به صورت تطبیقی نشان داده شدند. طرزالعمل‌های آغاز ترانزکشن و بعد از آن طرزالعمل‌هایی که توسط آن‌ها با یوزر دیتا کار صورت گرفت، نشان داده شده و پایان ترانزکشن در دو حالت تأیید (Commit) و رد (Rollback) با نتایج متوقعه در دو مثال عملی کار شده و تصویرهای صفحات کود سیکویل جهت مستندسازی موضوع، شامل متن کتاب گردید.





## فعالیت های فصل سوم

۱. برنامه سیکویل را در کوماندلاین اجرا کنید.
۲. با طرزالعمل `CREATE DATABASE ...` دیتابیس مکتب را به اساس معلومات در فصل دوم کتاب ایجاد کنید.
۳. با طرزالعمل `SHOW DATABASES` دیتابیس های موجود را ببینید.
۴. با طرزالعمل `USE DATABASE ...` دیتابیس مکتب را فعال کنید.
۵. با طرزالعمل های `CREATE TABLE ...` جدول های مربوط دیتابیس مکتب را به اساس مثال فصل دو ایجاد کنید.
۶. با طرزالعمل `START TRANSACTION` یک ترانزکشن را آغاز کنید.
۷. با طرزالعمل `INSERT INTO ...` دیتا را به جدول کارمند داخل سازید.
۸. با کیوری `SELECT * FROM EMPLOYEE;` محتوای جدول کارمند را ببینید.
۹. با طرزالعمل `ROLLBACK` ترانزکشن را رد کنید.
۱۰. دوباره با کیوری `SELECT * FROM EMPLOYEE;` محتوای جدول کارمند را ببینید.
۱۱. نتیجه کارتان را یادداشت کنید.
۱۲. مراحل بالایی (شماره ۵ الی ۷) را دوباره کار کنید.
۱۳. با طرزالعمل `COMMIT` ترانزکشن را تأیید کنید.
۱۴. دوباره با کیوری `SELECT * FROM EMPLOYEE;` محتوای جدول کارمند را ببینید.
۱۵. نتیجه کارتان را دوباره یادداشت کرده و تفاوت کار با ترانزکشن مثبت و منفی را ببینید.



## سوالات فصل سوم

۱. میتادیتا (Metadata) را با مثال توضیح دهید.
۲. یوزردیتا (Userdata) را با مثال توضیح دهید.
۳. هدف اصلی استفاده از ترانزکشن‌ها در سیستم‌های مدیریت دیتابیس چیست؟
۴. طرزالعمل‌های اصلی یک ترانزکشن کدام‌ها اند؟
۵. در صورت قبول نداشتن تغییرات دیتا که توسط یک ترانزکشن به دیتابیس وارد شده باشد، از کدام طرزالعمل پایانی ترانزکشن استفاده می‌شود؟
۶. چهار خصوصیت یک ترانزکشن کدام‌ها است؟ هدف هر کدام را به اختصار بیان کنید.
۷. پارامترهای ترانزکشن انتقال پول از حساب بانکی مشتری الف به حساب بانکی مشتری ب کدام‌ها اند؟ صرف نام ببرید.
۸. استفاده از ترانزکشن به کدام دو شکل صورت می‌گیرد؟
۹. حد اقل تعداد طرزالعمل‌های اجرایی در یک ترانزکشن چند است؟
۱۰. حد اکثر تعداد طرزالعمل‌های اجرایی در یک ترانزکشن چند است؟
۱۱. نقیصه‌ها یا خرابی‌های ممکنه در ترانزکشن‌ها را نام بگیرید.
۱۲. خواندن دیتا از دیتابیس در کدام سطح انزواء (Isolation) انجام شده می‌تواند؟ (Read Only) و یا (Read Write).
۱۳. آیا سطح انزواء (Isolation Level) می‌تواند به شکل کلی (Global) تنظیم شود؟ اگر جواب بلی است کود آن را بنویسید.
۱۴. فرق بین طرزالعمل‌های تکمیلی ترانزکشن COMMIT و ROLLBACK را مختصراً بیان کنید.
۱۵. تفاوت بین طرزالعمل‌های DELETE FROM با DROP TABLE در چه است؟ واضح سازید.

## فصل چهارم

### مسائل پیشرفته در سیستم‌های مدیریت دیتابیس‌های ارتباط



هدف کلی: محصلان با مفهوم خصوصیات پیشرفته دیتا (Advanced Features) آشنا شوند.

اهداف آموزشی: در پایان این فصل محصلان قادر خواهند شد تا:

۱. اندکس (Index) را تعریف نمایند.
۲. با انواع اندکس‌ها در دیتابیس کار نمایند.
۳. شیوه استفاده از اندکس‌ها در تنظیم دیتا را توضیح نمایند.
۴. 4 تهدیدهای دیتابیس را توضیح کنند.
۵. امنیت و کنترل دسترسی (Security and access control) را تشریح نمایند.

ذخیره دیتا در دیتابیس‌ها به اساس قواعد مدل ارتباط یکی از مزیت‌های بارز این نوع سیستم‌ها به شمار می‌رود. زمانی که دیتا به اساس قواعد استاندارد در دیتابیس‌ها ذخیره می‌شود، مشکلاتی در قسمت تنظیم درست دیتا، جلوگیری از پاک‌شدن دیتا و عدم تکرار ریکوردهای دیتا، رفع می‌گردند. ذخیره دیتا به صورت کامل و درست یک‌طرف موضوع است. دسترسی و استفاده درست و کامل از دیتا یا به عبارت ساده‌تر، تبدیل کردن ریکوردهای دیتا به معلومات طرف دومی، مهم‌تر از استفاده دیتابیس‌ها به شمار می‌آید. اگر یک دیتابیس بسیار خوب دیزاین شده و دیتای کافی در آن ذخیره شده باشد ولی از آن دیتا هیچ استفاده نشود، دیتابیس مؤثریت لازم را نمی‌داشته باشد. به خاطر مؤثرساختن یک سیستم، مسئله دوم یعنی استفاده از دیتا حائز اهمیت است. به این منظور بحث بر استفاده کردن درست، کامل و به جا از دیتابیس‌ها ضرورت به تعمق بیش‌تر در موضوع را می‌نماید. طریقه‌هایی به خاطر دسترسی مناسب به دیتا و پیدا کردن معلومات لازم از یک دیتابیس با سرعت و دقت کافی از اصول اساسی موضوع فوق به شمار می‌رود. در دیتابیس‌ها قابلیت ایجاد و اجرای ساختارهایی به نام اندکس (Index) موجود است. اندکس‌ها به معنی اصلی خودشان بوده طوری که از طریق اندکس‌ها در یک کتاب بزرگ، صفحاتی که یک موضوع مشخص در آن‌ها ذکر گردیده است در یک‌زمان بسیار کوتاه و به آسانی پیدا می‌شود. دیتابیس‌ها و اندکس‌ها، استفاده‌کنندگان را قادر می‌سازد تا معلومات مورد ضرورت‌شان را با سرعت و دقت بالا پیدا کنند. در این فصل توضیحات لازم و طرزالعمل‌های استفاده عملی از اندکس‌ها در سیکویل توضیح داده شده است. بحث دیگری که به تفصیل به آن پرداخته شده، دسترسی استفاده‌کنندگان به سیستم امنیت و کنترل آن است. به خاطر استفاده مؤثر از دیتابیس، گروه‌های استفاده‌کنندگان با صلاحیت‌های مختلف تنظیم شده می‌تواند. این مسئله تحت عنوانی مجزا به صورت مفصل توضیح گردیده است. کنترل دسترسی (Access Control) با توضیح امکانات برای استفاده‌کنندگان از طرازهای مختلف به بحث گرفته شده است. در ادامه، تهدیدهایی که متوجه یک سیستم مدیریت دیتابیس بوده می‌توانند و از طریق آن‌ها یک دیتابیس مستقیماً متضرر شده می‌تواند با مثال‌های مناسب توضیح شده است. ایجاد حساب‌های استفاده‌کنندگان و حذف حساب‌ها به خاطر کنترل بهتر سیستم با مثال‌های عملی در سیکویل نشان داده شده است.

#### ۴.۱ اندکس‌ها (Indexes)

در دیتابیس‌ها یک اصل است که دیتا در جدول‌های دیتابیس‌ها به شکل ریکوردها ذخیره می‌شود. یکی از فیلدهای جدول و یا هم‌گروپی از فیلدهای یک جدول به خاطر شناسایی ریکوردها استفاده می‌شود. در یک جدول کوچک که معلومات در مورد اعضای یک خانواده در آن ثبت باشد، فیلدی که معلومات نام اعضای خانواده را دارد به خاطر شناسایی ریکوردها کفایت خواهد کرد. جدولی دیگری که در آن حاضری یک صنف 50 نفری ثبت باشد معمولاً نام شاگردان به خاطر شناسایی کفایت نمی‌کند و امکان موجودیت چندین نفر به یک نام است. به خاطر حل مشکل نام و نام پدر شاگردان به صورت یک‌جا برای شناسایی شاگردان به کار می‌رود. طور مثال فهرست نفوس یک ولسوالی، در این حالت فیلدهای نام و نام پدر اشخاص نیز به خاطر شناسایی کارآمد نخواهد بود. جهت شناسایی ریکورد اشخاص در هم‌چو موارد، نمبر تذکره تابعیت شاید یکی

از راه حل‌های مناسب به خاطر شناسایی اشخاص ساکن در یک ولسوالی باشد. همین فیلد یا فیلدهای که برای شناسایی ریکوردها در یک جدول استفاده می‌شوند به نام فیلد یا فیلدهای کلیدی یاد می‌شوند.

استفاده از اندکس‌ها عبارت از طریقه‌یی در ساختمان دیتا اند که توسط آن‌ها ریکوردهای فایل‌های دیتابیس به اساس مشخصه‌های ضروری، استفاده شده می‌تواند. به کارگیری اندکس‌ها در دیتابیس شبیه استفاده از اندکس‌ها (نشانه‌ها) در کتاب‌ها است. در یک کتاب موضوعاتی که استفاده شده‌اند در اخیر کتاب با ذکر صفحات موضوع به ترتیب الفبا نشان داده می‌شوند. طور مثال اصطلاحات XML و UML در یک کتاب مربوط به سیستم‌های معلوماتی در لیست اندکس‌ها در اخیر کتاب دیده می‌شوند. به اساس ترتیب حروف الفبا UML مقدم‌تر بر XML نشان داده شده و شماره صفحات این دو اصطلاح در پهلوی نام‌شان ذکر می‌گردد. استفاده‌کننده کتاب به آسانی می‌تواند به صفحات نشان داده شده، مراجعه و معلومات مورد ضرورت خود را در کتاب متذکره، مطالعه نماید. در دیتابیس‌ها، اندکس‌ها به اساس مشخصه‌های جدول‌ها ترتیب شده و به کار گرفته می‌شوند. اندکس‌ها انواع مختلف دارند، دسته‌یی از اندکس‌ها که بالای دیتای تنظیم‌شده (Ordered Data) کار می‌نمایند و اندکس‌هایی که بالای دیتای غیر تنظیم‌شده (Unordered Data) کار می‌نمایند به حیث دو دسته عمومی تعریف می‌شوند. دیتای تنظیم‌شده عبارت از همان دیتایی است که در آن کلید اولیه موجود بوده و توسط محتواهای کلید اولیه هر ریکورد موجود در جدول، شناسایی می‌شود. در حالت دیگر یعنی دیتای غیر تنظیم‌شده، کلید اولیه جدول مشخص نبوده و دیتا به شکل مجموعه‌های عمومی ذخیره است. اندکس‌هایی که با دیتای تنظیم‌شده کار می‌نمایند به نام‌های اندکس اولیه (Primary Index) و اندکس دومی (Secondary Index) یاد شده و اندکس‌هایی که با دیتای غیر تنظیم‌شده، کار می‌نمایند به نام اندکس کلستری (Clustered Index) یاد می‌شوند. البته انواع اندکس‌های اولیه، دومی و کلستری به صورت عمومی به نام اندکس‌های یک سطحی (One Level Indexes) یاد می‌شوند. عناوین پیشرفته در موضوع، برمی‌گردد به اندکس‌های چند سطحی (Multi-Level Indexes) که پیچیدگی بیش‌تری نظر به اندکس‌های یک سطحی دارند. اندکس‌های پیچیده یا چند سطحی، خارج از موضوع بحث در این کتاب بوده و در این جا از توضیح آن‌ها صرف نظر گردیده است. توضیحات مختصر در مورد اندکس‌های یک سطحی یا ساده با ذکر مثال‌های مناسب در ذیل ارائه می‌گردد.

### ۴.۱.۱ اندکس اولیه (Primary Index)

اندکس اولیه در یک فایل دیتا به شکل تنظیم‌شده، استفاده می‌شود یعنی دیتا در فایلی باید به اساس یک فیلد کلیدی تنظیم شده باشد. فیلد کلیدی معمولاً عبارت از فیلد کلید اولیه (Primary Key) در جدول دیتابیس است. کلید اولیه در یک جدول با در نظر داشت مدل ارتباط، دیتای تکراری را اجازه نداده و محتوای آن، ریکوردهای دیتای جدول را شناسایی می‌نماید؛ یعنی وقتی نمبر تذکره یک شخص از جدول معلوماتی‌اش گرفته می‌شود، نام و آدرس شخص معلوم شود.

اندکس اولیه در اصل، عبارت از یک جدول با دو فیلد با طول‌های ثابت دیتا در هر فیلد است؛ یعنی زمانی که اندکس اولیه بالای یک جدول تنظیم می‌شود، فیلد کلید اولیه جدول و فیلد جدیداً ایجاد شده، توسط اندکس

به یک ساختمان جدید دیتا در سیستم ذخیره می‌شود. در ساختمان جدول اندکس هر محتوا در فیلد جدید به آدرس محتوای فیلد کلید اولیه در جدول اصلی اشاره می‌کند. فایل اندکس در مقایسه با جدولی که برای آن ایجاد شده است، به مراتب ساحة کم‌تر را روی دیسک احتواء می‌کند. فرض شود یک جدول با 20 فیلد (که یکی آن کلید اولیه است) و 2000 ریکورد دیتا در دیتابیس موجود است. در حالت عادی (بدون اندکس) زمانی که کیوری بالای آن تطبیق شود، حد اقل 20 ضربدر 2000 یعنی حدود 40000 عملیه باید اجراء شود تا نتیجه کیوری نشان داده شود. زمانی که اندکس اولیه بالای این جدول وضع شود، تعداد عملیه‌ها به 2 ضربدر 2000 یعنی حدود 4000 عملیه تقلیل می‌یابد. اگر دقت شود، با استفاده از اندکس، تفاوت تعداد عملیه‌های اجراء شده به اساس یک کیوری در یک جدول ساده، ده برابر کمتر می‌شود که این یک تفاوت بالا بوده و امتیاز آن به استفاده از اندکس‌ها می‌رسد.

### ۴.۱.۲ اندکس دومی (Secondary Index)

اندکس دومی معمولاً از یک کلید کاندیدی (Candidate Key) در یک جدول تشکیل می‌شود و یا هم اندکس دومی به اساس یک فیلد غیرکلیدی (NonKey) با دیتای تکراری تشکیل می‌شود. محتوای کلید کاندیدی، غیر تکراری بوده و بعد از کلید اولیه، ریکوردهای جدول را شناسایی می‌نماید. اندکس دومی در کنار این که به اساس دیتای غیر تکراری ایجاد می‌شود، می‌تواند به اساس دیتای تکراری نیز ایجاد شود. زمانی که یک اندکس دومی به اساس فیلد کلید کاندیدی ایجاد شود، طریقه کار مشابه به اندکس اولیه بوده، صرف فیلد یا فیلدهایی که دیتای‌شان مورد پروسس قرار می‌گیرد از فیلد کلید اولیه به فیلد کلید کاندیدی تغییر می‌یابد. برای یک جدول در دیتابیس چندین اندکس دومی ساخته شده می‌تواند. در صورتی که اندکس دومی به اساس فیلدی ایجاد شود که در آن امکان تکرار دیتا موجود باشد، یعنی فیلد غیر کلیدی استفاده شود، در چنین حالتی محتوای فیلدهای اندکس شده در یک‌زمان به بیش‌تر از یک محتوا در جدول اصلی اشاره کرده و آن را مورد پروسس قرار می‌دهد. اندکس دومی به خاطر طول بیش‌تر محتوای فیلدها، معمولاً جای بیش‌تر را در حافظه کمپیوتر و زمان طولانی‌تر نظر به اندکس اولیه را اخذ می‌کند. ولی به هر صورت، اندکس دومی خوبی‌های خود را در مواردی دارد. از جمله می‌توان از تنظیم اضافه یک اندکس را برای عین جدول و ست دیتا نام برد.

### ۴.۱.۳ اندکس کستری (Clustering Index)

اندکس کستری به اساس یک دیتا فایل تنظیم شده، ایجاد می‌شود طوری که محتوای دیتا فایل توسط یک فیلد غیرکلیدی تنظیم شده باشد. فیلد غیرکلیدی می‌تواند محتوای تکراری داشته باشد و به خاطر شناسایی هر ریکورد استفاده نشده بلکه توسط یک فیلد غیر کلیدی، امکان شناسایی گروهی از ریکوردها متصور بوده می‌تواند. همین فیلد استفاده شده در اندکس کستری، به نام فیلد کستری و دیتا فایل مربوطه به نام فایل کسترتشده، یاد می‌شود. اندکس کستری به خاطر سرعت بخشیدن به پروسه پیدا کردن ریکوردهای دیتا در یک جدول به کار برده می‌شود که در آن ریکوردها، معلومات مشترک در فیلد کستری داشته باشند. همین کار با گروهی از محتواها، اندکس کستری را از اندکس اولیه که قبلاً توضیح شد مجزا می‌سازد.

طریقه کار با آن طوری است که در اندکس کلستری یک جدول تنظیم شده را با دو فیلد ایجاد می نماید طوری که یک فیلد از جدول اصلی گرفته شده و فیلد دوم اشاره به اندکس برای دسته یی از محتواهای فیلد کلستر شده جدول اصلی را در برمی داشته باشد. طور مثال یک جدول، دیتا در مورد شاگردان یک مکتب را دارد. از طریق اندکسی که در این جدول به کار رفته است، ولایات مربوطه شاگردان در نظر گرفته می شود. با تصور این که از 10 ولایت مختلف معلومات 500 نفر شاگرد در جدول متذکره شامل نام، نام پدر، صنف، شماره تلفون، ولایت و غیره ذخیره است. برای این جدول، اندکسی تنظیم شده است که به اساس آن گروپ های شاگردان را نظر به ولایت های مربوطه شان تهیه نماید؛ تا با به راه انداختن آن شاگردان دسته بندی شده و نتیجه، طوری که لازم است، ارائه گردد.

## ۴.۲ استفاده از اندکس ها در تنظیم دیتا

در دیتابیس های ارتباطی، اندکس ها به خاطر زود پیدا کردن دیتا در زمان اجراء کیوری ها بالای دیتا استفاده می شوند. ایجاد و از بین بردن اندکس ها بالای کود اصلی ساختمان دیتابیس تأثیری ندارد؛ یعنی کود مربوط به ساختمان های اندکس به صورت جداگانه در دیتابیس واقع گردیده و استفاده می شود. اندکس ها در عقب پرده به خاطر استفاده بهتر از دیتابیس ها تنظیم و به کار گرفته می شوند. ایجاد و به کارگیری از اندکس های مناسب، باعث بهتر شدن اجراء دیتابیس ها می شود. طوری که قبلاً نیز مثالی در زمینه ارائه شد، در سیکویل، اندکس ها در حالت عادی مشابه به استفاده از اندکس ها در خواندن یک کتاب است. زمانی که یک کتاب به خاطر پیدا کردن یک موضوع ورق زده می شود، مطالعه از صفحه اول کتاب آغاز شده و به ترتیب تا پیدا کردن موضوع پیش می رود. هر زمانی که موضوع پیدا شد، این بستگی به مطالعه کننده دارد تا متباقی صفحات کتاب را نیز مطالعه می کند و یا از همان صفحه یی که موضوع را پیدا نموده است، یادداشت مورد نظر خود را گرفته و از مطالعه قسمت باقی مانده کتاب صرف نظر می کند. این کار شاید وقت گیر و طاقت فرسا باشد. راه آسان تر در مورد بالا، دیدن صفحات آخری کتاب (فهرست اندکس ها) و پیدا کردن صفحاتی که موضوع یاد شده در آن ذکر گردیده است. حالت دوم پیدا کردن موضوع در یک کتاب که با استفاده از اندکس ها صورت می گیرد به مراتب سهل تر، بهتر و با دقت بیش تر و وقت کم تر نظر به طریقه اول اجراء می گردد.

اگر در یک دیتابیس از اندکس ها استفاده نشود، در زمان کار با دیتا تمام ریکوردهای ثبت شده در دیتابیس باید دیده شده و نتایج حاصله آن پیدا شود. یک دیتابیس شاید به تعداد صدها هزار و یا یک میلیون ها ریکورد دیتا داشته باشد. طور مثال دیتابیس ساده نفوس افغانستان با حد اقل سی میلیون ریکورد دیتا اگر در نظر گرفته شود، به خاطر پیدا کردن یک شخص از یک منطقه معین، اول باید میلیون ها عملیه باید اجراء شود. در صورت استفاده از اندکس ها در همین مثال اگر ریکورد شخص از یک منطقه معین مثلاً خوشحال خان مینه خواسته شود و در دیتابیس، اندکس هایی برای آدرس به ترتیب ولایت، ولسوالی، ناحیه و منطقه از قبل موجود باشند و به کار گرفته شوند، سرعت عمل به مراتب زیاد شده و سیستم با استفاده از تعداد محدودتری از عملیه ها، دیتای مورد نظر را کیوری می نماید. در صورت عدم استفاده از اندکس ها در یک دیتابیس آن را به نام بررسی جدول (Table Scan) یاد می کنند. البته بررسی کردن جدول یا Table Scan را همیشه هم نمی توان به شکل یک مشکل مطرح کرد، ولی زمانی که تعداد ریکوردها در یک جدول به هزارها و میلیون ها می رسد،

مشکلات ناشی از تعدد ریکوردهای دیتا باعث ضیاع وقت و کم‌کاری سیستم شده می‌تواند. جهت وضاحت بیش‌تر، جدول ASSIGNMENT از مثال‌های قبلی مربوط دیتابیس مکتب در شکل پایین به حیث یک مثال ساده در نظر گرفته شود.

ASSIGNMENT		
ProjectID	EmployeeNum	HoursWorked
1000	100	17.5
1000	300	12.5
1000	400	8
1000	500	20.25
1200	100	45.75
1200	400	70.5
1200	600	40.5
1400	200	75
1400	700	20.25
1400	500	25.25

شکل ۴-۱ جدول مأموریت با دیتای نمونه در آن

اگر کیوری پایین به شکل نمونه، بالای جدول مأموریت اجراء شود و از دیتابیس، معلومات پروژه‌ها و کارمندان خواسته شوند که ساعات کاری‌شان در یک دامنه معین بین 17 و 21 باشد.

```
SELECT ProjectID, EmployeeNum, HoursWorked
```

```
FROM ASSIGNMENT WHERE (HoursWorked > 17) AND (HoursWorked < 21)
```

در حالتی که اندکس در دیتابیس مکتب برای جدول مأموریت وضع نشده باشد تا کیوری بالا را کمک کند، سیستم مدیریت دیتابیس یک اسکن را انجام داده، تمام ریکوردهای جدول مأموریت را به ترتیب بررسی می‌کند و ساعات کاری را پیدا می‌نماید که در بین 17 و 21 باشد. در شکل پایین نحوه کار سیستم نشان داده شده است.

Searches	ProjectID	EmployeeNum	HoursWorked		
Search row 1	1000	100	17.5	Matches	Row 1
Search row 2	1000	300	12.5		
Search row 3	1000	400	8		
Search row 4	1000	500	20.25	Matches	Row 4
Search row 5	1200	100	45.75		
Search row 6	1200	400	70.5		
Search row 7	1200	600	40.5		
Search row 8	1400	200	75		
Search row 9	1400	700	20.25	Matches	Row 9
Search row 10	1400	500	25.25		



شکل ۴-۲ تطبیق کیوری با پارامترهای مشخص بالای جدول مأموریت

در طرزالعمل کیوری بالا اگر دیده شود، مراحل اجرایی آن تمام ریکوردهای جدول را یک به یک بررسی (اسکن) نموده و نتیجه مورد ضرورت را ارائه داشته است. اگر به عوض جدول مأموریت دیتابیس مکتب که در مثال بالا استفاده شد، از اندکس کار گرفته شود و فیلد ساعات کاری (HoursWorked) اندکس شود چه واقع خواهد شد؟ ...

در این صورت هر واحد اندکس، محتوای ستون ساعات کاری جدول مأموریت را برای ریکوردهای مربوطه در بر گرفته و یک ارتباط (ریفرنس) به ریکورد اصلی در جدول مأموریت را ایجاد خواهد کرد. سیکویل، قیمت‌های اندکس را به شکل صعودی ترتیب می‌نماید. اندکس زمینه پیدا کردن قیمت‌های خواسته‌شده را برای سیستم مدیریت دیتابیس مهیا ساخته و به اساس مثال بالا، قیمت‌های بین 17 الی 21 به سادگی شناسایی می‌شوند.

بعد از شناسایی اندکس‌های مذکور، سیستم مدیریت دیتابیس تمام جدول را اسکن ننموده بلکه به اساس اندکس‌های پیداشده، صرف ریکوردهای دیتا را نشان خواهد داد که در جریان عملیۀ اسکن پیدا شده است. در این مثال به عوض پروسس و اسکن کردن 10 ریکورد موجود در جدول مأموریت، تنها 3 ریکورد مورد پروسس قرار گرفته و نشان داده می‌شوند. حال اگر تعداد ریکوردها به هزاران برسد، تفاوت زمان و تعداد عملیه‌های که باید اجراء شود به چه اندازه تغییر خواهد نمود؟

### ۴.۳ استفاده از طرزالعمل اندکس در سیکویل

- طرزالعمل ایجاد اندکس‌ها در سیکویل: جهت ایجاد اندکس در سیکویل از طرزالعمل CREATE INDEX ... استفاده به عمل می‌آید. شکل عمومی طرزالعمل ایجاد اندکس بالای یک جدول دیتابیس در سیکویل طور ذیل است:

CREATE INDEX IndexName ON TableName (ColumnName)

در طرزالعمل بالا از اختیاراتی بیش‌تری که در طرزالعمل ایجاد اندکس موجود بوده، صرف نظر گردیده است و تنها قسمت‌های کلیدی طرزالعمل نشان داده شده‌اند. قسمت اول طرزالعمل، طبعاً اندکس را ایجاد می‌کند. کلمۀ IndexName حتمی بوده در آن نام اندکس توسط استفاده‌کننده تعریف می‌شود. بعدتر کلمۀ ON نیز حتمی بوده و نام جدولی که اندکس می‌شود به تعقیب آن همراه با فیلد مورد نظر جدول در داخل قوس باید ذکر گردد.

به خاطر تطبیق و ایجاد اندکس مثال قبلی از طرزالعمل سیکویل به شکل ذیل استفاده می‌شود:

>> CREATE INDEX IDX\_HoursWorked ON ASSIGNMENT (HoursWorked);

در طرزالعمل ایجاد اندکس برای جدول مأموریت نام اندکس IDX\_HoursWorked تعیین شده، ASSIGNMENT نام جدولی که اندکس برای آن تعریف شده و HoursWorked هم عبارت از فیلد اندکس‌شده جدول است.

نحوه کار اندکس‌ها طوری است که دیتابیس فیلد و یا فیلدهای تعریف‌شده، حیثیت اندکس را گرفته و محتواهای آن‌ها را به یک ساختمان دیتا به نام B-tree ترتیب می‌نماید. یک ساختمان B-tree اجازه یافتن دیتا را با سرعت زیاد و با استفاده کم‌ترین ساحه، روی دیسک انجام می‌دهد. این ساختمان به سیستم مدیریت دیتابیس اجازه می‌دهد تا با سرعت دیتا را یافته و به سادگی در وسط پروسه جست‌وجو (Search)، نقاط توقف و نقاط آغاز دوباره عمل جست‌وجو را به اساس کیوری‌های داده شده تنظیم نماید.

- طرز‌العمل دیدن اندکس‌های یک جدول در سیکویل: اندکس‌های تعریف‌شده برای جدول‌ها، قابل دیدن و چک کردن است. در سیکویل، اندکس‌های یک جدول توسط طرز‌العمل ذیل دیده شده می‌تواند:

SHOW INDEX FROM TableName

طرز‌العمل دیدن اندکس‌ها با همین سادگی آن دارای اهمیتی زیادی است. قبل از ایجاد اندکس در جدولی، استفاده‌کننده می‌تواند ببیند که آیا از قبل کدام اندکسی در جدول بوده و یا خیر. به خاطر دیدن اندکس‌های جدول مثال قبلی طور ذیل تایپ شود:

>> SHOW INDEX FROM ASSIGNMENT;

- طرز‌العمل پاک کردن اندکس از یک جدول در سیکویل: همان طوری که استفاده‌کننده یک اندکس را به جدولی در دیتابیس اضافه کرده می‌تواند به همان شکل یک استفاده‌کننده صلاحیت پاک کردن یک اندکس از قبل ایجاد شده را نیز دارا می‌باشد. به خاطر پاک کردن یک اندکس از طرز‌العمل DROP INDEX ... استفاده می‌شود. شکل عمومی طرز‌العمل پاک کردن اندکس عبارت است از:

DROP INDEX IndexName ON TableName

در طرز‌العمل بالا، کلمات ثابت به حروف بزرگ نوشته شده و کلماتی که نظر به جدول‌ها و اندکس‌های مربوطه‌شان متفاوت است با حروف کوچک و بزرگ نوشته شده است. به خاطر پاک کردن اندکس از جدول مأموریت طور ذیل تایپ شود:

>> DROP INDEX IDX\_HoursWorked ON ASSIGNMETN;

نتیجه اجرای طرز‌العمل‌های توضیح شده در شکل پایین دیده می‌شود.

```

C:\Program Files (x86)\MySQL\MySQL Server 5.5\bin\mysql.exe
mysql> DROP TABLE ASSIGNMENT;
Query OK, 0 rows affected (0.08 sec)

mysql> CREATE TABLE ASSIGNMENT(ProjectID int, EmployeeNum int, HoursWorked Decimal(5,2));
Query OK, 0 rows affected (0.09 sec)

mysql> DESCRIBE ASSIGNMENT;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ProjectID      | int(11)       | YES  |     | NULL    |       |
| EmployeeNum    | int(11)       | YES  |     | NULL    |       |
| HoursWorked    | decimal(5,2)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.03 sec)

mysql> CREATE INDEX IDX_HoursWorked ON ASSIGNMENT(HoursWorked);
Query OK, 0 rows affected (0.14 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> DESCRIBE ASSIGNMENT;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ProjectID      | int(11)       | YES  |     | NULL    |       |
| EmployeeNum    | int(11)       | YES  |     | NULL    |       |
| HoursWorked    | decimal(5,2)  | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)

mysql> SHOW INDEX FROM ASSIGNMENT;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Table          | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Index_type | Index_comment |
+-----+-----+-----+-----+-----+-----+-----+-----+
| assignment     | 0          | 1        | 1            | HoursWorked | A         | BTREE      |                |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> DROP INDEX IDX_HoursWorked ON ASSIGNMENT;
Query OK, 0 rows affected (0.16 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> SHOW INDEX FROM ASSIGNMENT;
Empty set (0.00 sec)

mysql>

```

شکل ۳-۴ نتیجه اجرای طرزالعمل‌های مربوط به اندکس در سیکویل

آن گونه که در شکل بالا دیده می‌شود در ابتداء با استفاده از طرزالعمل DROP TABLE ... جدول مأموریت از دیتابیس، پاک شده است. به تعقیب آن جدول مأموریت با استفاده از طرزالعمل CREATE TABLE ... دوباره ایجاد شده و توسط طرزالعمل DESCRIBE ... ساختمان آن دیده شده است. طوری که در شکل دیده می‌شود، در این مرحله تمام فیلدهای جدول مأموریت، فیلدهای عادی می‌باشد. طرزالعمل بعدی اندکسی را بالای جدول مأموریت ایجاد نموده و فیلد ساعات کاری به این منظور به کار رفته است. دوباره زمانی که ساختمان جدول دیده شده است، این تفاوت در ساختمان جدول با ذکر اصطلاح MUL در ستون Key هویدا است. به تعقیب کار طرزالعمل نشان‌کرده اندکس جدول مأموریت با استفاده از طرزالعمل SHOW INDEX ... به راه انداخته شده و نتیجه نیز یک ریکورد را نشان می‌دهد که عبارت از همان اندکس تعیین‌شده در

جدول است. در ادامه، طرزالعمل ... DROP INDEX به خاطر پاک کردن اندکس از جدول مأموریت مؤفقانه اجراء شده است. نتیجه نهایی با استفاده دوباره از طرزالعمل نشان کرده اندکس نتیجه ست خالی را نشان می دهد؛ یعنی آخرین طرزالعمل اجرایی یعنی پاک کردن اندکس جدول مأموریت مؤفقانه تطبیق شده است.

#### ۴.۴ امنیت و کنترل دسترسی به سیستم (Security and Access Control)

دیتا در بیش تر موارد دارای اهمیت زیاد است. در بعضی حالات دیتا اهمیت حیاتی دارد. در مواردی که دیتا اهمیت اقتصادی، سیاسی، شخصی و غیره داشته، افشاشدن و دسترسی غیرقانونی به دیتا منتج به تهدیدهای جدی در موارد یاد شده خواهد شد. به هر اندازه‌یی که دیتا اهمیت داشته باشد به همان اندازه نگهداری و استفاده به‌جا از آن نیز حائز اهمیت بوده اشخاص و سیستم‌های ذخیره و مدیریت دیتا باید به صورت مطمئن امور و عملیات مربوط به ذخیره، تنظیم و استفاده از دیتا را اجراء نمایند. امنیت دیتا و دسترسی به دیتا توسط سیستم‌های مدیریت دیتابیس یکی از بخش‌های مهم در نظر گرفته شده و استفاده می‌شود.

یک دیتابیس می‌تواند چندین استفاده‌کننده داشته باشد. استفاده‌کنندگان بیش تر یک سیستم واحد از نقاط قوت سیستم‌های مدیریت دیتابیس به شمار می‌آید. در یک سیستم دیتابیس چندین استفاده‌کننده سیستم مدیریت دیتابیس باید تخنیک‌هایی را ارائه دهد تا از طریق آن‌ها استفاده‌کنندگان اساسی و گروپ‌های استفاده‌کنندگان با صلاحیت به قسمت‌های انتخاب‌شده دیتابیس، دسترسی داشته باشند؛ و همین دسته از استفاده‌کنندگان به بخش‌هایی که برای آن‌ها ضروری نیست، دسترسی نداشته باشند. این موضوعات زمانی حائز اهمیت بیش تر می‌شود که یک سیستم مرکزی دیتابیس در یک اداره یا ارگان استفاده شود و در آن اداره، دسته‌های مختلف استفاده‌کنندگان از آن سیستم کار می‌گیرند. طور مثال معلومات معاش کارمندان یک اداره یا راپورهای فعالیت‌های کارمندان یک اداره که در دیتابیس عمومی ثبت است، باید از دسترسی عمومی استفاده‌کنندگان در امن باشد. تنها مسئولین مالی و اداری بالا باید قادر به دیدن آن باشند.

به عبارت دیگر، یک سیستم دیتابیس که مورد استفاده چندین فرد است، همه استفاده‌کنندگان آن ضرورت به دسترسی به تمام بخش‌های دیتابیس را نمی‌داشته باشند. به طور مثال، در دیتابیس یک بانک، کارمندان تأدیات، باید تنها قادر به دیدن بخشی باشند که در آن ریکورد پولی قابل پرداخت به کارمندان نشان داده می‌شود؛ یعنی بخش تأدیات نباید دسترسی به معلومات بیش تر در مورد کارمندان داشته باشند. به عین شکل به اساس پالیسی یک اداره، استفاده‌کنندگان بخش تدارکات باید صلاحیت‌های وظیفوی خود را داشته باشند و به معلومات محرمانه در قسمت تصمیم‌گیری عقد قرارداد و غیره دسترسی نداشته باشند. این امر باعث جلوگیری از فساد و سوءاستفاده در نهاد مذکور خواهد گردید.

به منظور تسهیل در امور فوق یک سیستم مدیریت دیتابیس معمولاً دارای یک بخش فرعی به نام سیستم فرعی امنیت دیتابیس و تنظیم صلاحیت‌ها است. توسط همین سیستم فرعی امنیت استفاده و دسترسی به بخش‌های مختلف دیتابیس‌ها کنترل و تنظیم گردیده از دسترسی استفاده‌کنندگان غیر مسئول جلوگیری صورت می‌گیرد. امنیت دیتابیس روی استفاده از دیتا به دو گونه تنظیم می‌شود:

- میکانیزم‌های اختیاری امنیتی (Discretionary Access Control): این نوع میکانیزم‌ها به خاطر تنظیم و دسترسی استفاده‌کنندگان به دیتا به کار برده می‌شود. میکانیزم‌های اختیاری امتیازهای به کارگیری دیتا، توسط استفاده‌کنندگان را تصدیق می‌کند. امتیازهای به کارگیری دیتا توسط استفاده‌کنندگان عبارت از دسترسی به دیتا فایل‌های مشخص، ریکوردها و یا فیلدهای دیتا در یک حالت مشخص است. حالت‌های مشخص عبارت از خواندن، داخل کردن، پاک کردن و به‌روز کردن دیتا در قسمت‌های یاد شده است.
  - میکانیزم‌های اجباری امنیتی (Mandatory Access Control): میکانیزم‌های اجباری جهت کنترل امنیت دیتابیس‌ها به خاطر اجرای امنیت در چندین سطح مختلف (Level) به کار برده می‌شوند. در این نوع میکانیزم‌ها دیتا و استفاده‌کنندگان هر کدام به دسته‌ها یا سطوح مختلف تقسیم می‌شوند. بعد از آن پالیسی‌های مناسب برای امنیت بخش‌ها و دیتا، بالای استفاده‌کنندگان و دیتا در یک اداره اجراء می‌شود. به طور مثال در یک اداره، پالیسی برای استفاده‌کنندگان دیتابیس وضع شده است که در آن یک گروه از استفاده‌کنندگان، تنها قادرند تا دیتای مربوط گروه خود و دیتای گروه‌هایی که از نظر صلاحیت پایین‌تر از گروه خودشان قرار دارد را ببینند. هیچ گروه دیتای گروه‌های بالاتر از خود را دیده نمی‌توانند.
- به طور مثال استفاده‌کنندگان به اساس میکانیزم‌های اجباری به گروه‌های الف، ب و ج به ترتیب تنظیم شده است. استفاده‌کنندگان گروه ب می‌توانند دیتای خود و دیتای گروه ج را ببینند ولی دیتای گروه الف را دیده نتوانند. در حالی که استفاده‌کنندگان گروه الف دیتای هر سه گروه را می‌بینند و استفاده‌کنندگان گروه آخر یعنی گروه ج، تنها می‌توانند به دیتای خود را دسترسی داشته باشند.
- یک نوع فرعی دیگری از میکانیزم‌های اجباری به نام «امنیت به اساس وظیفه» یاد می‌شود. در این نوع تنظیمات امنیتی صلاحیت‌های استفاده‌کنندگان به اساس پالیسی‌ها و امتیازات وظیفوی اداره مربوطه‌شان تعیین می‌شود؛ یعنی یک کارمند عادی، یا مدیر یک بخش و یا هم آمر بخش مربوطه، هر کدام صلاحیت‌های استفاده از دیتا و دیتابیس را با در نظر داشت وظیفه‌شان به دست می‌آورند. یک سیستم باید دسترسی به دیتا را برای جوانب مسئول ضمانت کند. زمانی که یک استفاده‌کننده به حیث استفاده‌کننده عادی با صلاحیت دیدن دیتا، دیتابیس را به کار می‌گیرد، باید تمام بخش‌هایی که در آن یوزردیتا ذخیره است را دیده بتواند. در کنار آن اگر استفاده‌کننده عادی با صلاحیت‌های خواندن و نوشتن دیتا در یک سیستم تعریف می‌شود، او باید بتواند در یوزردیتا بر علاوه خواندن، تغییراتی اعمال کرده و بروز رسانی نماید، سپس ریکوردهای جدید دیتا را داخل نموده و یا حتی دیتا را پاک کرده بتواند. به عین ترتیب استفاده‌کننده‌بی که صلاحیت کار با میتادیتا را در یک سیستم داشته باشد، سیستم مدیریت دیتابیس باید اجازه ایجاد و استفاده از دیتابیس، ایجاد جدول‌ها و ساختمان‌های دیگر و غیره کارهای مدیریت دیتا را بدهد.

#### ۴.۴.۱ انواع ایمنی (Security) در سیستم‌های مدیریت دیتابیس

امنیت دیتابیس و ایمن بودن دیتا یکی از بحث‌های وسیع در ساحة کمپیوترساینس به شمار می‌آید. مسایل مختلف در این بخش شامل بوده از جمله به تعدادی از آن‌ها در ذیل به صورت مختصر پرداخته می‌شود:

- مسایل اخلاقی در ارتباط به صلاحیت‌های دسترسی به معلومات: بعضی معلومات شاید در کتگوری معلومات شخصی شامل باشند که اشخاص و ارگان‌های غیر مسئول و شرکت‌های تجارتي نباید به هم‌چو دیتا دسترسی داشته باشد.
- مشکلات مختلف قانونی در ارتباط به صلاحیت‌های دسترسی به معلومات: بعضی معلومات شاید در کتگوری معلومات شخصی شامل باشند که تمام ارگان‌های دولتي نباید به هم‌چو دیتا دسترسی داشته باشد.
- مسایل مربوط پالیسی‌ها به سطوح مختلف دولتي شامل ادارات و شرکت‌های مختلف: هر نوع معلومات و دیتا در مورد اشخاص مربوطه‌شان نباید در دسترس عامه قرار بگیرد. مثال‌هایی هم‌چو دیتا و معلومات در موارد روابط سیاسی، امکانات اقتصادی، صحت و غیره مربوط کارمندان‌شان بوده می‌تواند.
- مسایل مربوط سیستم: کنترل و تطبیق وظایف مختلف امنیتی بالای یک سیستم به این بخش ارتباط می‌گیرد؛ یعنی یک وظیفه امنیتی در سطح فزیکي (سخت‌افزار کمپیوتر) تطبیق شود، یا در سطح سیستم عامل کنترل شود و یا هم توسط سیستم مدیریت دیتابیس (DBMS) کنترل شود.
- ضرورت تشخیص سطوح مختلف امنیتی برای استفاده‌کنندگان: در بعضی ادارات و ارگان‌ها، سطوح مختلف امنیتی برای دیتا و کارمندان مربوطه به حیث استفاده‌کنندگان سیستم تعریف و در نظر گرفته می‌شود. تعیین صلاحیت‌ها و نام‌گذاری آن‌ها شاید از یک اداره تا اداره دیگر متفاوت باشد. مثال‌های سطوح دسترسی به دیتا در یک اداره، عبارت از محرم بودن درجه یک، محرمانه سری و عمومی بوده می‌توانند.

#### ۴.۵ تهدیدها به دیتابیس‌ها (Threats to Databases)

تهدیدات و خطرها که دیتابیس‌ها با آن مواجه است از جمله مسایل مهم به شمار می‌رود. این تهدیدها باعث از دست‌دادن اهداف امنیتی و تنزل دیتابیس‌ها شده می‌توانند. اهداف امنیتی دیتابیس عبارت از درستی (Integrity)، در اختیار بودن (Availability) و محرمانگی (Confidentiality) است. در صورت نبود هر کدام از این‌ها، تهدیداتی متوجه دیتابیس خواهد بود.

- از دست‌دادن دقت و درستی (Loss of Integrity): جهت اطمینان از درست بودن دیتابیس، معلومات باید از پیرایش و تغییرات بی‌جا در امان باشد. پیرایش دیتا شامل ایجاد ست‌های جدید دیتا، داخل کردن دیتا، به‌روز کردن دیتا، تغییر دادن حالت دیتا و پاک کردن دیتا می‌شود. اگر هم‌چون تغییرات بدون مسئولیت به دیتا وارد می‌شود، آن را می‌توان از تهدیدهای دیتابیس نامید. تغییرات دیتا امکان دارد به طور آگاهانه و یا هم ناآگاهانه صورت گیرد. در هر دو حالت، تهدید دیتابیس در قسمت درست بودن (Integrity) گفته می‌شود. در صورت عدم رسیدگی به چنین تهدید و ادامه این



مشکل استفاده از یک سیستم نادرست و دیتای نادرست، زمینه را برای تصامیم نادرست، تقلب و اشتباه بیش‌تر مساعد می‌گرداند.

- از دست دادن در اختیاربودن (Unavailability): یکی دیگر از تهدیدات دیتابیس عبارت از حالتی است که در آن بخش‌هایی از دیتابیس شاید در اختیار استفاده‌کننده ویا برنامه‌ که دیتابیس را به حیث منبع دیتا استفاده می‌کند، نباشد. اطمینان از درست‌کارکردن و در اختیار قرارگرفتن تمامی بخش‌های یک دیتابیس، چنین تهدیدات را کم می‌کند.
  - از دست دادن محرمت (Loss of Confidentiality): در دیتابیس‌ها، بخش‌هایی موجود است که باید به صورت سری نگاه داشته و استفاده‌کنندگان عمومی، دیتای مربوطه را نبینند. در مواردی این مسئله رعایت نشده و هم‌چون معلومات به صورت عمومی شریک ساخته شده و به نحوی در اختیار استفاده‌کنندگان غیر مسئول قرار می‌گیرد. این امر یک نوع دیگری از تهدیدهای دیتابیس به شمار رفته و امکان از دست دادن اعتبار یک اداره و ارگان را به بار خواهد آورد.
- به خاطر حفاظت دیتابیس‌ها در مقابل سه نوع تهدیدی که توضیح شد، چهار نوع اندازه‌کننده، کنترل استفاده می‌شوند که عبارت‌اند از:

۱. کنترل دسترسی (Access Control)

۲. کنترل برداشت و نتیجه‌گیری (Inference Control)

۳. کنترل جریان (Flow Control)

۴. رمزگذاری دیتا (Data Encryption)

## ۴.۶ کنترل دسترسی (Access Control)

یکی از پرابلم‌های عام ایمنی در سیستم‌های کمپیوتری، عبارت از منع اشخاص غیر مسئول از دسترسی به یک سیستم است. منع هم‌چو اشخاص یا بهتر بگوییم استفاده‌کنندگان سیستم به خاطر گرفتن معلومات و یا هم به خاطر آوردن تغییرات منفی و نادرست در قسمت‌هایی از دیتابیس یک امر ضروری پنداشته می‌شود. میکانیزم امنیتی یک سیستم مدیریت دیتابیس باید مقررات وضع محدودیت‌های دسترسی به یک سیستم دیتابیس را داشته باشد. این وظیفه به نام کنترل دسترسی (Access Control) یاد می‌شود.

کنترل دسترسی کمک می‌نماید تا حساب‌های استفاده‌کنندگان (User Accounts) با رمز عبور (Password) ایجاد گردد و از طریق همین حساب‌ها، پروسه‌های داخل‌شدن و استفاده از دیتابیس‌ها (Login Process) توسط سیستم مدیریت دیتابیس کنترل شود. مدیر دیتابیس که در انگلیسی به او Database Administrator گفته می‌شود و به اختصار به حروف DBA نشان داده می‌شود، عبارت از شخص صلاحیت‌دار مرکزی یک سیستم دیتابیس است. یکی از مسئولیت‌های مدیر دیتابیس تنظیم صلاحیت‌ها برای استفاده‌کنندگان دیتابیس است. تقسیم‌بندی استفاده‌کنندگان و دیتا در روشنایی پالیسی‌های اداره مربوطه نیز از جمله مسئولیت‌های مدیر دیتابیس به حساب می‌آید. مدیر دیتابیس در یک سیستم مدیریت دیتابیس، یک حساب استفاده از سیستمی را به نام DBA Account دارا است. این نوع حساب به نام‌های حساب سیستم

(System Account) و یا حساب استفاده‌کننده سوپر (Superuser Account) نیز یاد می‌شوند. حساب مدیر دیتابیس، صلاحیت‌های بالا را در قسمت کنترل و تنظیم سیستم دارا می‌باشد. حساب مدیر دیتابیس صلاحیت کامل به خاطر ایجاد، کنترل و حذف حسابات استفاده‌کنندگان عادی را در سیستم‌های مدیریت دیتابیس دارد.

حساب مدیر دیتابیس یا DBA طرزالعمل‌هایی را به راه انداخته می‌تواند که از طریق آن صلاحیت‌های استفاده از سیستم به حساب‌های انفرادی، استفاده‌کنندگان یا گروه‌های از استفاده‌کنندگان در موارد پایین داده شده و یا از ایشان گرفته می‌شود:

- ایجاد حساب (Account Creation): از طریق حساب استفاده‌کننده سوپر، یک حساب جدید و رمز عبور ایجاد شده می‌تواند. حساب جدید می‌تواند برای استفاده یک شخص باشد و یا هم برای گروهی از استفاده‌کنندگان ایجاد شود. با انجام این کار استفاده‌کننده و یا گروهی از استفاده‌کنندگان قادر به استفاده از سیستم مدیریت دیتابیس خواهند شد.
- دادن صلاحیت (Privilege Granting): با این امکانات یک مدیر دیتابیس می‌تواند صلاحیت‌ها و امتیازهای لازم استفاده از سیستم را به استفاده‌کنندگان بدهد.
- فسخ صلاحیت (Privilege Revocation): یک مدیر دیتابیس می‌تواند صلاحیت‌هایی را که قبلاً به حساب‌های استفاده‌کنندگان داده شده، فسخ نموده و حسابات‌شان را بسته نماید.
- تعیین سطح امنیت (Security Level Assignment): تعیین حساب‌های استفاده‌کنندگان به یک سطح مناسب پاک‌سازی امنیتی نیز جزء از صلاحیت‌های حساب DBA به شمار می‌رود.

مدیر دیتابیس (DBA) مسئول تمام مسایل مربوط به امنیت سیستم دیتابیس است. صلاحیت اول در فهرست بالا (ایجاد حساب) در مجموع به خاطر کنترل دسترسی به سیستم مدیریت دیتابیس است. صلاحیت‌های دو و سه (دادن و گرفتن صلاحیت‌ها به استفاده‌کنندگان عادی) به خاطر کنترل اختیارهای دیتابیس در نظر گرفته شده و صلاحیت ذکرشده چهارم (تعیین سطح امنیتی حسابات) همانا به خاطر کنترل اختیارهای اجباری به کار گرفته می‌شود.

دیتابیس‌های احصائیوی به خاطر ارائه معلومات و احصائیه‌ها یا خلاصه‌های مقادیر، به اساس معیارهای مختلف استفاده می‌شوند. به گونه مثال اگر دیتابیس احصائیوی، نفوس یک مملکت در نظر گرفته شود، این نوع دیتابیس، احصائیة نفوس را به اساس گروه‌های سنی، سطوح درآمد، اندازه خانواده، سطوح تعلیمی و غیره ارائه خواهد کرد.

استفاده‌کنندگان این نوع دیتابیس‌ها تنها اجازه دیدن و به کارگیری گراف‌های حاصله از نتایج کیوری‌های آن‌ها را خواهند داشت. به خاطر مسایل امنیتی و مشکلات استفاده نادرست، معلومات تفصیلی و ارقامی که گراف‌ها و راپورهای خلاصه به اساس آن‌ها توسط سیستم مدیریت دیتابیس تهیه شده‌اند، نباید در اختیار استفاده‌کنندگان داده شود. امنیت چنین دیتابیس‌ها باید اطمینان از اجراآت فوق را داشته باشد و به استفاده‌کنندگان اجازه دسترسی به معلومات شخصی افراد درج‌شده در دیتابیس را ندهد.



## ۴.۷ کنترل‌های برداشت و نتیجه‌گیری (Inference Control)، جریان (Flow Control) و رمزگذاری دیتا (Data Encryption)

سنجش و اندازه‌کنترل متناظر، به نام کنترل‌های برداشت و نتیجه‌گیری (Inference Control) یاد می‌شوند. کنترل‌های جریان (Flow Control) یکی دیگر از مسایل مربوط امنیت را تشکیل می‌دهد. در این نوع کنترل‌ها از انتشار معلومات به اشخاص غیر مسئول جلوگیری صورت می‌گیرد. راه‌هایی که معلومات از طریق آن‌ها به صورت پنهانی به مراجع غیر مسئول به صورت غیرقانونی انتقال پیدا می‌کند به نام چینل‌های پنهان (Covert Channels) یاد می‌شوند. از تفصیل بیش‌تر در دو مورد کنترل‌های برداشت و نتیجه‌گیری و کنترل‌های جریان صرف نظر گردیده است.

چهارمین یا آخرین اندازه‌کننده کنترل دیتا، رمزگذاری دیتا (Data Encryption) است. طریقه‌های رمزگذاری دیتا در اصل به خاطر مقاصد نظامی به میان آمده و رمزگشایی آن‌ها مسئلهٔ بس دشوار بوده، به سادگی پی بردن به آن ممکن نیست. این نوع رمزگذاری به خاطر حفاظت از دیتای حساس مانند شمارهٔ کارت حساب بانکی اشخاص به کار می‌رود. شماره‌های حساب بانکی در شبکه‌های ارتباطات به خاطر معاملات پولی استفاده می‌شوند. رمزگذاری می‌تواند در دیتابیس‌ها نیز برای قسمت‌های مهم و دیتای حساس به کار برده شود.

در دیتابیس‌ها به خاطر انجام این کار، دیتای مورد نظر به کمک الگوریتم‌های کودگذاری به کود جدید تبدیل می‌شود. استفاده‌کنندگان غیر مسئول که دسترسی به این دیتاها را داشته باشند، به آسانی قادر به خواندن دیتای رمزگذاری شده نخواهند بود. در حالی که استفاده‌کنندگان مسئول با کلید رمزگشایی که توسط سیستم برای‌شان به نحوی تهیه می‌شود، قادر خواهند بود تا دیتای رمزگذاری شده را بازخوانی کنند.

## ۴.۸ ایجاد حساب‌های استفاده‌کنندگان

طوری که در درس‌های قبلی بحث شد، به خاطر استفاده مناسب از دیتابیس و به خاطر تطبیق مسایل امنیتی در دیتابیس‌ها، حساب‌های استفاده‌کنندگان با صلاحیت‌های معین تعریف و ایجاد شده می‌تواند. جهت انجام این کار در سیکویل، طرزالعملی به نام ایجاد (CREATE ...) استفاده می‌شود. توسط این طرزالعمل استفاده‌کنندگان با صلاحیت‌های ذیل ایجاد شده می‌توانند:

- ایجاد حساب مدیران دیتابیس (Database Administrators)
- ایجاد حساب استفاده‌کنندگان عادی (Users)

طرزالعمل CREATE ... در اصل جزء طرزالعمل‌های کنترل دیتا (Data Control Language) سیکویل نیست. این طرزالعمل، شامل طرزالعمل‌های تعریف دیتا در سیکویل (Data Definition Language) است. در این بحث طرزالعمل ایجاد به حیث یک طرزالعمل کنترل دیتا و بخش مدیریت دیتابیس به کار گرفته شده است.

## ۴.۸.۱ ایجاد حساب مدیران دیتابیس

بخش امنیت دیتابیس‌ها برمی‌گردد به مدیران دیتابیس‌ها که در اختصار به نام DBA یاد می‌شوند. مدیران دیتابیس‌ها مسئولیت تعریف و کنترل دیتابیس‌ها را به عهده دارند. از آدرس امنیت دیتابیس‌ها، مدیران دیتابیس‌ها مسئول فعالیت‌های ذیل در یک سیستم‌اند:

- اضافه‌ی استفاده‌کنندگان؛
  - حذف استفاده‌کنندگان؛
  - اجازه‌ی دسترسی به بخش‌های معین در یک دیتابیس؛
  - محدود ساختن دسترسی به بخش‌های یک دیتابیس؛
  - دادن امتیاز و صلاحیت به خاطر دیدن و تغییر آوردن در بخش‌های دیتابیس؛
  - تغییر آوردن و یا مسترد کردن صلاحیت‌ها و امتیازاتی که قبلاً به استفاده‌کنندگان داده شده است.
- استفاده‌کننده‌یی که یک دیتابیس را ایجاد می‌کند، به طور خودکار مدیر آن دیتابیس محسوب شده و از صلاحیت‌های مدیریت دیتابیس بر دیتابیس تازه ایجاد شده برخوردار است. پس همین استفاده‌کننده‌ی اول صلاحیت تنظیم و ایجاد حساب‌های جدید مدیریتی برای همین دیتابیس را دارد.

## ۴.۸.۲ ایجاد و حذف حساب‌های استفاده‌کنندگان عادی

ایجاد حساب استفاده‌کنندگان عادی از طریق حساب مدیریت دیتابیس در سیکویل امکان‌پذیر است؛ یعنی شخصی که دسترسی به حساب مدیر دیتابیس DBA را داشته باشد، می‌تواند حساب استفاده‌کننده‌ی عادی را ایجاد نماید. طرزالعمل ایجاد یک حساب برای استفاده‌کننده در سیکویل طور ذیل است:

```
CREATE USER 'UserName' IDENTIFIED BY 'Password'
```

در مثال ذیل یک حساب استفاده‌کننده با نام احمد و با رمز عبور abc در سیکویل ایجاد می‌شود.

```
>> CREATE USER 'Ahmad' IDENTIFIED BY 'abc';
```

در صورتی که حساب استفاده‌کننده به نام محمود بدون پاسورد یا رمز عبور ایجاد شود طور ذیل تایپ می‌شود:

```
>> CREATE USER 'Mahmod';
```

حذف حساب استفاده‌کنندگان مانند ایجاد حساب‌های استفاده‌کنندگان از صلاحیت‌های مدیران دیتابیس‌ها به شمار می‌رود. در مثال‌های قبلی دیده شد که حساب‌هایی به نام‌های احمد و محمود یکی با رمز عبور و دیگری بدون رمز عبور ایجاد شدند. به خاطر حذف کردن حساب‌های استفاده‌کنندگان در سیکویل طرزالعمل DROP USER ... به کار برده می‌شود. شکل عمومی طرزالعمل حذف استفاده‌کنندگان طور ذیل است:

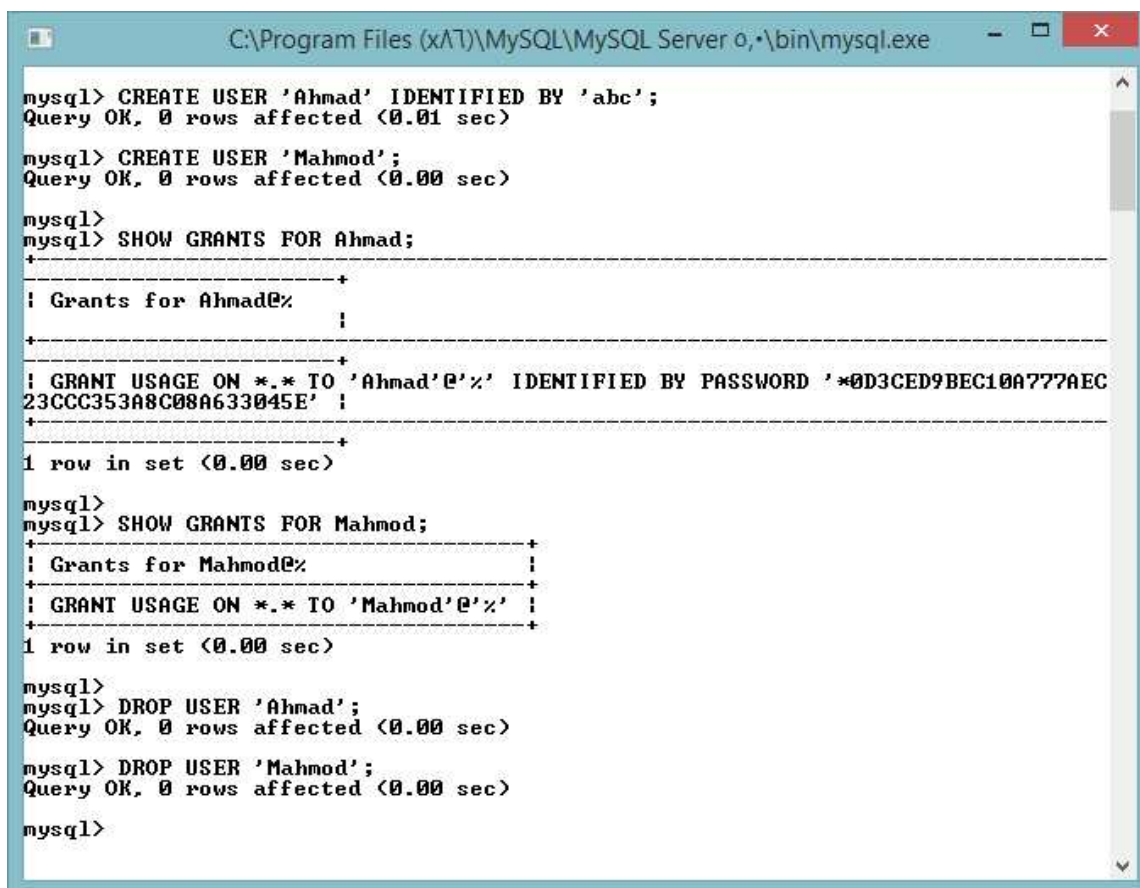
```
DROP USER 'Username'
```

مثال‌های ذیل حساب‌های قبلاً ایجاد شده به نام‌های احمد و محمود را حذف می‌نماید.

>> DROP USER 'Ahmad';

>> DROP USER 'Mahmod';

شکل پایین تطبیق طرزالعمل‌های ایجاد استفاده‌کنندگان، دیدن استفاده‌کنندگان، جدیداً ایجاد شده توسط طرزالعمل ... SHOW GRANTS و حذف استفاده‌کنندگان با طرزالعمل ... DROP USER را نشان می‌دهد.



```
C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin\mysql.exe

mysql> CREATE USER 'Ahmad' IDENTIFIED BY 'abc';
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE USER 'Mahmod';
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> SHOW GRANTS FOR Ahmad;
+-----+
| Grants for Ahmad@% |
+-----+
| GRANT USAGE ON *.* TO 'Ahmad'@'%' IDENTIFIED BY PASSWORD '*0D3CED9BEC10A777AEC23CCC353A8C08A633045E' |
+-----+
1 row in set (0.00 sec)

mysql>
mysql> SHOW GRANTS FOR Mahmod;
+-----+
| Grants for Mahmod@% |
+-----+
| GRANT USAGE ON *.* TO 'Mahmod'@'%' |
+-----+
1 row in set (0.00 sec)

mysql>
mysql> DROP USER 'Ahmad';
Query OK, 0 rows affected (0.00 sec)

mysql> DROP USER 'Mahmod';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

شکل ۴-۴ ایجاد حساب استفاده‌کنندگان، دیدن حساب‌ها و حذف کردن حساب‌های ایجاد شده در سیکویل

در شکل بالا دو بار طرزالعمل ایجاد استفاده‌کننده جدید به کار گرفته شده و دو حساب استفاده‌کننده، ایجاد شده است. در صورتی که به همین نام حساب قبلی در سیستم وجود داشته باشد، سیستم مدیریت دیتابیس پیام غلطی صادر خواهد نمود. به خاطر جلوگیری از هم‌چون اشتباه، قبل از ایجاد یک حساب به یک نام معین، می‌توان طرزالعمل حذف کردن حساب (DROP USER ...) را تطبیق نموده و به تعقیب آن طرزالعمل‌های ایجاد حساب به نام معین را به کار برد. البته مواردی که گفته شد، به خاطر تمرین و کار کردن با موضوع مدیریت حساب‌های استفاده‌کنندگان به صورت امتحانی می‌باشد. در دیتابیس‌هایی که با دیتای واقعی کار می‌نمایند و استفاده‌کنندگان واقعی موجود باشند، نباید چنین کاری بدون اطلاع قبلی صورت گیرد.

طرز العمل دیگری که در شکل بالا استفاده شده است، عبارت از طرز العمل دیدن امتیازهای یک استفاده کننده مشخص است. این طرز العمل عبارت از طرز العمل SHOW USER ... است. طوری که در شکل دیده می شود، این طرز العمل با شکل عمومی پایین به کار رفته و نتیجه آن در شکل نشان داده شده است.

SHOW USER UserName

طریقه های دیگری نیز به خاطر ایجاد حساب های استفاده کنندگان موجود بوده و با امکانات بیش تر مدیریتی قابل استفاده است. در ذیل با استفاده از طرز العمل GRANT ... برای ایجاد چهار حساب استفاده کننده در سیکویل نشان داده شده است.

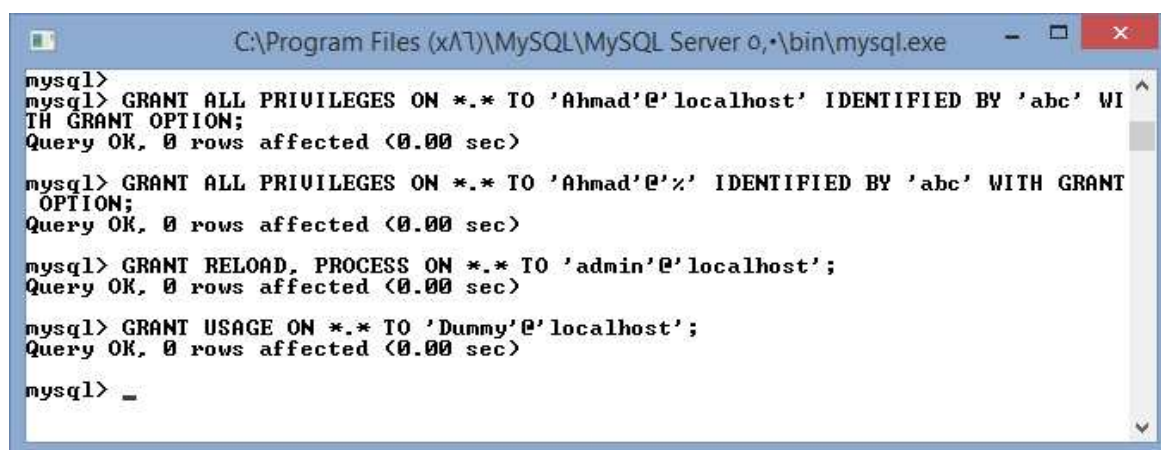
```
>> GRANT ALL PRIVILEGES ON *.* TO 'Ahmad'@'localhost' IDENTIFIED BY 'abc'
WITH GRANT OPTION;
```

```
>> GRANT ALL PRIVILEGES ON *.* TO 'Ahmad'@'%' IDENTIFIED BY 'abc' WITH
GRANT OPTION;
```

```
>> GRANT RELOAD, PROCESS ON *.* TO 'admin'@'localhost';
```

```
>> GRANT USAGE ON *.* TO 'Dummy'@'localhost';
```

نتیجه تطبیق طرز العمل های بالا در شکل ذیل نشان داده شده است.



شکل ۴-۵ تطبیق طرز العمل های ایجاد حساب های استفاده کنندگان در سیکویل با استفاده از طرز العمل GRANT

حساب های استفاده کنندگان که توسط طرز العمل های فوق ایجاد شده اند دارای مشخصات زیر می باشند:

- دو حساب در طرز العمل ها به نام احمد با رمز عبور abc ساخته شده است. هر دو حساب استفاده کننده سوپر با تمام صلاحیت های کنترولی می باشند. حساب اولی 'Ahmad'@'localhost' تنها می تواند از کمپیوتر محلی (Local Host) به دیتابیس وصل شود. حساب دومی به همین نام 'Ahmad'@'%' می تواند از هر جا به دیتابیس وصل شود. استفاده از سمبول % به همین منظور به کار گرفته شده و

این امکان را مساعد می‌سازد تا استفاده‌کننده به صورت آزادانه از ترمینل‌های مختلف بتواند به سیستم دسترسی پیدا کند.

- مثال سومی در شکل بالا حساب 'admin'@'localhost' را به شکلی ایجاد می‌کند که در آن پاسورد یا رمز عبور تعریف نشده؛ یعنی بدون رمز عبور، حساب استفاده‌کننده ایجاد شده است. این حساب تنها توسط مدیر به کار گرفته شده که با استفاده آن از کمپیوتر محلی (Local Host) به دیتابیس وصل شده می‌تواند. طوری که در متن طرزالعمل دیده می‌شود به این حساب تنها صلاحیت‌های مدیریتی RELOAD و PROCESS داده شده است. به خاطر علاوه‌کردن صلاحیت‌های بیش‌تر بازهم از طرزالعمل GRANT ... استفاده شده می‌تواند.

- حساب چهارمی ایجاد شده در طرزالعمل‌های بالا عبارت از 'dummy'@'localhost' است که آن‌هم بدون رمز عبور ایجاد شده است. این حساب نیز تنها از کمپیوتر محلی (Local Host) به سیستم وصل شده می‌تواند. برای این حساب هیچ صلاحیتی تعریف نشده است. استفاده از کلمه USAGE در طرزالعمل GRANT امکان ایجاد حساب‌های بدون هیچ‌گونه صلاحیت را می‌دهد؛ یعنی یک حساب استفاده‌کننده، طوری ایجاد می‌شود که تمام صلاحیت‌های آن در سیستم با اختیار N نفی شده باشند. در این طرزالعمل نیز از همین اختیار استفاده شده است. در چنین مواردی یک مدیر دیتابیس می‌تواند امکاناتی را استفاده کند که از طریق آن، حساب بدون صلاحیت ایجاد نموده و در وقت ضرورت صلاحیت‌های لازم را برای هم‌چون حساب‌ها تعریف نماید.



## خلاصه فصل چهارم

در این فصل موضوعاتی از قبیل مدیریت دیتا و مدیریت استفاده کنندگان دیتا با ذکر مثال‌های مناسب و تطبیق مثال‌ها در سیکویل توضیح داده شد. به خاطر مدیریت بهتر دیتا، ساختمان‌هایی به نام اندکس‌ها مورد بحث قرار گرفته و اهمیت اندکس‌ها با مثال‌های مشابهی در زندگی روزمره نشان داده شدند. استفاده و به کارگیری اندکس‌ها به خاطر پیدا کردن یک موضوع از یک کتاب با حجم بیش‌تر از 2000 صفحه در ظرف چند دقیقه محدود، از جمله مثال‌های استفاده از اندکس‌ها در ساحات غیر از دیتابیس است. انواع اندکس‌ها در دیتابیس و نحوه کارشان و نیز استفاده از اندکس‌ها در تنظیم دیتا تا حدودی با تفصیل لازم به توضیح گرفته شد. طرزالعمل‌های اضافه کردن، به کارگیری و حذف اندکس‌ها در سیکویل با مثال‌های عملی نشان داده شده است.

کنترل دسترسی به دیتا، دسته دیگری از مسایل پیشرفته در دیتابیس‌ها به شمار رفته و در این فصل به طور مفصل به بحث گرفته شد. موضوعات عملی در این بخش شامل کار با حساب‌های استفاده کنندگان در نظر گرفته شده و طریقه‌های ایجاد حساب‌های جدید استفاده کنندگان با شکل عمومی کود سیکویل و با ذکر مثال‌های مربوطه آن نشان داده شد. تنظیم صلاحیت‌های لازم برای استفاده کنندگان و داشتن کنترل دسترسی به حساب‌های استفاده کنندگان موضوعات جالب و قابل بحثی را به محتویات فصل اضافه نموده‌اند. در نهایت حذف استفاده کنندگان موجود با مثال‌های لازم در زمینه به شکل طرزالعمل‌های عمومی و نیز طرزالعمل‌های اجرایی در سیکویل شرح داده شده و نتایج آن به شکل عملی به کمک تصاویر صفحه کمپیوتر نشان داده شده است. قسمت اخیر فصل با سؤالات مربوط به محتویات و فعالیت‌های لازم برای شاگردان، آذین گردیده است.



## فعالیت های فصل چهارم

۱. یک دیتابیس را به نام فروشگاه (Store) ایجاد کنید.
۲. دیتابیس فروشگاه را فعال سازید.
۳. در دیتابیس فروشگاه جدولی به نام اشیاء (Objects) با فیلدهای نمبر، نام، نوع و تعداد ایجاد کنید.
۴. فیلد نمبر را در جدول اشیاء به حیث کلید اولیه تنظیم کنید.
۵. با استفاده از طرزالعمل ... DESCRIBE ساختمان جدول جدید (جدول اشیاء) را ببینید.
۶. به جدول اشیاء حدود 10 ریکورد دیتای مناسب داخل کنید.
۷. با استفاده از کیوری ... WHERE ... SELECT محتویات جدول اشیاء را ببینید. (در حالتی که اندکس استفاده نشده است).
۸. با استفاده از طرزالعمل ... CREATE INDEX فیلد تعداد را اندکس بسازید.
۹. دوباره کیوری ... WHERE ... SELECT را به راه بیاندازید. (در حالتی که اندکس استفاده شده است).
۱۰. تفاوت در زمان اجرای کیوری های بالا را مقایسه کنید، در صورت عدم تفاوت در نتیجه، تعداد ریکوردها را در جدول اشیاء اضافه نموده و دوباره کیوری ها را در حالات بدون استفاده از اندکس و با استفاده از اندکس تمرین کنید. (به خاطر پاک کردن اندکس از فعالیت شماره بعدی استفاده کنید)
۱۱. با استفاده از طرزالعمل ... DROP INDEX اندکس تعبیه شده در جدول اشیاء را دوباره دور کنید.
۱۲. با استفاده از طرزالعمل ... SHOW INDEX اندکس جدول اشیاء را ببینید که آیا پاک شده است و یا خیر؟
۱۳. با استفاده از طرزالعمل ... CREATE USER یک حساب استفاده کننده به نام حمید با پاسورد 123 ایجاد کنید.
۱۴. با استفاده از طرزالعمل ... SHOW GRANTS FOR معلومات در مورد حساب جدید به نام حمید را نمایش دهید.
۱۵. با استفاده از طرزالعمل ... DROP USER حساب ایجاد شده در فعالیت قبلی را حذف کنید.
۱۶. با استفاده از طرزالعمل ... GRANT یک بار دیگر حساب استفاده کننده یی به نام حمید با رمز عبور 123 را طوری ایجاد کنید که حمید بتواند تنها از کمپیوتر محلی (Local Host) به دیتابیس وصل شود.
۱۷. با استفاده از طرزالعمل ... GRANT حساب استفاده کننده یی به نام حمید با رمز عبور 123 را طوری ایجاد کنید که حمید بتواند از هر موقعیت شبکه به دیتابیس وصل شود.



## سوالات فصل چهارم

۱. اندکس‌ها در دیتابیس چیست؟ توضیح دهید.
۲. اندکس‌ها در کتاب‌ها چه‌طور استفاده می‌شوند؟
۳. اندکس‌های اولیه را توضیح دهید.
۴. اندکس‌های دومی را توضیح دهید.
۵. اندکس‌های کلهستری را با مثال توضیح دهید.
۶. تفاوت بین دیتای تنظیم‌شده (Ordered Data) و دیتای غیر تنظیم‌شده (Unordered Data) را واضح سازید.
۷. اندازه فایل اندکس برای یک جدول نظر به سایز خود جدول بر روی دیسک کوچک‌تر است، چرا؟ توضیح دهید.
۸. شکل عمومی ایجاد اندکس در سیکویل را بنویسید.
۹. شکل عمومی طرز‌العمل دیدن اندکس‌های یک جدول در سیکویل را بنویسید.
۱۰. اهمیت طرز‌العمل SHOW INDEX ... را به زبان خود به طور مختصر بیان کنید.
۱۱. شکل عمومی طرز‌العمل پاک کردن اندکس‌های یک جدول را در سیکویل بنویسید.
۱۲. یک جدول از دیتابیزی در سیکویل با استفاده از کدام طرز‌العمل پاک می‌شود؟ با مثال بنویسید.
۱۳. ساختمان یک جدول با کدام طرز‌العمل سیکویل دیده می‌شود؟
۱۴. مفهوم کنترل دسترسی (Access Control) چه بوده و در کدام موارد استفاده می‌شوند؟
۱۵. DBA اختصار کدام کلمات بوده و در ساحة دیتابیس به چه مفهوم است؟
۱۶. طرز‌العمل ایجاد یک حساب استفاده‌کننده در سیکویل را با ذکر یک مثال بنویسید؟
۱۷. طرز‌العمل حذف یک حساب استفاده‌کننده در سیکویل را با ذکر یک مثال بنویسید؟



## فصل پنجم

### مدل‌های دیتا (Data Models)



**هدف کلی:** محصلان با مدل‌های دیگر دیتا مانند XML، Text و Multimedia شناخت حاصل نمایند.

**اهداف آموزشی:** در پایان این فصل محصلان قادر خواهند شد تا:

۱. فارمت (eXtensible Markup Language) XML را توضیح کنند.
۲. نحوه نوشتن کد XML را بفهمند.
۳. معلومات در مورد مدل‌های دیتا XML داشته باشند.
۴. مدل‌های دیتای متنی را توضیح دهند.
۵. انواع دیتای متنی در دیتابیس را به اساس ضرورت‌های آن شناسایی و استفاده کنند.
۶. در مورد دیتای مولتی‌مدیا و فارمت‌های آن معلومات حاصل کنند.

مدل‌های دیتا به خاطر مدیریت بهتر و استفاده مؤثر از آن در ساحة دیتابیس‌ها و تنظیم دیتا از اهمیت خاصی برخوردار است. زمانی که از مدل‌های دیتا نام گرفته می‌شود، اشخاص مسلکی ساحة مدیریت و انکشاف دیتابیس‌ها به فکر مدل‌های تحلیل دیتا جهت ایجاد دیتابیس‌ها می‌افتند که این یک امر طبیعی است. به خاطری که اصطلاحات مورد استفاده در هر ساحة، مفهوم خاصی را افاده کرده و کارکنان یک ساحة تخنیکی، آن اصطلاح را از دید خود نگریسته و تحلیل می‌نمایند. در این فصل، هدف از مدل‌های دیتا به مدیریت دیتا در دیتابیس‌ها برمی‌گردد؛ یعنی چگونه دیتای قابل استفاده در کمپیوتر (دیتای دیجیتل) در دیتابیس‌ها با استفاده از فارمت‌های استندرد به کار گرفته شود.

یک سِتی از دیتا امکان دارد با فارمت مشخصی موجود باشد، آیا همین فارمت موجوده دیتا، مناسب است تا در کمپیوتر و دیتابیس استفاده شود؟ و یا کدام مدل دیگر دیتا به شکل بهتر می‌تواند همین دیتا را در کمپیوتر و دیتابیس‌ها به استفاده بگذارد. به طور مثال دیتای متنی، با فارمت فایل برنامه میکروسافت ورد موجود است. آیا همین فارمت برای دیتای متذکره خوب است و یا فارمت XML بهتر خواهد بود. معلومات در مورد مدل‌های دیتای دیجیتل و نیز خصوصیات استفاده از آن‌ها رهنمودهای روشنی در مورد ارائه خواهند نمود. مدل‌های دیتای قابل ذخیره و استفاده در کمپیوتر موجوداند. این مدل‌ها شامل فارمت‌های مشخصی بوده که بیش‌تر مورد استفاده قرار می‌گیرند. یکی از این فارمت‌ها طوری که پیش‌تر به آن اشاره گردید، عبارت از فارمت XML بوده که بر گرفته شده از کلمات eXtensible Markup Language است. XML یک لسان متنی بوده و دارای فارمت‌های مشخص برای نوشتن طرزالعمل‌ها است. لسان XML یک لسان مکمل پروگرام‌نویسی نبوده صرف به خاطر ذخیره و استفاده دیتا به کار گرفته می‌شود. مدل دیتای متنی (TEXT Data Model) نوع دیگری از دیتا در کمپیوتر و دیتابیس‌ها است که قسمت اعظم معلومات آن از طریق کمپیوترها و دیتابیس‌ها ذخیره و استفاده می‌شود. انواع دیتای متنی در دیتابیس‌های سیکویل تعریف و استفاده می‌شود، تفاوت‌های مشخصی بین انواع دیتای متنی موجود بوده که هر کدام مزایا و نواقص خود را در حالات مشخص دارد. در بعضی موارد فارمت‌هایی برای مدل‌های دیتا موجوداند که کمک زیادی در قسمت اجرای درست عملیه‌ها و نیز در قسمت استفاده کم‌تر حافظه کمپیوتر محسوس می‌شوند.

به خاطر پیدا و یا به اجراء گذاشتن مواردی چون معلومات عمومی و نتایج تجارب از اهمیت به‌سزایی برخوردار است. مولتی‌مدیا (Multimedia) انواع دیگر مدل‌های دیتا در کمپیوترساینس به حساب می‌آید. استفاده از مولتی‌مدیا تأثیر مستقیم به ارتباط استفاده‌کننده با کمپیوتر دارد. به هر اندازه‌یی که معلومات جالب ساخته شود به همان اندازه، انتقال آن‌ها به استفاده‌کننده بهتر می‌شود؛ به عبارت دیگر، رُل مولتی‌مدیا در استفاده بهتر از کمپیوتر مستقیماً هویدا است. به کارگیری درست انواع دیتای مولتی‌مدیا زمینه را به خاطر سرعت کار کمپیوتر و سیستم مدیریت دیتابیس فراهم ساخته و مؤثریت انتقال دیتا را بیش‌تر می‌سازد. عناصر مولتی‌مدیا شامل تصویرها و فارمت‌های مختص به آن‌ها، فایل‌های صوتی و ویدیویی و نیز انیمیشن‌ها (Animations) می‌باشد.

## ۵.۱ فارمت XML (eXtensible Markup Language)

XML اختصار کلمات eXtensible Markup Language بوده و عبارت از یک نوع دیتا به فارمت مشخصی در کمپیوترها بوده که از آن استفاده می‌شود. لسان XML یک لسان متنی بوده که طرزالعمل‌های آن از لسان استاندارد به نام SGML که اختصار کلمات Standar Generalized Markup Language است، گرفته شده است. لسان متنی به این معنی که طرزالعمل‌های لسان XML با استفاده از کلمات و حروف لسان نوشتاری انگلیسی نوشته و استفاده می‌شوند. در این لسان تگ‌هایی (Tags) به کار می‌روند که وظیفه آن‌ها به خاطر شناسایی دیتا در لسان XML است. تگ‌های لسان XML جهت ذخیره و تنظیم دیتا استفاده می‌شوند. تگ‌ها در لسان‌های دیگر به اهداف مختلف به کار گرفته می‌شوند. طور مثال در لسان HTML تگ‌ها به خاطر نمایش د معلومات و دیتا روی صفحه وب استفاده می‌شوند. لسان XML به لسان‌های دیگر مانند لسان HTML در قسمت نمایش و تنظیم بهتر دیتا کمک می‌نماید اما نه به مثابه‌یی که لسان XML به حیث لسان جایگزین برای HTML باشد. توسط این لسان اکثر امکاناتی HTML به شکلی انکشاف داده می‌شوند تا نتایج بهتری از آن‌ها به دست بیاید.

در XML سه خصوصیت مهم موجود است که کارایی این نوع دیتا را در سیستم‌های مختلف بالا می‌برد.

- XML گسترش‌پذیر است.
  - XML اجازه می‌دهد تا استفاده‌کننده، تگ‌های مورد ضرورت خود را تعریف و به آن اضافه کند. تگ‌های تعریف‌شده و اضافه‌شده جدید، توسط استفاده‌کننده مانند تگ‌های عادی XML کار می‌نمایند.
  - XML دیتا را ذخیره می‌کند، نه این که دیتا را نمایش بدهد
  - در فارمت XML دیتا ذخیره‌شده می‌تواند. به خاطر نمایش دادن دیتا، طریقه‌های مختلف استفاده شده می‌تواند طوری که XML در قسمت نمایش دیتا مسئولیت نداشته و خصوصیات نمایش بستگی به برنامه و تکنالوژی مورد استفاده می‌داشته باشد.
  - XML یک استاندارد عمومی است استاندارد XML توسط اورگانایزیشن Word Wide Web Consortium (W3C) انکشاف داده شده و با کد مکمل آن به شکل منبع باز (Open Source) به همه استفاده‌کنندگان میسر و قابل دسترس است.
- موارد استفاده از XML شامل فهرست پایین بوده ولی محدود به آن نیست.**

- XML در عقب پرده توسط یک برنامه استفاده شده می‌تواند طوری که توسط آن صفحات HTML برای وبسایت‌های بزرگ ساخته شوند.
- فارمت XML به خاطر تبادل معلومات بین اداره‌ها و سیستم‌ها به کار برده می‌شود.
- فارمت XML به خاطر گرفتن (Offloading) و دوباره لودکردن (Reloading) یک دیتابیس استفاده می‌شود.

- فارمت XML به خاطر ذخیره و مرتب کردن دیتای مورد ضرورت استفاده می شود.
  - XML با فارمت هایی که به خاطر آمیختن با استایل شیت ها (Style Sheets) جهت ایجاد خروجی های مناسب ضروری اند، به کار گرفته می شود.
  - هر نوع دیتا قبل از ذخیره در کامپیوتر تا اندازه یی با XML مدیریت شده می تواند.
- یکی دو موضوع مربوط به XML به صورت سؤال ها مطرح شده و در عناوین فرعی توضیح می شوند.

### ۵.۱.۱ مفهوم کلمه Markup چه است؟

طوری که در بالا نیز گفته شد، XML یک لسان مارکاپ (Markup) است. به این مفهوم که دسته یی از رُل ها به خاطر کودگذاری (رمزگذاری) سندها، طوری استفاده می شوند که فارمت آن ها قابل خواندن توسط انسان ها و نیز توسط کامپیوتر است؛ یعنی کلمات و حروف لسان انگلیسی در قسمت کودگذاری استفاده می شود. این کلمات به مفهوم اصلی شان در لسان های انسانی بوده و توسط ماشین (کامپیوتر) نیز با عین مفهوم قابل خواندن و شناسایی می باشند. مفهوم کود استفاده شده (کلمات معنی دار) و نیز ارتباط بین بخش های معلومات به صورت واضح در لسان های مارکاپ به مشاهده می رسد.

به طور واضح تر یک لسان مارکاپ مجموعه یی از سمبول هایی را شامل می شود که این سمبول ها می تواند در متن یک فایل استفاده شوند. سمبول های جا به جا شده در متن، قسمت های مشخص در داخل سند (Document) را نام گذاری کرده و قسمت های مورد نظر را از قسمت های دیگر آن متمایز می سازد. مثال پایین قطعه کود XML را نشان می دهد. این کود را می توان در بین یک متن آورد.

```
<message>
```

```
<text>Hello, World! </text>
```

```
</message>
```

قطعه کود بالا سمبول های مارکاپ یا تگ های لسان XML را در بر دارد. تگ ها با علامه های استندرد <، > و / احاطه و نشان داده شده اند. تگ های <message>...</message> بلاک اصلی این قطعه کود XML را با تگ های آغاز و انجام نشان می دهد و تگ های <text>...</text> قسمت متنی را نشان می دهد که ذخیره شده است و آماده پروسس است. تگ های آغازین فقط نام تگ در بین علامه های مربوطه بوده و تگ های بسته کننده قسمت کود با یک خط سلس (/) قبل از نام تگ نشانی می شوند.

تگ ها به صورت جوهره یی استفاده می شوند؛ یعنی زمانی که یک تگ باز شده باشد، به خاطر داشتن نتیجه همان تگ، باید در یک قسمت بسته شده باشد. در غیر آن صورت کود XML ناقص بوده و در زمان اجراء توسط کامپیوتر پیام غلطی نشان داده خواهد شد. تنها یک نوع تگ به صورت یک تایی استفاده می شود و آن هم عبارت از تگ اختیاری به نام تگ اعلامیه است که بعدتر توضیح می شود.

## آیا XML یک لسان پروگرام‌نویسی است؟

یک لسان پروگرام‌نویسی شامل قواعد گرامری مختص به خود بوده، کلمات و طرزالعمل‌های مشخص را به خاطر نوشتن پروگرام‌های کمپیوتر به کار می‌برد. پروگرام‌هایی که توسط یک لسان پروگرام‌نویسی ایجاد می‌شوند کمپیوتر را به خاطر انجام یک کار معین رهنمایی می‌کند. به طور مثال گرفتن دیتا از ترمینل و به اجراء درآوردن وسایل جانبی کمپیوتر به خاطر انجام‌دادن یک فعالیت فیزیکی و غیره را می‌توان به حیث خروجی یک پروگرام کمپیوتر یاد کرد. این نوع XML توانایی‌های اجرای یک پروگرام مکمل را ندارد، بنابراین، یک لسان پروگرام‌نویسی گفته نمی‌شود. توسط XML انجام محاسبات و یا اجرای الگوریتم‌ها ممکن نیست. محتویات و کد SQL معمولاً در یک فایل متنی ذخیره می‌شود. فایل‌های XML توسط نرم‌افزارهای مخصوص که قادر به خواندن دیتا از این نوع فایل‌ها است، به کار گرفته شده و کد آن‌ها به اجراء آورده می‌شود.

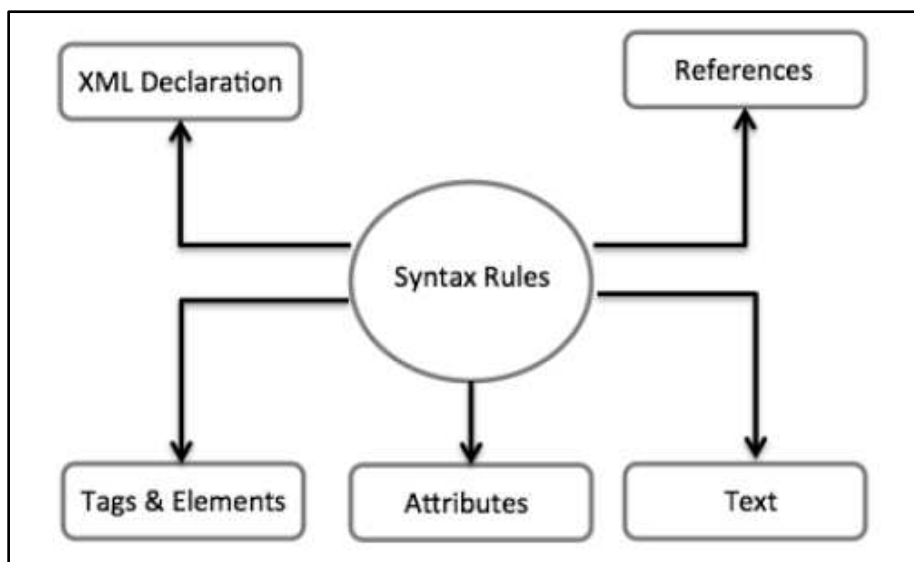
## ۵.۲ نحوه نوشتن کد XML – (XML Syntax)

در این عنوان قواعد ساده نوشتن کد XML در یک فایل مورد مطالعه قرار می‌گیرد. به خاطر وضاحت بیش‌تر موضوع، یک کد ساده ولی مکمل XML در ذیل آورده شده است.

```
<? xml version="1.0"?>
<contact-info>
<name>Farid Ahmad</name>
<company>Software House</company>
<phone> (0093) 777-456789</phone>
</contact-info>
```

اگر به کد بالا دقت شود در آن دو نوع معلومات دیده می‌شود:

- مارکاپ، مانند <contact-info>...</contact-info>
  - متن یا دیتای متنی، مانند فرید احمد، Software House و شماره تلفون.
- شکل پایین دیاگرامی را نشان می‌دهد که در آن قواعد نوشتن انواع مختلف Markup ها و متن در یک فایل XML نشان داده شده است. ناگفته نباید گذاشت که عناصر نشان داده شده در شکل زیر، با ترجمه و کمی تفصیل توضیح داده می‌شود.



شکل ۵-۱ قواعد نوشتن انواع مختلف مارکاپها و متن در فایل های XML.

### ۵.۲.۱ اعلامیه XML – (XML Declaration)

یک فایل XML به صورت اختیاری می تواند این بخش را داشته باشد و یا هم نداشته باشد. قسمتی که به نام اعلامیه XML یاد می شود، منظم بودن کد را نشان داده و نیز ورژن کودنگ، فارمت کودنگ و معرفه دیگری به نام حالت Standalone در آن نشان داده می شوند. معلومات ورژن و فارمت کودنگ اگر در قسمت اعلامیه یا Declaration ذکر هم نگردد، این اختیاریها به صورت خودکار توسط اپلیکیشن استفاده می شوند. اعلامیه کد XML طوری که در نمونه کد بالا نیز قسماً دیده شد، طور ذیل بوده و اولین سطر کد را تشکیل می دهد.

```
<?xml version="1.0" encoding="UTF3- 8 "?>
```

در سطر بالا ورژن همان نمونه کد XML را نشان می دهد که در این جا ورژن آن یک است و اختیار دومی (encoding) فارمت کودگذاری متن را نشان می دهد.

### ۵.۲.۲ قواعد نحوی برای نوشتن اعلامیه XML

- کوچک بودن و بزرگ بودن حروف استفاده شده در سطر اعلامیه XML مهم است. اعلامیه XML باید با کد <?xml> طوری آغاز شود که xml به حروف کوچک نوشته شده باشد.
- نامها در اعلامیه XML همیشه به حروف کوچک نوشته می شوند.
- در صورتی که جمله اختیاری اعلامیه XML در یک فایل XML استفاده شده باشد، این سطر باید در یک کد (در ابتدای فایل XML) نوشته شود.

<sup>3</sup> UTF اختصار کلمات Unicode Transformatoin Format بوده و به خاطر کودگذاری دیتا در کمپیوتر استفاده می شود.

- ترتیب نوشتن پارامترها در سطر اعلامیه مهم است. از طرف چپ اول باید ورژن، بعد از آن کودگذاری و بعدتر نیز پارامتر مستقل گذاشته شود؛ یعنی

(The correct order is: version, encoding and standalone.)

- در نوشتن محتواها از قوس ناخنک دوتایی (») و یا هم از قوس ناخنک یک‌تایی (') استفاده شود.
  - اعلامیه XML تگ بسته‌شدن ندارد؛ یعنی صرف یک تگ در آن استفاده می‌شود.
  - یک پروتوکول HTTP که معمولاً در محیط خارج از فایل XML تنظیم شده است، می‌تواند اعلامیه استفاده شده XML را تغییر داده و مطابق طرزالعمل تعبیه شده خود، فایل XML را به اجراء گذارد؛ یعنی صلاحیت پروتوکول بیرونی اولویت داشته و بالای کد XML تأثیرگذار است.
- چند مثال از اعلامیه XML به اساس توضیحات بالا در ذیل نشان داده شده‌اند.

اعلامیه XML بدون پارامترها:

```
<?xml >
```

اعلامیه XML با تعریف ورژن:

```
<?xml version="1.0">
```

اعلامیه XML با تعریف تمام پارامترها:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

اعلامیه XML با تعریف تمام پارامترها با استفاده از قوس ناخنک‌های یک‌تایی برای محتواها:

```
<?xml version='1.0' encoding='iso-8859-1' standalone='no'?>
```

## ۵.۲.۳ تگ‌ها و عناصر XML

یک فایل XML تشکیل شده از چندین عنصر مختلف یا اجزای مختلف XML است. این عناصر به نام نودهای XML (XML Nodes) و یا هم به نام تگ‌های XML یاد می‌شوند (در این کتاب نام تگ‌ها استفاده شده است). سمبولی که به خاطر نشان کرده عناصر XML استفاده می‌شوند، به شکل قوس‌های سه ضلعی یعنی < و > است. نمونه تگ‌ها که در مثال بالا نیز استفاده شده است، طور ذیل است.

```
<element>
```

تگ‌ها بر علاوه بخش‌های اساسی یک فایل XML، به خاطر داخل کردن تبصره‌ها، تعریف تنظیمات ضروری به خاطر تجزیه محیط و به خاطر داخل کردن توصیه‌های خاص نیز به کار برده می‌شوند. تگ‌های XML به انواع ذیل تقسیم‌بندی می‌شوند.

- تگ آغاز (Start Tag): آغاز هر عنصر غير خالى XML توسط تگ آغاز علامت‌گذاری می‌شود. يك مثال خالى از تگ آغاز به نام آدرس طور ذيل است:

<address> ...

- تگ انجام (End Tag): هر عنصری که با تگ آغاز شروع شده باشد یعنی تگ آغاز داشته باشد، باید با تگ انجام بسته شود. مثالی از تگ انجام به ادامه مثال قبلی (قبلاً باز شده است) طور ذیل است:

... </address>

تفاوت تگ انجام با تگ آغاز در اضافه شدن سمبول (/) قبل از نام تگ است. لازم به یادآوری است که تگ آغاز و تگ انجام باید عین کلمه باشد.

- تگ خالی (Empty Tag): متنی که در بین تگ آغاز و تگ انجام گنجانیده می‌شود به نام محتوای تگ یاد می‌شود. يك عنصری که هیچ محتوا ندارد به نام تگ خالی یاد می‌شود. تگ خالی به دو طریقه، نشان داده می‌شود:

a. يك تگ آغاز متعاقباً توسط تگ انجام بسته می‌شود؛ یعنی

<hr></hr>

b. يك تگ کاملاً خالی و بی جوره استفاده می‌شود؛ یعنی

<hr />

دو مثال تگ خالی که ذکر شد هر دو معادل‌اند. تگ‌های خالی برای عناصری که محتوا ندارند استفاده شده می‌توانند.

## ۵.۲.۴ قواعد نحوی برای تگ‌ها و عناصر XML

- نحوه نوشتن تگ و یا نود XML: هر تگ و یا نود XML زمانی که استفاده می‌شود (باز می‌شود) دوباره باید بسته شود. بسته کردن يك تگ حتمی بوده ولی تسلسل آن شرط نیست؛ یعنی يك تگ می‌تواند تگ‌های دیگری در بین خود داشته باشد، ولی در نهایت تگ باز شده باید دوباره بسته شود. شکل عمومی باز کردن و بسته کردن تگ‌ها در XML طور ذیل است:

<element>...</element>

و یا در موارد ساده به این شکل هم قبول می‌شود:

<element>...<element/>

- تودرتو ساختن (Nesting) عناصر: عناصر XML زمانی استفاده شده می‌توانند که اول تگ به طور مثال، باز می‌شود، قبل از بستن آن تگ و یا تگ‌های باز شده دیگری، دیتا داشته باشد. این یکی از کارهای معمول بلکه ضروری در نوشتن کد XML است. در هم‌چو موارد شرط مهم جلوگیری از



تداخل کاری بین عناصر است؛ یعنی یک تگ آخری (بسته کردن) باید عین نام را با آخرین تگ باز شده با همان نام داشته باشد.  
در مثال پایین تگ‌های Nested به شکل نادرست استفاده شده‌اند.

```
<?xml version="1.0"?>
<contact-info>
  <company>Software House
</contact-info>
</company>
```

ولی در این مثال دیگر تگ‌های Nested به شکل درست آن استفاده شده‌اند.

```
<?xml version="1.0"?>
<contact-info>
  <company>Software House</company>
</contact-info>
```

- عنصر ریشه (Root Element): یک فایل XML می‌تواند تنها یک عنصر ریشه داشته باشد. در کد پایین به اساس این موضوع مشکل وجود دارد؛ و آن به خاطری که هر دو عنصر x و y در سطح بالا (Top Level) بدون استفاده از عنصر ریشه قرار گرفته است.

```
<x>...</x>
<y>...</y>
```

در مثال بعدی این مشکل رفع گردیده و یک عنصر ریشه در سطح بالاتر از عناصر x و y قرار گرفته است.

```
<root>
  <x>...</x>
  <y>...</y>
</root>
```

- مسئله کوچک بودن و بزرگ بودن حروف: استفاده از حروف کوچک و یا بزرگ در نام‌های عناصر در XML باید مراعات شود؛ یعنی نام عناصر مانند اعلامیه XML تابع مسئله Case-Sensitivity می‌باشند. به عین شکل نام‌های تگ شروع و تگ ختم هم از نظر املا و هم از نظر کوچک بودن و

بزرگ بودن حروف باید یکی باشند. طور مثال تگ‌های <contact-infor> و <Contact-Info> با هم تفاوت دارند.

## ۵.۲.۵ مشخصه‌های XML – (XML Attributes)

یک مشخصه، خصوصیت یک عنصر XML را طوری تعیین می‌کند که در آن یک نام و یک محتوا برایش داده می‌شود. یک تگ در XML می‌تواند یک یا بیش‌تر از یک مشخصه داشته باشد. طور مثال

```
<a href=http://www.softwarehouse.com/>Software Hourse!</a>
```

در مثال بالا href عبارت از نام مشخصه بوده و <http://www.softwarehouse.com/> محتوای مشخصه را نشان می‌دهد.

## ۵.۲.۶ قواعد نحوی برای مشخصه‌های XML

- استفاده از حروف کوچک و بزرگ در نام مشخصه‌های XML مهم است. این امر در HTML مهم نبوده سهولت استفاده از حروف خورد و بزرگ بدون تغییر مفهوم موجود است. در XML مشخصه‌یی به نام href با مشخصه‌یی به نام HREF فرق می‌کند.
- عین مشخصه نمی‌تواند بیش‌تر از یک محتوا در یک طرزالعمل داشته باشد. به خاطر تعریف دو محتوا برای یک مشخصه به نام b، مثال ذیل درست نیست.

```
<a b="x" c="y" b="z">...</a>
```

- نام‌های مشخصه‌ها بدون قوس ناخنک نوشته شده ولی نام محتواها همیشه باید داخل قوس‌های ناخنک (Quotation Marks) نوشته شوند. مثال پایین یک متن نادرست XML را نشان می‌دهد.

```
<a b=x>...</a>
```

## ۵.۲.۷ رفرنس‌های XML – (XML References)

ریفرنس‌ها اکثراً اجازه اضافه کردن و شامل ساختن متن اضافی و یا مارکاپ اضافی را به فایل XML می‌دهد. رفرنس‌ها همیشه با سمبول & آغاز می‌شوند و با سمبول سیمیکولن (;) ختم می‌شود. رفرنس‌های XML به دو نوع است:

- رفرنس‌های Entity: یک رفرنس Entity دارای یک نام در بین آغاز و انجام قسمت‌های حائل است. طور مثال در &amp; کلمه amp عبارت از نام رفرنس Entity است. نام مذکور به یک رشته از قبل تعریف‌شده متن و یا یک مارکاپ (Markup) اشاره می‌کند.
- رفرنس‌های کرکتر: در این نوع رفرنس‌ها اشاره به کود اسکی یک کرکتر یا حرف می‌شود. طور مثال &#65; رفرنس به حرف (کرکتر) A است.

## متن XML

نوشتن نام‌های عناصر XML و نیز مشخصه‌های XML به اساس حروف کوچک و بزرگ مهم است. به خاطر جلوگیری از مشکلات در قسمت کودگذاری حروف، تمام فایل‌های XML باید به فارمت یونیکود UTF<sup>4</sup>-8 و یا UTF-16 ذخیره شوند. کرکترهای جاهای خالی، مانند خانه‌های خالی، تب‌ها (Tabs) و فاصله خط‌ها (Line Breaks) بین عناصر XML و بین مشخصه‌های XML نادیده (Ignore) گرفته می‌شوند. بعضی از کرکترها توسط XML ریزرف شده و به صورت مستقیم استفاده شده نمی‌توانند. به خاطر سهولت کار، بعضی اصطلاحات معادل به عوض کرکترهای ریزرفی قابل استفاده است. این کودها در جدول پایین فهرست شده‌اند.

جدول ۵-۱ اصطلاحات معادل کرکترهای ریزرف‌شده (غیرمجاز) در XML

کرکترهای ریزرف‌شده در XML	اصطلاح جانشین کرکتر ریزرف‌شده	توضیحات
<	&lt;	کوچک‌تر از (Less Than)
>	&gt;	بزرگ‌تر از (Greater Than)
&	&amp;	امپرسند (Ampersand)
'	&apos;	اپاسترافی (Apostrophe)
«	&quot;	قوس ناخنک (Quotation Mark)

## ۵.۳ سندهای XML – (XML Documents)

یک سند XML عبارت از یک واحد معلومات XML بوده که شامل عناصر و Markup های دیگر در یک بسته ترتیب شده است. یک سند XML می‌تواند انواع مختلف دیتا را در خود داشته باشد. به طور مثال یک دیتابییسی از ارقام می‌تواند در سند XML ذخیره شود، طوری که ارقام ساختمان مالیکولی و یا معادلات ریاضی را به نمایش بگذارند.

---

<sup>4</sup> UTF اختصار کلمات Unicode Transformatoin Format بوده و به خاطر کود گذاری دیتا در کمپیوتر استفاده می‌شود. نمبرهای 8 و 16 عبارت از سائز بیت‌ها در آن‌ها می‌باشد. نوع 8 بیتی 2 به توان 8 کرکتر و نوع دومی آن 2 به توان 16 کرکتر را استفاده می‌کند.

در جدول پایین، مثالی از یک سند ساده XML با توضیح بخش‌های آن نشان داده شده است.

جدول ۵-۲ مثال یک سند ساده XML و توضیح در مورد بخش‌های آن

توضیح	کود XML
قسمت اعلامیه XML که به نام پرلوگ برنامه نیز یاد می‌شود	<?xml version="1.0"?>
عناصر سند	<pre> &lt;contact-info&gt; &lt;name&gt;Farid Ahmad&lt;/name&gt; &lt;company&gt;Software House&lt;/company&gt; &lt;phone&gt;(0093) 777- 456789&lt;/phone&gt; &lt;/contact-info&gt; </pre>

### قسمت پرلوگ سند XML

پرلوگ (Prolog) سند در قسمت بالایی قبل از محتویات و عنصر اصلی روی صفحه می‌آید. در پرلوگ دو نوع معلومات گنجانیده می‌شوند.

- اعلامیه XML (XML Declaration) ... قبلاً توضیح شده است.
- اعلامیه نوع سند (Document Type Declaration)

### قسمت عناصر سند XML

عناصر سند XML قسمت اصلی فایل را تشکیل می‌دهد. در این قسمت سند به ترتیب به سکشن‌هایی تقسیم می‌شود که هر سکشن به خاطر یک هدف خاص در نظر گرفته می‌شود. یک سند XML به چندین سکشن مختلف تقسیم شده می‌تواند. تقسیم‌بندی یک سند به چندین بخش سهولت در قسمت استفاده مؤثر از آن را مهیا ساخته و نیز به Search-Engine ها زمینه پیدا کردن معلومات را با سرعت بالا میسر می‌سازد. عناصر می‌توانند ساختمان‌هایی باشند که در آن‌ها متن و عناصر دیگر تنظیم شده و قابل دسترس باشند.

## ۵.۴ متن (Text)

در دیتابیس‌ها انواع مختلف دیتا به خاطر ذخیره و استفاده متن (Text) موجود بوده و در موارد مختلف استفاده می‌شوند. دسته‌یی از انواع دیتا (Data Types) در دیتابیس که به خاطر تنظیم دیتای متنی به کار برده می‌شود عبارت‌اند از:

CHAR, VARCHAR, BINARY, VARBINARY, BLOB, TEXT, ENUM, SET ...

در این بحث روی طریقه کارکردن انواع دیتای متنی و استفاده از این نوع دومین (Domain) در کیوری‌های دیتابیس‌ها توضیحات و مثال‌های لازم ارائه می‌شوند.

### انواع دیتای متنی CHAR و VARCHAR

نوع دیتای CHAR با نوع دیتای VARCHAR مشابه است؛ یعنی در زمان تعریف هر کدام از این نوع دیتا، متن در جدول و ستون معین، ذخیره شده می‌تواند. تفاوت هر دو در طریقه ذخیره‌شدن دیتا در آن‌ها و استفاده دوباره از دیتای ذخیره شده است. تفاوت دیگر در این دو نوع (CHAR و VARCHAR) در طول اعظمی و در خالیگه‌هایی که به تعقیب محتواها اضافه می‌شوند، است. این دو نوع دیتا زمانی که به خاطر ذخیره دیتا، متنی در یک جدول دیتابیس تعریف می‌شود، طول اعظمی نیز برای‌شان تعیین می‌شود. طور مثال CHAR(30) یک نوع دیتا کرکتر بوده و عناصر دیتا را الی طول 30 کرکتر می‌تواند ذخیره نماید و مورد پروسس قرار دهد. طول نوع CHAR به صورت ثابت در نظر گرفته شده و مورد پروسس قرار می‌گیرد و آن عبارت از همان مقداری است که در زمان ایجاد جدول دیتابیس توسط استفاده‌کننده تعیین می‌شود. طول CHAR می‌تواند یکی از 256 قیمت را بگیرد؛ یعنی از صفر الی 255 تنها یک قیمت را گرفته می‌تواند. در نوع CHAR طول اعظمی یک فیلد در یک ریکورد به اندازه یک بایت معادل 8 بیت است. کمترین طول محتوای قابل ذخیره در چنین فیلدی مساوی به صفر و بیش‌ترین آن 2 به توان 8 منهای یک بوده که همان عدد 255 را می‌دهد.

متن‌های که در فیلدهای نوع VARCHAR در یک جدول دیتابیس ذخیره می‌شوند، می‌توانند دارای طول‌های متغیر (Variable) باشند. طول این نوع دیتا بین صفر الی 65535 کرکتر بوده می‌تواند؛ یعنی یک متن با حد اقل طول صفر و حد اکثر طول 2 بایت (2 به توان 16 بیت منهای یک) در خانه ستون مورد نظر ذخیره شود. یکی از تفاوت‌ها بین انواع دیتای CHAR با VARCHAR در قسمت طول محتوای متن قابل ذخیره در جدول است. طوری که گفته شد، در فارمت CHAR طول اعظمی 255 کرکتر و در فارمت VARCHAR به تناسب نوع CHAR یک قیمت بسیار بالا یعنی بیش از 65000 کرکتر دیتای متنی قابل ذخیره و مدیریت است. تفاوت دیگر در زمان استفاده از این نوع دیتاها، به خاطر ذخیره دیتای متنی در قسمت طریقه دسترسی و مقدار ساحه‌یی که روی دیسک ذخیره اطلاعات توسط هر دو نوع دیتا استفاده می‌شود، است. در فارمت CHAR زمانی که یک طول برای یک فیلد تعیین شد، به همان تعداد کرکترها برای فیلد متذکره در هاردیسک ریزرف می‌شوند. طور مثال فیلد الف در یک جدول با نوع دیتای CHAR(30) مشخص شده است.

در فیلد الف جدول متذکره برای هر ریکورد 30 موقعیت ریزرف شده و امکان بالارفتن سایز جدول در روی دیسک است. در مقابل اگر برای همین فیلد نوع دیتای VARCHAR(30) تعریف شده باشد، سایز جدول در روی دیسک بالا نرفته، یعنی 30 کرکتر ثابت در روی دیسک ریزرف نمی‌شود. بلکه، هر زمانی که دیتا به فیلد یاد شده ذخیره می‌شود، در همان زمان کرکترها در دیسک ذخیره گردیده و سایز جدول بالا در محدوده اعظمی 30 کرکتر برای هر محتوا بالا می‌رود. نوع دیتای CHAR سایز ثابت محتوای فیلد را در نظر می‌گیرد ولی نوع دیتای VARCHAR در طول دیتا تعریف شده، یک کرکتر اضافه‌تر را در دیسک احتواء می‌کند. این تفاوت در جدول پایین نشان داده شده است. جدول پایین تفاوت‌ها بین انواع دیتای CHAR و VARCHAR را با نتایج حاصله با مثال‌هایی در متن وارد شده به دیتابیس نشان می‌دهد.

جدول ۵-۳ تفاوت‌های عمده بین انواع دیتا CHAR و VARCHAR در دیتابیس‌های سیکویل

محتوا	CHAR(4)	ساحه روی دیسک	VARCHAR(4)	ساحه روی دیسک
' '	' '	4 حرف	' '	1 حرف
'ab'	'ab '	4 حرف	'ab'	3 حرف
'abcd'	'abcd'	4 حرف	'abcd'	5 حرف
'abcdedfgh'	'abcd'	4 حرف	'abcd'	5 حرف

اگر یک محتوا در فیلدی با نوع دیتای CHAR(4) و همان محتوا در فیلد دیگری با نوع دیتای VARCHAR(4) ذخیره شده باشد، دیتاهای نشان داده شده از هر دو فیلد همیشه یک چیز نیست. این تفاوت به خاطر این است که فارمت CHAR خانه‌خالی‌هایی را پاک می‌کند که در اخیر محتواها باشند. مثال شکل پایین این مسئله را واضح می‌سازد.

```

C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin\mysql.exe
mysql> use school2
Database changed
mysql> create table tb1 (v varchar(4), c char(4));
Query OK, 0 rows affected (0.22 sec)

mysql> insert into tb1 values ('ab ', 'ab ');
Query OK, 1 row affected (0.08 sec)

mysql> select * from tb1;
+----+----+
| v  | c  |
+----+----+
| ab | ab |
+----+----+
1 row in set (0.00 sec)

mysql> select concat('<'>', v, '<'>'), concat('<'>', c, '<'>') from tb1;
+-----+-----+
| concat('<'>', v, '<'>') | concat('<'>', c, '<'>') |
+-----+-----+
| <ab > | <ab> |
+-----+-----+
1 row in set (0.08 sec)

mysql>
  
```

شکل ۵-۲ تطبیق و استفاده از انواع دیتای CHAR و VARCHAR در قسمت دور کردن خانه‌های خالی اخیر محتواها توسط نوع دیتای CHAR

طوری که در شکل بالا دیده می‌شود، چند طرز العمل سیکویل به کار رفته که طی آن‌ها در دیتابیس مکتب 2، یک جدول به نام tb1 با دو فیلد از انواع CHAR و VARCHAR هر کدام با طول‌های 4 کرکتر ایجاد شده است. بعد از ایجاد مؤفقانه جدول، یک قسم دیتا به شکل 'ab' یعنی دو حرف با دو خانه خالی به هر دو فیلد آن علاوه شده است. یک بار به شکل عادی کیوری شده که در آن‌ها تفاوتی احساس نمی‌شود. بعدتر به کمک تابع Concatenate و اضافه قوس‌ها دوباره کیوری صورت گرفته است. در نتیجه کیوری دوم به طور واضح دیده می‌شود که محتوای CHAR خانه‌های خالی بعد از حروف را حذف کرده ولی فیلد VARCHAR آن‌ها را نشان می‌دهد. این مثال به خاطر دیدن تفاوت کاری انواع دیتای CHAR و VARCHAR که هر دو به خاطر ذخیره متن استفاده می‌شوند، بیان گردید.

#### ۵.۴.۱ انواع دیتای متنی BINARY و VARBINARY

انواع دیتای BINARY و VARBINARY شباهت زیاد به انواع CHAR و VARCHAR دارند. تفاوت عمده‌شان در استفاده از کرکترها است. در انواع دیتای باینری از سلسله‌های باینری استفاده شده، در حالی که این امر در مورد انواع دیتای کرکتر صدق نمی‌کند. به این معنی که در انواع دیتای باینری محتواهایی که در فیلدها ذخیره می‌شوند به شکل ارقام (کود کرکترها) بوده و خود کرکترها به صورت مستقیم در آن‌ها ذخیره نمی‌شوند. طول فیلدهای باینری نیز مانند فیلدهای کرکتر بوده با این تفاوت که در فیلدهای باینری به همان تعداد بایت‌ها قابل ذخیره شدن است؛ یعنی فیلد BINARY بین صفر الی 255 بایت را ذخیره می‌کند و در فیلد VARBINARY بازهم حد اقل صفر و حد اکثر به تعداد 65535 بایت قابل ذخیره شدن است.

#### ۵.۴.۲ انواع دیتای متنی BLOB و TEXT

اصطلاح BLOB اختصار کلمات Binary Large Object بوده و از جمله فارمتهایی است که به خاطر ذخیره کردن مقدارهای بزرگ دیتای متنی در دیتابیس‌ها استفاده می‌شود. نوع دیتا BLOB به چهار کتگوری تقسیم شده و عبارت‌اند از:

- TINYBLOB
- BLOB
- MEDIUMBLOB
- LONGBLOB

تفاوت بین این دسته‌ها برمی‌گردد به طول دیتای متنی که در آن‌ها ذخیره می‌شود. به عین ترتیب نوع دیتا TEXT نیز به منظور مشابه، استفاده شده و این نوع دیتا به چهار کتگوری مشابه طور ذیل تقسیم می‌شود:

- TINYTEXT
- TEXT
- MEDIUMTEXT
- LONGTEXT

انواع دیتای متنی از نوع TEXT با انواع دیتای متنی از نوع BLOB با هم‌دیگر هم‌خوانی دارند؛ یعنی چهار صفت هر دو نوع با هم مشابه بوده طول رشته‌های دیتای متنی قابل ذخیره در آن‌ها نیز مشابه‌اند. محتوای BLOB مانند رشته‌های باینری در نظر گرفته می‌شوند. در نوع دیتای BLOB مقایسه و ترتیب محتواها به خاطر میخانیکیت کار به اساس نمبرها صورت می‌گیرد. در مقابل محتواهای نوع دیتای TEXT به شکل غیر باینری یعنی رشته‌های کرکترها در نظر گرفته می‌شوند. عملیه‌هایی مانند مقایسه کردن، ترتیب و غیره به اساس حروف و سمبول‌های استفاده‌شده (کرکترها) صورت می‌گیرد.

با صرف نظر از تفاوت‌های تخنیکی، یک فیلد با نوع دیتای BLOB مانند ستون VARBINARY استفاده شده می‌تواند و عین مورد مشابهت کاری در بین نوع دیتای TEXT و نوع دیتای VARCHAR نیز صدق می‌کند. تفاوت بین انواع دیتای متنی BLOB و TEXT و انواع دیتای VARBINARY و VARCHAR در موارد ذیل است:

- از اندکس‌هایی که برای انواع دیتای BLOB و TEXT استفاده می‌شوند، مشخص کردن طول پیشوند اندکس در آن‌ها حتمی است. در حالی که عین مسئله برای انواع دیتای CHAR و VARCHAR اختیاری است؛ یعنی تعیین طول پیشوند اندکس‌ها برای دو نوع اخیر ضرورت به مشخص کردن نداشته، خود سیستم مدیریت دیتابیس این کار را انجام می‌دهد.
- برای ستون‌هایی با نوع دیتای BLOB و یا TEXT اختیار تعیین محتواها از قبل (Default Values) استفاده شده نمی‌تواند.

### ۵.۴.۳ نوع دیتای متنی ENUM

اصطلاح ENUM گرفته شده از کلمه Enumeration به معنی شمارش است. در این نوع دیتا، یک محتوا از یک فهرست محتواهای از قبل آماده شده و به جدول دیتا اضافه می‌شود. فهرست محتواهای ممکنه به خاطر استفاده در فیلدی با نوع دیتای ENUM در زمان ایجاد جدول تعریف می‌شود. فواید استفاده از نوع دیتای ENUM عبارت‌اند از:

- سایز کوچک دیتا روی دیسک یکی از مزایای این نوع دیتا است. زمانی که فهرست محتواهای ممکنه بسیار طولانی نباشد و در مقابل تعداد ریکوردهای دیتا در جدولی که ذخیره می‌شوند بسیار باشد، در



هم‌چو حالت، مؤثریت استفاده از نوع دیتای ENUM به وجه احسن هویدا است. زمانی که از نوع دیتای ENUM در یک جدول استفاده شود، رشته‌های دیتا در زمان ثبت به شکل نمبرها (اندکس‌ها) در نظر گرفته شده و تمام طول یک محتوا به طول یک نمبر در دیتابیس معادل می‌شود.

- در کیوری‌هایی که به اساس هم‌چون فیلدها به راه انداخته می‌شوند، محتواهای فیلدهای با نوع دیتای ENUM دوباره به شکل متن قابل خواندن می‌باشند. در زمان اجراء کیوری‌ها، اندکس‌های ذخیره‌شده به حیث محتواهای فیلدهای ENUM دوباره به مفاهیم اصلی‌شان ترجمه‌شده و نشان داده می‌شوند. جهت وضاحت بیش‌تر موضوع، یک مثال استفاده از نوع دیتای ENUM در شکل پایین نشان داده شده است. طرز‌العمل‌های سیکویل که در مثال بالا به کار گرفته شده، به اختصار توضیح می‌گردند.

```

C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin\mysql.exe
mysql> DROP TABLE IF EXISTS shirts;
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE shirts (
  ->     name VARCHAR(40),
  ->     size ENUM('x-small', 'small', 'medium', 'large', 'x-large')
  -> );
Query OK, 0 rows affected (0.08 sec)

mysql> INSERT INTO shirts (name, size) VALUES ('dress shirt','large'),
  -> ('t-shirt','medium'), ('polo shirt','small');
Query OK, 3 rows affected (0.05 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM shirts;
+-----+-----+
| name      | size  |
+-----+-----+
| dress shirt | large |
| t-shirt    | medium |
| polo shirt | small  |
+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT name, size FROM shirts WHERE size = 'medium';
+-----+-----+
| name      | size  |
+-----+-----+
| t-shirt    | medium |
+-----+-----+
1 row in set (0.00 sec)

mysql> UPDATE shirts SET size = 'small' WHERE size = 'large';
Query OK, 1 row affected (0.05 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT * FROM shirts;
+-----+-----+
| name      | size  |
+-----+-----+
| dress shirt | small  |
| t-shirt    | medium |
| polo shirt | small  |
+-----+-----+
3 rows in set (0.00 sec)

mysql> _

```

شکل ۳-۵ استفاده از نوع دیتای ENUM با فهرستی از عناصر قابل ثبت در جدولی به نام پیراهن‌ها

اولین طرز‌العمل عبارت از طرز‌العمل از بین‌بردن جدول (DROP TABLE ...) با در نظر داشت این شرط که طرز‌العمل در صورتی اجراء شود که جدولی به نام پیراهن‌ها موجود باشد. شرط و یا اختیار IF EXISTS کمک

می‌کند تا از اشتباه جلوگیری شود؛ یعنی اگر جدول موجود نباشد، از طرزالعمل بین‌بردن جدول، صرف نظر شود. طرزالعمل بعدی عبارت از ایجاد جدولی به نام پیراهن‌ها است.

در جدول دوم فیلد تعریف شده است که موضوع مورد بحث همان فیلد دومی جدول است. نوع دیتای فیلد دومی ENUM تعریف شده و به خاطر داخل کردن محتواها، نام 5 محتوا در کد ایجاد جدول اضافه شده است. این فهرست محتواها، سیستم مدیریت دیتابیس را کمک می‌کند که به هر محتوا یک نمبر نشانی (اندکس) تخصیص داده شود تا در جریان پروسس دیتا (کیوری کردن) از آن اندکس استفاده شود. بعد از ایجاد مؤفقانه جدول پیراهن‌ها، طرزالعمل بعدی سه ریکورد دیتا را به این جدول داخل می‌کند. در ادامه یک کیوری عمومی (بدون قید شرایط) به راه انداخته شده و محتوای جدول را نشان می‌دهد.

کیوری بعدی با اضافه عمل شرطی به راه انداخته شده است که این کیوری هم به شکل مؤفقانه اجراء شده و نتیجه‌ی را که شامل یک ریکورد می‌شود، نشان می‌دهد. طرزالعمل بعدی به خاطر تغییر آوردن در دیتا به راه انداخته شده و در محتوای فیلد اندازه، سائزهای کلان به سائزهای کوچک تبدیل شده است. محتوای به‌روزشده نیز شامل فهرست محتواهای مجاز برای فیلد اندازه بوده و باعث موفقیت طرزالعمل می‌گردد. داخل کردن یک میلیون ریکورد به جدول پیراهن‌ها با محتوای medium برای فیلد سائز، به یک میلیون بایت حافظه ضرورت خواهد داشت. در صورت استفاده از نوع دیتای VARCHAR به عوض نوع دیتای ENUM، طور مثال، برای فیلد سائز ساحه‌ی که روی دیسک به خاطر ذخیره دیتا است شش برابر یعنی شش میلیون خواهد گردید. به خاطری که در حالت استفاده از نوع دیتای ENUM به خاطر ذخیره محتوای هر ریکورد در این فیلد یک بایت در نظر گرفته می‌شود. در حالی که برای به کارگیری انواع دیگر دیتا به اندازه طول‌های تعریف‌شده، از حجم روی دیسک استفاده می‌نماید.

### قیمت‌های اندکس برای فیلدهای نوع ENUM:

هر محتوای تعریف‌شده در فهرست، برای یک فیلد ENUM دارای یک شماره شناسایی (اندکس) است:

- در یک فهرست عناصر قابل داخل شدن به فیلد ENUM حد اکثر به تعداد 65535 عدد می‌باشند.
- اندکس هر عنصر به ترتیبی که در فهرست قرار می‌گیرد از یک به بعد شماره‌گذاری می‌شود.
- شماره اندکس محتواهای خالی در فیلد ENUM عبارت از صفر است؛ یعنی به خاطر پیدا کردن محتواهای نادرست در فیلد نوع ENUM در یک جدول کیوری ذیل استفاده شده می‌تواند:

```
>> SELECT * FROM TableName WHERE ENUMColumn = 0;
```

- اندکس محتوای ناشناخته (Null) نیز عبارت از NULL است.
- اندکسی که به خاطر نشان دادن موقعیت عناصر فهرست محتواهای قابل داخل شدن به فیلد نوع ENUM استفاده می‌گردد، به اندکس‌هایی که در جدول‌ها تنظیم و استفاده می‌شوند ارتباط نداشته، صرف به منظور کار با عناصر فهرست ENUM توسط سیستم مدیریت دیتابیس تنظیم و استفاده می‌شود.

- در صورت استفاده از محتوای نمبر در فیلد نوع ENUM، در عملیه‌های حسابی اندکس، فیلد در نظر گرفته شده و از محتوای اصلی مانند محتوای متنی صرف نظر می‌گردد.  
در مثال بالا که لیستی از عناصر برای فیلدِ سایز در جدول پیراهن‌ها ترتیب شده بود اگر اندکس‌های عناصر فهرست دیده شوند، طور ذیل خواهند بود:

جدول ۴-۵ اندکس‌های فهرست عناصر قابل داخل شدن در فیلد سایز از نوع ENUM جدول پیراهن‌ها

محتوا	اندکس
ناشناخته (NULL)	ناشناخته (NULL)
'x-small'	0
'small'	1
'medium'	2
'large'	3
'x-large'	4
	5

در صورت استفاده یک فیلد نوع ENUM در مسایل عددی، قیمت اندکس محتوای فیلد ENUM توسط سیستم مدیریت دیتابیس گرفته شده و استفاده می‌شود. طور مثال استفاده از توابع SUM() و AVG() بالای فیلدهایی از نوع ENUM با اندکس فیلدها، عملیه را انجام می‌دهند؛ و یا به خاطر علامت‌گذاری شماره اندکس یک عملیه ریاضی، مستقیماً بالای اندکس اجراء شده می‌تواند. به خاطر وضاحت بیش‌تر موضوع، در شکل پایین کیوری‌ها بالای جدول پیراهن‌ها اجراء شده و نتایج مطلوب آن‌ها نشان داده شده‌اند.

```

C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin\mysql>
mysql> SELECT * FROM shirts;
+-----+-----+
| name      | size |
+-----+-----+
| dress shirt | large
| t-shirt    | medium
| polo shirt  | small
+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT size, size+0 AS 'Index of size' FROM shirts;
+-----+-----+
| size      | Index of size |
+-----+-----+
| large     | 4
| medium    | 3
| small     | 2
+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT SUM(size), AVG(size) FROM shirts;
+-----+-----+
| SUM(size) | AVG(size) |
+-----+-----+
| 9         | 3         |
+-----+-----+
1 row in set (0.00 sec)

mysql>

```

شکل ۴-۵ کاربرد عملی‌های سیکویل به خاطر نشان‌دهنده استفاده از اندکس از نوع دیتا ENUM

طوری که در شکل بالا و مثال‌های اجرا شده دیده می‌شود، فیلد سایز در جدول پیراهن‌ها از نوع ENUM بوده؛ و در طرزالعمل وسطی شماره اندکس، عناصر فهرست نشان داده می‌شود؛ یعنی عنصر small در شماره 2، عنصر medium در شماره 3 و عنصر large در شماره 4 فهرست عناصر نشان داده شده است. اگر به کد ایجاد جدول و تعریف سایز فیلد نظر اندازی شود، همین ترتیب به وضوح مشاهده می‌شود.

CREATE TABLE shirts

name VARCHAR(40),

size ENUM('x-small', 'small', 'medium', 'large', 'x-large')

;

در طرزالعمل آخری شکل بالا دو تابع عددی بالای فیلد سایز، اجرا شده است. اگر دقت شود تابع SUM() مجموعه اندکس تمام ریکوردهای جدول را در فیلد سایز جمع کرده و نتیجه 4 + 3 + 2 یعنی 9 را نشان می‌دهد؛ و تابع AVG() نیز مجموعه اندکس‌ها را تقسیم تعداد ریکوردهای جدول نموده و در نتیجه 3 را نشان می‌دهد.

## ۵.۵ مولتی‌مدیا (Multimedia)

مولتی‌مدیا یک اصطلاح عام و تقریباً جدید در ساحة دیتای دیجیتل و کمپیوترساینس بوده و به معنای مجموعه‌یی از متن، تصویر گرافیکی، تصویر متحرک (Animation)، صدا و ویدیو است. مجموعه هر عنصر مولتی‌مدیا، عبارت از دیتای مربوط به مولتی‌مدیا است؛ یعنی یک تصویر عبارت از دیتای مولتی‌مدیا است. یک فایل صدا (Audio) عبارت از دیتای مولتی‌مدیا است. یک فایل ویدیو عبارت از دیتای مولتی‌مدیا است. یک انیمیشن عبارت از دیتای مولتی‌مدیا است. یک متن به شکل یک فایل عبارت از دیتای مولتی‌مدیا است؛ و زمانی که تعدادی از دیتای مولتی‌مدیا با هم ترکیب شوند یک پروژه مولتی‌مدیا را تشکیل می‌دهند. مثال‌های پروژه‌های مولتی‌مدیا بی‌شمارند. به طور مثال یک گزارش ویدیویی می‌تواند شامل تصویرهای متحرک، صدا و متن روی تصاویر را شامل شود. تولیدات مولتی‌مدیا ضرورت به کارهای ایجادکننده، مهارت‌های تخیلی و نرم‌افزارهای تخصصی دارند. کارهای ایجادکننده یا تفکر خلاق برای کسانی که در بخش تولید مولتی‌مدیا کار می‌کنند یکی از رمزهای کامیابی در کارشان به شمار می‌رود. چون نمایش مولتی‌مدیا به خاطر رساندن معلومات به استفاده‌کنندگان است، بناءً کسانی که در این بخش کار می‌نمایند، باید آگاهی کامل از نحوه برقراری ارتباطات با مردم و قابلیت‌های گرفتن معلومات توسط انسان را به خوبی بلد باشند. انسان‌ها با استفاده از حواس پنج‌گانه‌شان به حیث ترمینل‌های اصلی، معلومات را از محیط می‌گیرند. حواس پنج‌گانه انسان‌ها از جمله قابلیت‌هایی اساسی در برقراری ارتباطات با محیط به شمار رفته و عبارت‌اند از: حس بینایی، حس شنوایی، حس بویایی، حس لامسه و حس احساس درد. مولتی‌مدیا باید طوری دیزاین شود تا برای استفاده‌کنندگان (انسان‌ها) قابل درک و جالب باشد. بناءً دانستن میخانیکیت‌های ارتباطات انسان‌ها و گرفتن دیتا از یک پروژه مولتی‌مدیا ضرورت به داشتن مهارت‌های ایجادکننده و خلاق توسط اشخاص مسلکی این بخش دارد.

مهارت‌های تخیلی اشخاصی که با انکشاف پروژه‌های مولتی‌مدیا کار می‌نمایند، یکی دیگر از نکات کلیدی به حساب می‌آید. به خاطر ساختن یک پرزنتیشن مناسب مولتی‌مدیا، استفاده‌کنندگان باید مهارت‌های کافی در قسمت کار با فارمت‌های دیتای مربوطه را داشته باشند. به خاطر تنظیم تصویرها، به گونه مثال، اشخاص مسلکی باید با فارمت‌های مختلف تصویرها بلدیت داشته باشند. برای نشان دادن مناظر طبیعی، فارمت‌های خاص از آدرس مولتی‌مدیا در ساحة دیتای دیجیتل موجوداند. برای نشان دادن تصاویر غیرطبیعی با قطعات بزرگ مثل یک رنگ در تصویر مانند کارتونها، فارمت‌های خاصی موجوداند. برای نمایش ویدیوکلپ‌های کوتاه، ویدیوهای طولانی و یا ویدیوهای که به اساس فریم‌ها ایجاد می‌شوند، فارمت‌های متفاوت تعریف شده است. استفاده از فارمت مناسب برای نمایش دیتای مولتی‌مدیا ارتباط مستقیم به کیفیت، سرعت و ساینز دیتا در روی دیسک دارد. به هر اندازه‌یی که فارمت مناسب برای دیتای مولتی‌مدیا انتخاب شود به همان اندازه کیفیت کار بالا بوده، پروسس دیتا توسط کمپیوتر با سرعت و دقت انجام شده و در نهایت استفاده‌کنندگان راضی می‌باشند.

ضرورت به نرم‌افزارها یا سافت‌ویر تخیلی در ایجاد و تنظیم دیتای مولتی‌مدیا یکی دیگر از مسایل مورد بحث است. به خاطر تنظیم دیتای مولتی‌مدیا، به سافت‌ویر تخیلی ضرورت است. طوری که قبلاً اشاره شد، برای

تصویرها، فارمت‌های مختلف موجود بوده و استفاده از فارمت مناسب یکی از مسایلی است که یک پروژه مولتی‌مدیا را به کامیابی می‌رساند. به خاطر کار با فارمت‌های تصویرها و تبدیل آن به یک تصویر به فارمت مناسب، به سافت‌ویر تخنیکی ضرورت است. به عین شکل، زمانی که با دیتا ویدیو کار می‌شود، سافت‌ویر مربوطه باید در کمپیوتر موجود باشد و انکشاف‌دهنده پروژه مولتی‌مدیا باید مهارت‌های لازم استفاده از سافت‌ویر مورد نظر را داشته باشد. پس وجود خلاقیت در ایجاد دیتای مناسب مولتی‌مدیا، سافت‌ویر تخصصی به خاطر کار با دیتای مولتی‌مدیا و مهارت‌های تخنیکی انکشاف‌دهندگان از رموز کامیابی یک پروژه مولتی‌مدیا به حیث فکتورهای اساسی به حساب می‌آیند.

## ۵.۶ مولتی‌میدیای خطی و غیر خطی (Linear and Nonlinear Multimedia)

طوری که از متن عنوان فهمیده می‌شود، مولتی‌مدیا از نظر نمایش به دو دسته تقسیم‌بندی می‌شود. یکی به نام مولتی‌میدیای خطی و دیگری به نام مولتی‌میدیای غیرخطی مسمما است. این تقسیم‌بندی به خاطر انترکشن (Interaction) بین استفاده‌کننده و وسیله‌ی که مولتی‌مدیا را به نمایش می‌گذارد، است. در صورتی که استفاده‌کننده در نمایش مولتی‌مدیا رُل داشته باشد، آن را به نام مولتی‌مدیا غیر خطی (Nonlinear) یاد می‌کنند. در صورتی که استفاده‌کننده در نمایش دیتای مولتی‌مدیا رُل نداشته باشد، به نام مولتی‌میدیای خطی (Linear) یاد می‌شود.

یک نمایش مولتی‌مدیا ضرور نیست که حتماً باید با استفاده‌کننده یا واضح‌تر گفته شود با بیننده انترکشن مستقیم داشته باشد. در چنین حالتی استفاده‌کننده فقط نشسته و دیتا را تماشا می‌کند و یا آن را می‌شنود. مثال‌های مولتی‌میدیای خطی زیاد بوده از جمله، دیدن تصویرها و یا دیدن و شنیدن نشرات تلویزیون می‌باشد. در زمان دیدن تلویزیون استفاده‌کننده در کنترل نمایش نشرات تلویزیون کدام نقشی نداشته و فقط بیننده است. نمایش خطی به نام مولتی‌میدیای Passive یاد می‌شود.

مولتی‌میدیای غیرخطی عبارت از طریقه‌ی است که در زمان نمایش به استفاده‌کنندگان صلاحیت کنترل مولتی‌مدیا داده شده باشد. به طور مثال دیدن یک ویدیو از طریق کمپیوتر که در آن استفاده‌کننده بتواند در هر نقطه ویدیو را توقف داده و به قسمت بعدی برود و یا هم یک قسمت را بار بار ببیند و غیره؛ و یا تصاویری را به هر شکلی که خواسته باشد دیده بتواند. چنین نوع دیدن دیتای مولتی‌مدیا که در آن استفاده‌کننده در تنظیم نمایش رُل داشته باشد به نام مولتی‌میدیای فعال (Interactive) یاد می‌شود؛ یعنی مولتی‌میدیای غیرخطی به نام مولتی‌میدیای Interactive نیز یاد می‌شود.

در یک فهرست دیتای متنی که به شکل مولتی‌میدیای Interactive دیزاین شده باشد و در آن عناصر متن به ترتیب نشان داده شوند و استفاده‌کننده کنترل نمایش را در دست داشته باشد، به هر پیمانه‌ی که موقعیت‌های بیش‌تر کنترل در اختیار استفاده‌کننده قرار داده شود به همان پیمانه رضایت استفاده‌کننده بیش‌تر می‌شود. معمولاً در یک وب‌سایت مینوهای اساسی به خاطر کنترل محتویات در هر صفحه گذاشته می‌شوند تا از هر صفحه فرعی یک استفاده‌کننده بتواند دوباره به صفحه اصلی و یا بخش‌های دیگر وب‌سایت

برود. همین امر باعث بالارفتن استفاده و رضایت استفاده‌کننده شده و به حیث نمونه‌یی از نمایش مناسب مولتی‌مدیا گفته می‌شود.

### ۵.۶.۱ عناصر مولتی‌مدیا (تصویرها)

تصویرها یکی از قسمت‌های اساسی در دیتای مولتی‌مدیا به شمار می‌روند. یک اصطلاح است که «یک تصویر هزار کلمه را ارائه می‌کند»؛ یعنی با استفاده از تصویر، معلومات به شکل کامل و با استناد به استفاده‌کننده انتقال پیدا می‌کند. هدف از گفتن تصویرها در این بحث، همان تصویرهای دیجیتالی است که در کمپیوترها و وسایل الکترونیکی استفاده می‌شوند. تصویرها به صورت عمومی به دو شکل در کمپیوترها ذخیره و پروسس می‌شوند. یکی به نام تصویرهای نقطه‌یی یاد شده و نقاط یا پیکسل‌ها (Pixels) کوچک‌ترین عناصر تشکیل‌دهنده تصویرها می‌باشند؛ و دیگری به نام تصویرهای وکتوری یاد شده که عناصر تشکیل‌دهنده این دسته عبارت از خط‌ها و قوس‌ها می‌باشند.

دسته دوم به خاطر دیزاین‌های انجینیری و نمایش دیتای سمبول‌ها استفاده شده و خارج از بحث مولتی‌مدیا است. دسته اول یا تصویرهای نقطه‌یی با فرمت‌های مختلف ایجاد و از آدرس مولتی‌مدیا استفاده می‌شوند. تصاویر نقطه‌یی به نام‌های Bitmap و یا هم Raster یاد می‌شوند. سافت‌ویرهای که به خاطر ایدیت کردن تصاویر راستر استفاده می‌شوند، به نام پروگرام‌های رنگ‌آمیزی (Painting) یاد شده و سافت‌ویرهای ایجاد و ایدیت تصاویر نوع وکتور به نام پروگرام‌های رسامی (Drawing) یاد می‌شوند. به خاطر گرفتن جای کم در حافظه، فایل‌های تصویری توسط کمپیوتر به شکل فشرده‌شده (Compressed)، ذخیره و استفاده می‌شوند. در اکثر فرمت‌های این نوع فایل، کار فشرده‌سازی در خود میتود استفاده از فرمت فایل تنظیم شده و استفاده می‌شود. طور مثال فرمت‌های GIF<sup>5</sup>، JPEG<sup>6</sup> و PNG<sup>7</sup> در تصویرهای راستر از جمله فرمت‌هایی اند که فشرده‌سازی بیت‌های آن‌ها توسط الگوریتم‌ها در خود فرمت‌های تصویرها صورت می‌گیرد. سه فرمت تصویرها که ذکر شد و فرمت TIFF<sup>8</sup> از جمله فرمت‌های معمول به خاطر استفاده از تصویرهای راستر در کمپیوترساینس به شمار می‌روند. طوری که گفته شد، تصویرهای راستر متشکل از نقاط (پیکسل‌ها) است. هر پیکسل یک بیت دیتا را تمثیل کرده و یک بیت می‌تواند دو قیمت صفر و یا یک را به خود بگیرد. در صورت استفاده از یک بیت به خاطر نشان‌کرده قیمت یک پیکسل تصویر سیاه و سفید تشکیل می‌گردد. با اضافه بیت‌ها برای هر پیکسل در یک تصویر، کیفیت تصویر بالا رفته و گنجایش نمایش رنگ‌ها در پیکسل‌های تصویر پیدا می‌شود. یک تصویر راستر 2 بیتی دارای 4 رنگ بوده می‌تواند و پیکسل‌های یک تصویر 8 بیتی الی 255 رنگ داشته می‌تواند؛ یعنی زمانی که یک پیکسل در تصویری بتواند 8 بیت یا یک بایت را استفاده کند، قیمت‌های آن پیکسل 2 به توان 8 رنگ را که در واقع همان عدد 256 را نشان می‌دهد، احتواء می‌کند. به همین ترتیب، تصویرهایی

<sup>5</sup> GIF اختصار کلمات Graphics Interchange Format بوده و به خاطر تصویرهای با رنگ‌های کم استفاده می‌شود.

<sup>6</sup> JPEG اختصار کلمات Joint Photographic Experts Group بوده و این فرمت برای تصویرهای با رنگ‌های ملیونی استفاده می‌شود.

<sup>7</sup> PNG اختصار کلمات Portable Network Graphics بوده و تصاویر فرمت PNG معمولاً برای تصویرهای انکشافی استفاده می‌شوند.

<sup>8</sup> TIFF اختصار کلمات Tagged Image File Format بوده و بخاطر تصویرهای باکیفیت بالا استفاده می‌شود.



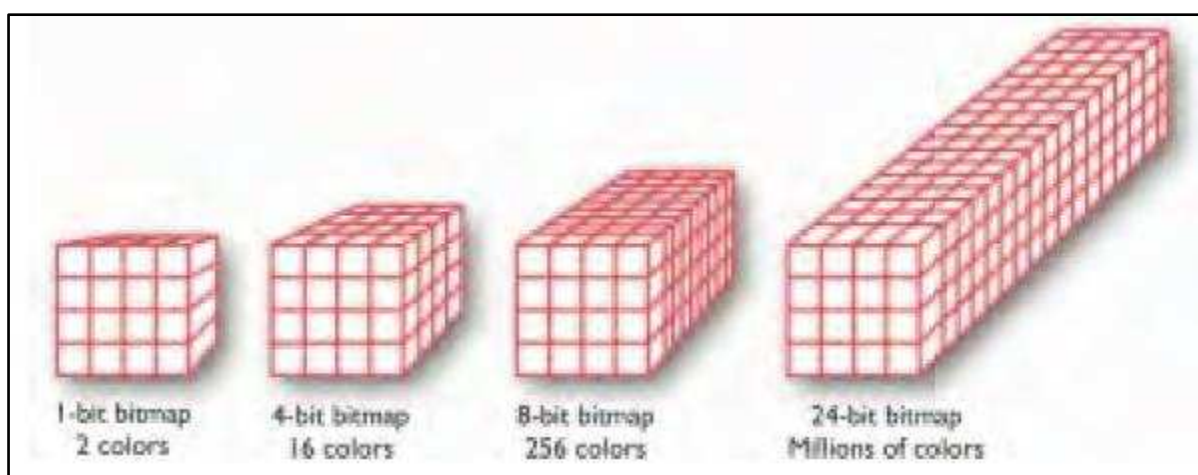
باکیفیت بالا معمولاً 3 بایت را برای رنگ‌های هر پیکسل اجازه می‌دهد که با این اختیار، اضافه از شانزده میلیون (2 به توان 24) رنگ قابل استفاده باشد.

جدول پایین تعداد رنگ‌های قابل استفاده را به اساس تعداد بیت‌ها برای یک پیکسل نشان می‌دهد.

جدول ۵-۵ تعداد رنگ‌ها به تفکیک بیت‌ها در یک پیکسل و ارقام باینری ممکنه آن‌ها

تعداد بیت‌ها	تعداد رنگ‌های ممکنه	تعداد یک‌ها و صفرها (باینری)
1 bit	2	0,1
2 bit	4	00,01,10,11
4 bit	16	0000,0001,0011,0111,1111,0010,0100,1000, 0110,1100,1010,0101,1110,1101,1001,1011
8 bit	256	...
3 Byte یا 24 bit	16777216	...

جهت وضاحت بیش‌تر موضوع، تعداد بیت‌ها (اختیارهای انتخاب رنگ‌ها) برای یک پیکسل در شکل پایین نشان داده شده است.



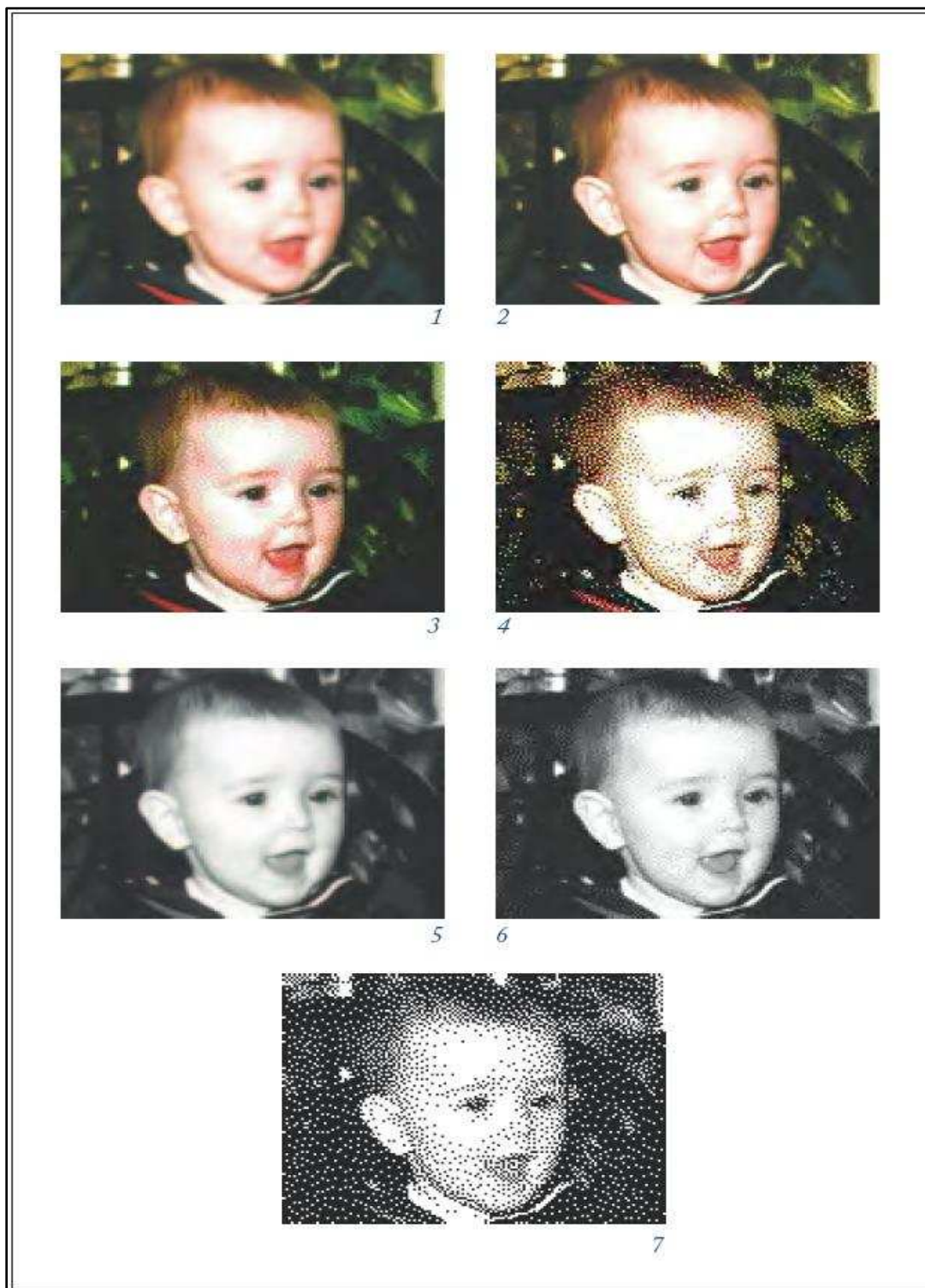
شکل ۵-۵ عمق پیکسل (تعداد بیت‌ها برای هر پیکسل) با ابعاد مختلف ۱ بیت، ۴ بیت، ۸ بیت و ۲۴ بیت الی یک میلیون‌ها رنگ یک پیکسل.

مثالی برای استفاده از فرمت‌های مختلف تصویرها در ذیل در نظر گرفته شده است. در این مثال یک تصویر به هفت فرمت مختلف ارائه شده است. به شکل پایین توجه شود.

- تصویر شماره ۱ در فرمت 24 بیت (بیش‌ترین تعداد رنگ با اضافه شانزده میلیون) تنظیم شده است. طوری که دیده می‌شود کیفیت تصویر با وجود بلندترین تعداد رنگ ضعیف‌تر معلوم می‌شود. بناءً همیشه بلندبودن تعداد رنگ در یک تصویر خوب‌ترین کیفیت را نمی‌داشته باشد.



- تصویر شماره 2 در فارمت 8 بیت (یک بایت با 256 رنگ) تنظیم شده است. این تصویر بالاترین کیفیت را دارد؛ یعنی به هم‌چون تصویر استفاده از 8 بیت بهترین اختیار است. سائز تصویر در روی دیسک مورد نظر به تصویر شماره 1 سه برابر کمتر می‌شود که از مزیت‌های دیگر استفاده از این تعداد رنگ در تصویر است.
- تصویر شماره 3 هم‌چنان به فارمت 8 بیت تنظیم شده است. طوری که دیده می‌شود در این تصویر کیفیت ضعیف‌تر شده است. این به خاطری است که در این فارمت از 256 رنگ عادی استفاده شده است در حالی که در تصویر شماره 2 از 256 رنگ انتخابی کار گرفته شده بود. (به مهارت تخنیکی کسی که فارمت تصویر را تنظیم می‌کند، بستگی دارد).
- تصویر شماره 4 به عمق 4 بیت تنظیم شده است. تعداد رنگ‌ها در این تصویر به 2 به توان 4 یعنی 16 محدود می‌شود. (کیفیت تصویر به ترتیب تنزیل پیدا می‌کند).
- تصویر شماره 5 با فارمت 8 بیت خاکی (Grayscale) یعنی 256 رنگ بین سیاه و سفید تنظیم شده است.
- تصویر شماره 6 با فارمت 4 بیت خاکی (Grayscale) یعنی 16 رنگ بین سیاه و سفید تنظیم شده است.
- و بالاخره تصویر شماره 7 با فارمت 1 بیت با دو حالت سیاه و سفید تنظیم شده است. در این حالت هر پیکسل یا سیاه است و یا هم سفید. قیمت‌های پیکسل‌های که بین 0 و 127 اند به رنگ سفید (و یا هم سیاه) تبدیل شده و قیمت‌های پیکسل‌های که بین 127 و 255 اند به رنگ مقابل آن یعنی سیاه (و یا هم سفید) تغییر یافته است.



شکل ۵-۶ یک تصویر را با تعداد بیت‌های مختلف برای هر پیکسل

توضیحات لازم در مورد شکل بالا در صفحه پیش‌تر آن ارائه شد. تصویرهای ۴ و ۶ در شکل بالا عین سائز را در روی دیسک احتواء می‌کنند. طوری که دیده می‌شود تصویر شماره ۶ نظر به تصویر شماره ۴ به مراتب دارای کیفیت بالاتر است؛ یعنی بعضاً عدم استفاده از رنگ‌های عمومی و استفاده از رنگ‌های خاکی (Grayscale) کیفیت کار را بالا می‌برد.

## ۵.۶.۲ عناصر مولتی‌مدیا (صدا و ویدیو)

استفاده و به کارگیری دیتای دیجیتل مربوط صدا و ویدیو یکی دیگر از مسایل عمده در ساحه کمپیوترساینس به شمار می‌رود. صدا و تصویر متحرک (ویدیو) قسمت قابل ملاحظه‌یی از معلومات را از یک انترفیس کمپیوتر به استفاده‌کننده انتقال می‌دهد؛ مانند انواع دیگر دیتا، صدا و ویدیو نیز قبل از انکشاف کمپیوترها به شکل آنالوگ و یا میخانیکی موجود بود. با به میان آمدن دنیای دیجیتل و انکشاف تکنالوژی‌های کمپیوتر، این انواع دیتا نیز به نحوی وارد این سلسله گردیده است. به خاطر ذخیره و استفاده دوباره از صدا و ویدیو در کمپیوتر، فرمت‌هایی تعریف شده که توسط آن دیتاهای یاد شده به شکل فایل‌های قابل استفاده ثبت می‌شوند. فایل‌های ثبت شده صدا و ویدیو نوعی از دیتا را در مولتی‌مدیا افاده می‌نمایند. ذخیره، تنظیم و در مجموع مدیریت دیتای صوتی و تصویری توسط برنامه‌های مخصوص یکی از ساحات مهم در کمپیوترساینس محسوب می‌شود. فرمت‌های استاندارد به منظور ثبت و استفاده دیتای صوتی و ویدیویی در کمپیوترها تعریف شده و استفاده می‌شوند.

یکی از فرمت‌های عام در قسمت ثبت دیتای صوتی به نام فرمت MP3 یاد می‌شود. این فرمت توسط دسته‌یی از محققان تحت نام MPEG، گرفته شده از کلمات (Moving Picture Experts Group) در دهه ۱۹۹۰ انکشاف داده شد. سیستم صدا (Sound) سیستم عامل ویندوز نیز بر مبنای همین فرمت بوده و به نام WMA گرفته شده از کلمات (Windows Media Audio) انکشاف داده شده است.

فرمت دیگری به نام MP4 یاد می‌شود که هم برای ذخیره فایل‌های صوتی به کار می‌رود و هم به خاطر ذخیره فایل‌های ویدیویی استفاده می‌شود. این فرمت به اساس فرمت mov مربوط Quick Time movie شرکت اپل است. فرمت MP4 زمانی برای یک فایل استفاده می‌شود که در آن دیتای صوتی و تصویری هر دو موجود باشد. اگر تنها دیتای صوتی هدف باشد، برای آن اکستنشن m4a استفاده می‌شود. فرمت‌های زیادی به خاطر استفاده دیتای صوتی و ویدیویی موجود است.

از جمله فرمت‌های موجود برای ذخیره فایل‌های صوتی در ذیل تنها نام گرفته می‌شود:

- Mp3, mp4, wav, aif, aac, flv, mov, swf, wma, ogg, m4a, m4r, aac, midi, mmf, 3g2, 3gp, 3gp2, 3gpp

و فرمت‌های موجود برای ذخیره فایل‌های ویدیویی عبارت‌اند از:

- MPEG (.mp4), ogg, Flash Video (.flv), QuickTime (.mov), Windows Media Player Format (.wmv), Webm (.webm), RealMedia (.rm)



مدل‌های دیتا به خاطر ذخیره و استفادهٔ دیتای دیجیتال در کامپیوترها و دیتابیس‌ها، طریقه‌هایی را معرفی می‌کند که مؤثریت انواع دیتا را بیش‌تر می‌سازد. در این فصل سه عنوان عمومی در مورد انواع دیتای دیجیتال مورد بحث قرار گرفته شد که عبارت از فرمت دیتای XML، نوع دیتای متنی (TEXT) و دیتای مولتی‌مدیا (Multimedia) است؛ و معلومات عمومی در مورد هر سه کتگوری ارائه گردید. طریقهٔ نوشتن کد لسان XML و فرمت فایل‌های XML با مثال‌ها توضیح شد. عناوین فرعی که تحت عنوان نوع دیتای متنی و طریقهٔ استفاده از آن بود نیز توضیح گردید. تفاوت‌های قسمی در انواع دیتای متنی نیز با ذکر مثال‌های لازم به بحث گرفته شدند.

انواع دیتای مولتی‌مدیا و استفاده از این نوع دیتا در کامپیوتر و دیتابیس‌ها توضیح شدند. فرمت‌های شامل مولتی‌مدیا که به خاطر ذخیره و ترتیب دیتا موجوداند، تا اندازه‌یی توضیح گردیدند. عناصر مولتی‌مدیا شامل تصویرها، دیتای صوتی و تصویری تحت عنوان مولتی‌مدیا توضیح شدند. به تصویرها به خاطر ارتباط بیش‌تر موضوع یک اندازه بیش‌تر پرداخته شد. در اخیر بحث دیتای صوتی و تصویری از فرمت‌های موجود در هر دو نوع (فایل‌های صوتی و فایل‌های ویدیویی) فهرست شدند تا شاگردان علاقه‌مند بتوانند در مورد آن به مطالعات بیش‌ترشان ادامه دهند.



## فعالیت های فصل پنجم

۱. نحوه نوشتن تگ‌ها را با ذکر یک مثال در XML بنویسید.
۲. یک کود ساده XML را بنویسید که توسط آن نام شخص، شرکت شخص و شماره تلفن نشان داده شود.
۳. قطعه کود فعالیت قبلی را با سطر اعلامیه XML یا (XML Declaration) تکمیل کنید.
۴. مثالی از یک مشخصه XML (Attribute) را بنویسید که در آن آدرس الکترونیکی <http://www.wikipedia.org> نشان داده شود.
۵. با استفاده از طرزالعمل CREATE TABLE ... یک جدول در سیکویل ایجاد کرده و نوع دیتای فیلدهای آن را CHAR و VARCHAR تعیین کرده و به جدول جدید دیتا را طوری داخل کنید که محتواها با خانه‌های خالی همراه باشند.
۶. یک کیوری ساده را با استفاده از SELECT \* FROM ... بالای جدول ایجادشده در فعالیت قبلی اجرا کنید.
۷. کیوری فعالیت قبلی را با استفاده از تابع CONCAT() طور ذیل دوباره اجرا کنید.  

```
>> SELECT CONCAT('(', fieldName, ')'), CONCAT('(', fieldName, ')') FROM TableName;
```

حال نتیجه کیوری را با نتیجه کیوری فعالیت قبلی مقایسه نموده، تفاوت را مشاهده کنید.
۸. با استفاده از طرزالعمل DROP TABLE IF EXISTS TableName یک‌بار ببینید که جدولی به نام مشخصی که در فعالیت بعدی باید ایجاد شود در دیتابیس موجود نیست. (اختیار IF EXISTS در این طرزالعمل کمک می‌کند که از اشتباه جلوگیری شود).
۹. با استفاده از طرزالعمل CREATE TABLE ... جدولی بسازید که در آن یکی از فیلدها، نوع دیتای ENUM را استفاده کرده و فهرست عناصر از قبل تعیین‌شده را اجازه داخل شدن به جدول بدهد.
۱۰. با استفاده از طرزالعمل INSERT INTO ... به جدول جدیداً ایجاد شده در فعالیت قبلی 3 ریکورد دیتای مناسب را داخل کنید.
۱۱. یک کیوری ساده را با استفاده از طرزالعمل SELECT \* FROM ... بالای جدول ایجاد شده در فعالیت قبلی اجرا کنید.
۱۲. یک کیوری را با استفاده از طرزالعمل SELECT ... و توابع SUM() و AVG() بالای فیلد نوع ENUM در جدول ایجادشده اخیر اجرا کنید. نتیجه تابع‌ها را با محتواهای فیلد مقایسه کنید.



## سوالات فصل پنجم

۱. لسان XML چه نوع لسان است؟ توضیح دهید.
۲. XML و SGML از کدام کلمات گرفته شده است؟
۳. هدف از استفاده تگ‌ها (Tags) در XML چیست؟
۴. سه خصوصیت مهم لسان XML را مختصراً توضیح دهید.
۵. مفهوم کلمه Markup را در یک جمله بیان کنید.
۶. تگ‌های یک تایی در XML کدام‌ها اند؟ توضیح دهید.
۷. اعلامیه XML یا (XML Declaration) دارای کدام اختیارات است؟ مشخص کنید.
۸. آیا استفاده از حروف کوچک و بزرگ در کد اعلامیه XML تفاوت می‌کند یا خیر؟ بیان کنید.
۹. آیا استفاده از حروف کوچک و بزرگ در نام تگ‌های XML تفاوت می‌کند یا خیر؟ شرح دهید.
۱۰. آیا اعلامیه XML بدون پارامترها استفاده شده می‌تواند؟ اگر جواب بلی است، مثال آن نوشته شود.
۱۱. سمبول‌هایی که در شروع و ختم نام ریفرنس‌ها در XML استفاده می‌شوند کدام‌ها اند؟ با مثال توضیح دهید.
۱۲. در قسمت پرلُگ یک سند XML کدام چیزها مجاز است که نوشته شوند؟
۱۳. حد اعظمی طول محتوای نوع دیتای متنی CHAR در سیکویل، بیش‌تر است و یا از نوع دیتای VARCHAR بیش‌تر است؟ با دلیل روشن سازید.
۱۴. چهار کتگوری نوع دیتای BLOB را نام بگیرید.
۱۵. چهار کتگوری نوع دیتای TEXT را نام بگیرید.
۱۶. نوع دیتای ENUM از کدام کلمه گرفته شده است؟ آن را مختصراً شرح دهید.
۱۷. مولتی‌مدیا را تعریف کرده عناصر آن را نام بگیرید.
۱۸. فارمت‌های ذخیره فایل‌های صوتی و تصویری (ویدیوها) را بنویسید.

## فصل ششم

### دیتای روی وب (Data on the Web)



**هدف کلی:** محصلان در مورد دیتای روی وب (Data on the Web) معلومات حاصل نمایند.

**اهداف آموزشی:** در پایان این فصل محصلان قادر خواهند شد تا:

۱. ترکیب دیتا (Data Integration) و سطوح ترکیب دیتا را توضیح کنند.
۲. مفاهیم کلیدی شامل انواع پیوندهای دیتا، ارزیابی کیفیت ترکیب دیتا و سناریوهای ترکیب دیتا را بفهمند.
۳. دانستن هدف و نحوه کاری نماها (Views) را بدانند.
۴. استفاده از طریقه‌های بازیافتن معلومات را بلد شوند.
۵. کیوری‌های ساختاری روی وب را ایجاد و استفاده کنند.

یک اصطلاح معروف است که در آن معلومات به مثابه قوت شمرده شده است؛ یعنی داشتن معلومات در یک عرصه، اشخاص صاحب صلاحیت در آن عرصه را قادر به تصمیم‌گیری‌های مؤثر، تأیید و یا رد مسایل و غیره کارها می‌سازد. به هر اندازه‌یی که معلومات در یک موضوع وسیع باشد به همان اندازه اطمینان از استفاده به‌جا و کارایی از معلومات بیش‌تر می‌شود. ترکیب دیتا از منابع مختلف باعث می‌شود تا مجموعه بزرگ از معلومات دقیق، ایجاد شوند. دیتا از منابع مختلف به سطوح مختلف با هم ترکیب شده می‌تواند. از ترکیب دیتا به سطح ریکوردهای دیتا و یا هم ترکیب دیتا به سطوح بالاتر مانند ترکیب نتایج کیوری‌ها را می‌توان در زمینه مورد نظر به کار گرفت. به خاطر به دست آوردن نتایج مطلوب و اطمینان از ترکیب دیتا، ارزیابی کیفیت ترکیب دیتا به اساس قواعد و یافته‌های مؤثر صورت می‌گیرد. بازیافتن معلومات از منابعی که دیتا در آن‌ها ذخیره شده به کمک طریقه‌های عملی انجام شده می‌تواند. معلومات امکان دارد به اشکال مختلف در کمپیوترها و منابع آنلاین موجود باشد؛ یعنی معلوماتی که در کمپیوترها و وسایل الکترونیکی قابل استفاده بوده و از نوع معلومات دیجیتل به حساب می‌آیند، قابل بازیابی و استفاده دوباره است. بازیافتن معلومات و بازیافتن دیتا، دو مورد مهم در زمینه به شمار می‌روند. در قسمت بازیافتن معلومات فایل‌ها و سندهایی که معلومات را در بر داشته باشند به حیث دخولی دیتا استفاده می‌شوند. در مورد دوم معلومات به اساس ریکوردهای دیتای تحلیل شده، بازیابی و به کار گرفته می‌شود. تفاوت‌هایی در هر دو حالت وجود دارد که در جریان فصل به آن‌ها پرداخته خواهد شد. کیوری‌ها عبارت از وسایل مهم در قسمت استخراج دیتا از منابع آن‌ها به شمار می‌روند. در دیتابیس‌ها به خاطر دیدن، فیلتر و ترکیب دیتا از کیوری‌ها استفاده می‌شود. زمانی که بحث به معلومات روی وب می‌رود، کیوری‌ها باز هم از وسایل مهم در زمینه مورد نظر به شمار می‌روند. یک کیوری می‌تواند بالای معلوماتی که در یک صفحه وب گذاشته شده است، به راه انداخته شده و نتایجی از آن استخراج شود. مدل‌ها به خاطر کار با دیتا و معلومات روی صفحات وب طرح شده و استفاده می‌شوند. این مدل‌ها مستقیماً به کمک کیوری‌های ساختاری روی وب تنظیم می‌شوند. سه مدل نمونه، به نام‌های مدل استخراج کردن سکیم، مدل کیوری متنی و مدل استخراج (دیتا) برای دیتابیس از مثال‌های قابل استفاده در قسمت معلومات آنلاین به شمار می‌روند.

## ۶.۱ ترکیب دیتا (Data Integration)

ترکیب دیتا به مفهوم یک جاسازی ریکوردهای دیتا از منابع مختلف به خاطر یک هدف واحد است. دیتا در ساحة کمپیوترساینس هم می‌تواند در دیتابیس‌ها تنظیم و ذخیره شود و هم می‌تواند با فارمت‌های مختلف دیگر به شکل دیجیتل موجود باشد. در ترکیب دیتا انواع مختلف مانند دیتای سروی درباره یک موضوع و دیتای بخش‌های گوناگون اداری با هم ترکیب شده و معلومات کامل را در ارتباط به موضوع تهیه می‌شود. دیتابیس‌ها به کمک دسته‌یی از پروگرام‌ها به نام سیستم مدیریت دیتابیس (DBMS) ایجاد و استفاده می‌شوند. دیتای که در دیتابیس‌ها ذخیره باشند دوباره توسط همین سیستم‌ها به کمک کیوری‌های جامع، قابل استفاده بوده که این خود یک مثال ترکیب دیتا از جدول‌های مختلف داخل یک دیتابیس است. زمانی که دیتای ترکیبی از محدوده یک دیتابیس خارج می‌شود و منابع دیگری مانند دیتابیس‌های دومی و سومی و یا هم



فارمتهای دیگر دیتای دیجیتل در کار باشد، رُل مهم ترکیب دیتا (Data Integration) بیش تر هویدا می گردد.

میخانیکیت ترکیب دیتا در این است که دیتای منابع مختلف با فارمتهای مختلف، قابل خواندن و ترجمه با مفاهیم واحد باشد. شناسایی ریکوردهای دیتا به اساس استندهای قبول شده و استفاده مناسب از آن ریکوردها از جمله مسایل قابل بحث در ترکیب دیتا است؛ یعنی زمانی که دیتا از یک منبع، طور مثال دیتابیس مشخص در یک سیستم مدیریت دیتابیس، گرفته شده و با دیتای دیگری که از یک صفحه انترنت اخذ شده، ترکیب شود، آیا اطمینان کامل از ترجمه مفاهیم واحد در آن ها است و یا خیر. معلوم کردن هم چون موارد با تحلیل نتایج به دست آمده از معلومات راپورهای نهایی تهیه شده می تواند. دقت راپورهای معلوماتی موفقیت پروسه ترکیب دیتا را نشان خواهد داد.

زوایای دیگر موضوع ترکیب دیتا، مسایلی از قبیل ارتباط ریکوردها (Record Linkage) و پیدا کردن دیتای هم خوان (Data Matching) می باشند؛ یعنی طریقه های ترکیب دیتا، از نظر ارتباط بین ریکوردهای دیتا و یا هم پیدا کردن بخش های ارتباطی دیتا از نظر مفاهیم نهفته در دیتا را مد نظر گرفته و مورد پروسس قرار بگیرند. در حالت اول (ارتباط بین ریکوردها) مسئله بیش تر به سمت دیتابیس ها و فارمتهای مشابه به دیتابیس ها مانند شیت های برنامه میکروسافت اکسیل می رود.

در این حالت معمولاً فیلدهای شناسه (Identifier Fields) رُل بیش تری را در ارتباط بین ریکوردهای دیتا افاده می کنند. حالت بعدی (Data Matching) برمی گردد به دیتای متنی و بسته های دیتا که در آن معلومات به شکل رشته های از حروف و جملات ذخیره شده باشند. در این حالت، پیدا کردن دیتای هم خوان از نظر معنی توسط الگورتم ها انجام می گیرند. نتیجه هر دو حالت شامل طریقه های ترکیب دیتا (Data Integration) بوده و تحت شرایط مشخص مورد مطالعه قرار می گیرند.

### ۶.۱.۱ سطوح ترکیب دیتا (Data Integration Levels)

ترکیب دیتا امکان دارد به سطح پایینی انجام شود و یا هم به سطح بالایی به اجراء گذاشته شود. سطح پایین ترکیب دیتا، همان ارتباط بین ریکوردهای دیتا را شامل شده و به نام میکرو-لیول (Micro Level) یاد می شود. نوع دیگر ترکیب دیتا که به شکل پیدا کردن دیتای هم خوان (Data Matching) پیش تر قسمتاً توضیح شد، به نام ترکیب دیتای سطح بالایی یا مکرو-لیول (Macro Level) یاد می شود. زمانی که ترکیب دیتا به سطح پایین انجام می شود، ریکوردهای دیتای یک واحد یک بخش، به ریکوردهای دیتای یک واحد در بخش دیگر پیوند داده می شوند. یک واحد عبارت از همان کوچک ترین ساختمان یک دیتابیس است که در آن ریکوردهای دیتا ذخیره می شوند. اصطلاح تخنیککی کوچک ترین واحد ذخیره دیتا در مدل ارتباط عبارت از رابطه (Relation) بوده و در اصطلاح آن را جدول می گویند.

پیوند دادن واحدها و بخش های مختلف منابع دیتا، می تواند به صورت مستقیم بین جدول های طرفین باشد و یا هم در بین بخش هایی ایجاد شود که دارای خصوصیت های مشابه باشند. در ترکیب دیتا به سطح بالایی

(Macro Level) مجموعه‌های احصائیوی دیتا به اساس دسته‌هایی از معلومات در مورد یک موضوع مشخص با معلومات دیگر از عین موضوع مقایسه شده و نتایج به دست می‌آید. در موضوعات این کتاب بیش‌تر به ترکیب دیتا به سطح پایین (Micro Level) پرداخته شده و از نوع دوم ترکیب دیتا (سطح بالایی) صرف نظر صورت گرفته است.

## ۶.۲ مفاهیم کلیدی در قسمت ترکیب دیتا

ترکیب دیتا (Data Integration) یکی از بحث‌های جالب در ساحة مدیریت دیتا به شمار می‌آید. تبدیل کردن ریکوردهای دیتا به معلومات کارآمد و مهم از جمله مسایلی به شمار می‌رود که قوت استفاده از معلومات را بیش‌تر می‌سازد. ترکیب دیتا از منابع مختلف با فارمت‌های مختلف یکی از ساحات مهم کار با دیتا و ارقام به شمار می‌رود. کارکردن در ساحة ترکیب دیتا ایجاب می‌نماید تا استفاده‌کننده با مفاهیم کلیدی در این ساحة بلدیت داشته باشد. بعضی مفاهیم کلیدی که در ساحة ترکیب دیتا حائز اهمیت بوده و به استفاده‌کنندگان زمینه کارکردن در این ساحة را بیش‌تر مساعد می‌سازد، در ذیل به توضیح گرفته شده است.

### ۶.۲.۱ ترکیب دیتا به اهداف تهیه معلومات احصائیوی و مدیریتی

زمانی که ریکوردهای دیتا به هدف تهیه معلومات احصائیوی با هم ترکیب می‌شوند، واحدهای دیتا به خاطر ایجاد پیوند بین‌شان شناسایی و پیدا می‌شوند. این نوع پیوندها به کمک ساختمان‌های جداگانه از منابع دیتا ایجاد می‌شوند. به کمک پیوندهای دیتا از منابع اصلی آن‌ها، چه دیتابیس باشند و یا فارمت‌های دیگر، به شکل فقط خواندنی (Read Only) گرفته شده و در ست جدید معلومات احصائیوی گذاشته می‌شود. بعد از آنکه پیوند بین واحدهای دیتا در یک منبع و یا هم در منابع مختلف دیتا ساخته شود، ضرورت بعدی به تشخیص منابع که دیتا از آن‌ها گرفته شده است، تقلیل پیدا می‌کند. دیتا ست پیوندشده به منابع دیتا، به خاطر گرفتن دیتا و راپوردادن آخرین ارقام از منابع استفاده می‌شود.

یعنی استفاده‌کننده در هر بار استفاده از معلومات، ضرورت ندارد تا به منبع اصلی دیتا رفته و از آنجا معلومات را به دست آورد. فقط فارمت ترکیب‌شده دیتا کفایت می‌کند تا توسط استفاده‌کننده به کار گرفته شده و معلومات به‌روزشده را از آن بگیرد. به طور مثال واحدهای دیتای یک فروشگاه و دیتای اشخاص یک منطقه از دو منبع مختلف گرفته شده و با هم ترکیب‌شده است. اول انترفیس وب، دیزاین شده است که نشان می‌دهد چه کسی، در کدام تاریخ از فروشگاه مشخص شده، چه چیزی را خریداری نموده است. این ترکیب امکان دارد با استفاده از نمبر تذکره شخصی که در هر دو منبع دیتا (دیتابیس فروشگاه و دیتای اشخاص منطقه مذکور) استفاده شده است، تنظیم شده باشد. هر زمانی که دیتای ترکیب‌شده از طریق وب‌سایت دیده می‌شود، الگوریتم استفاده شده در آن در عقب پرده (Backend) به ریکوردهای منابع دیتا رفته، آخرین ارقام را کیوری نموده و نمایش می‌دهد.

## ۶.۲.۲ پیوند دقیق و پیوند احتمالی

به خاطر پیوند کردن دیتا از آدرس ترکیب دیتا (Data Integration) دو طریقه استفاده شده می‌توانند. یکی به نام پیوند دقیق (Exact Linkage) یاد شده و دیگری به نام پیوند احتمالی (Probabilistic Linkage) یاد می‌شود. نوع اول پیوندی است که به اساس فیلدهای شناسایی (Identifier Fields) جدول‌ها در منابع دیتا صورت می‌گیرد. طوری که از نام آن پیداست، نتیجه ترکیب این نوع دیتا دقیق و عاری از اشتباه است. شرط تطبیق این نوع پیوند، موجودیت عین فیلدهای شناسایی در منابع دیتا است. نمبر تذکره اشخاص از مثال‌های قابل اجراء در پیوندهای نوع دقیق به شمار می‌روند. در مثال بحث قبلی، معلومات خریداری اشخاص یک منطقه از یک فروشگاه مشخص نیز به اساس چنین فیلدی صورت گرفته است. استفاده از پیوند دقیق، ساده‌ترین و مؤثرترین طریقه به خاطر ارتباط دادن دیتا ست‌های مختلف و سافت‌ویرهای احصائیوی به شمار می‌رود.

نوع دوم ارتباط بین واحدهای دیتا به نام پیوند احتمالی (Probabilistic Linkage) در مواردی استفاده می‌شود که در واحدهای دیتا یک شناسه غیر تکراری دیتا موجود نباشد و یا هم قابل اعتبار نباشد؛ یعنی در قدم نخست زمانی که دیتا ترکیب می‌شود، از پیوند دقیق باید استفاده شود، در صورت عدم موجودیت امکانات برای تطبیق پیوند دقیق به اختیار دومی یعنی استفاده از پیوند احتمالی وصل می‌شود. در صورت استفاده از پیوند احتمالی، بخش‌های دیگری از دیتا ست‌ها در نظر گرفته می‌شود که در هر دو واحد دیتا موجود بوده و از عین میخانیکیت برخوردار باشد. طور مثال نام‌ها، آدرس‌ها، تاریخ‌های تولد و جنس اشخاص می‌تواند یکی از گزینه‌های قابل استفاده باشد. پیوند احتمالی به تناسب پیوند دقیق بیش‌تر پیچیده است. به خاطر استفاده درست با نتایج بهتر به کارگیری و استفاده از سافت‌ویر مناسب جهت تطبیق پیوند احتمالی ضروری است.

## ۶.۲.۳ ارزیابی کیفیت ترکیب دیتا

راه‌های پیوند واحدهای دیتا از منابع مختلف، ممکن است که باعث اشتباهات در نتایج حاصله ست دیتایی نهایی شوند. اشتباهاتی که در موارد یاد شده امکان اتفاق افتادن آن‌ها متصور است می‌توانند یکی از این دو نوع باشند:

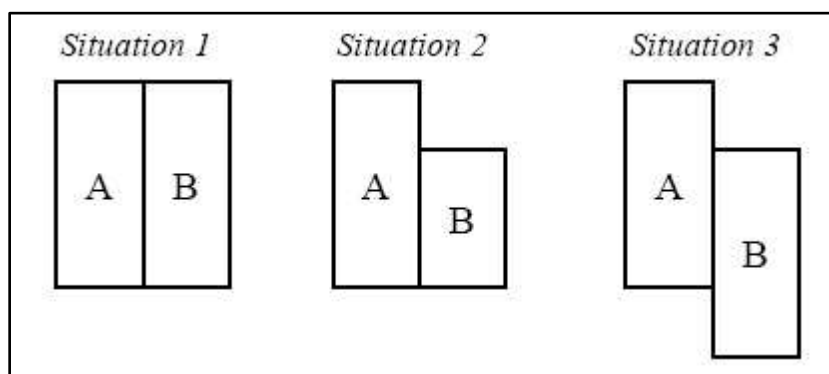
- False Positive Matching: نتیجه پیوند ریکوردهای دیتا مثبت بوده ولی از نظر هدف پلان‌شده درست نیست.
- False Negative Matching: نتیجه پیوند ریکوردهای دیتا منفی بوده و هم‌چنان نتیجه از نظر هدف پلان‌شده نیز درست نیست.

در حالت اول دو ریکورد دیتا از واحدهای مختلف با هم‌دیگر به صورت مؤفقانه پیوند می‌شوند ولی ریکوردهای دیتا مربوط یک شخص یا یک واحد موضوع نمی‌باشند. با در نظر داشت مثال قبلی، ریکورد خریداری‌های فروشگاه توسط شخص الف انجام شده و در زمان ترکیب دیتا، نتیجه عملیه مثبت بوده ولی خریداری‌ها برای شخص ب نشان داده شده‌اند.

در حالت دوم، نتیجه پیوند بین دو ریکورد از واحدهای مختلف که در ترکیب دیتا از ایشان استفاده شده است، منفی یا ناقص است؛ یعنی پیوند بین ریکوردهای دیتا انجام نمی‌شود. در حالی که ریکوردهای دیتای پیوند ناشده در واحدهای مختلف به یک شخص و یا به یک موضوع نیز مربوط نمی‌باشد. در صورت انجام مؤفقانه پیوند، نتیجه باید نادرست می‌بود.

## ۶.۲.۴ سناریوهای ترکیب دیتا

طریقه‌های مختلف به خاطر ترکیب دیتا موجود است. اول استفاده‌کننده می‌تواند نظر به حجم دیتا و امکانات استفاده از سافت‌ویرهای موجود، یکی از راه‌های مناسب را جهت ترکیب دیتا اختیار کند. در شکل پایین حالت‌های ممکنه به خاطر ترکیب دیتا بین دو واحد منابع مختلف دیتا به شکل گرافیکی نشان داده شده است.

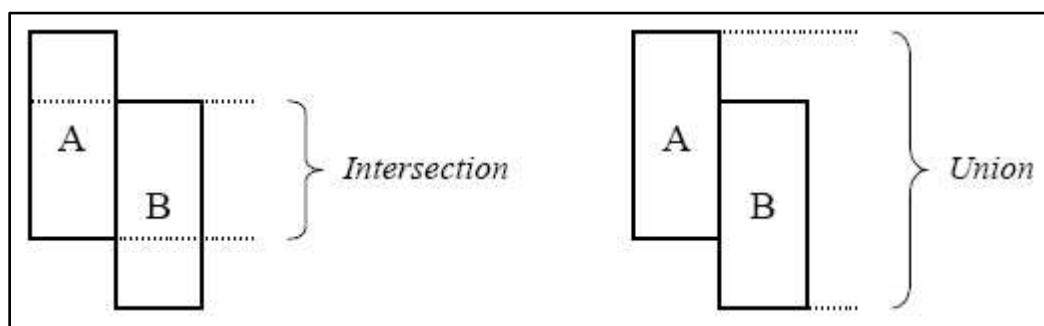


شکل ۶-۱ حالت‌های ممکنه به خاطر ترکیب دیتا بین دو واحد مختلف دیتا

- حالت اول شکل بالا: در این حالت هر ریکورد دیتا در دیتا ست الف، هم‌چنان در دیتا ست ب موجود است؛ و نیز هر واحد دیتا در دیتا ست ب، هم‌چنان در دیتا ست الف موجود است. طور مثال دیتا ست الف حاوی آدرس‌های اجناس در یک فروشگاه است و دیتا ست ب در بر گیرنده قیمت برای هر جنس موجود در دیتا ست الف است؛ یعنی قیمت‌های تمام اجناس جدول الف در جدول ب موجود بوده و به عین ترتیب تمام قیمت‌های ذکر شده در جدول ب مربوط به اجناس در جدول الف است.
- حالت دوم شکل بالا: در این حالت هر ریکورد دیتا در دیتا ست ب هم‌چنان در دیتا ست الف موجود است در حالی که در دیتا ست الف ریکوردهایی موجوداند که در دیتا ست ب به مشاهده نمی‌رسند؛ یعنی دیتا ست ب یک ست فرعی دیتا ست الف است. مثال این حالت، جدول الف شامل معلومات تمام اجناس فروشگاه و اجناس ذخیره‌گاه فروشگاه می‌باشد. جدول ب تنها قیمت‌های اجناسی را داشته باشد که آماده فروش بوده و در سالون فروشگاه قرار دارند؛ یعنی قیمت‌های اجناس ذخیره‌گاه در جدول ب موجود نباشند.
- حالت سوم شکل بالا: در این حالت، طوری که در شکل نیز دیده می‌شود، دسته‌هایی از ریکوردها در هر دو دیتا ست الف و ب موجود است که ریکوردهای معادل‌شان به خاطر ترکیب در جدول مقابل موجود نمی‌باشند؛ یعنی بر علاوه ریکوردهای مشابه که در هر دو دیتا ست موجود است، ریکوردهای اضافی نیز در هر دیتا ست‌ها دیده می‌شوند. مثال حالت سوم: جدول الف شامل معلومات تمام اجناس

فروشگاه و ذخیره‌گاه فروشگاه باشد. جدول ب قیمت‌های اجناسی را داشته باشد که آماده فروش بوده در فروشگاه موجود است و بر علاوه، قیمت اجناسی را هم ذخیره نموده باشد که در فروشگاه موجود نبوده ولی در مارکیت پیدا می‌شوند.

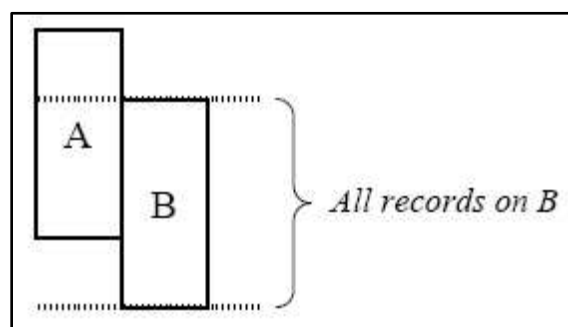
بر علاوه این‌ها بعضی نتایج متفاوت از دیتا ست‌هایی که در ترکیب دیتا به کار گرفته می‌شوند، به اساس عملیه‌های استاندارد ست‌ها مانند اتحاد (Union) و تقاطع (Intersection) نیز بالای ریکوردهای دیتا ست‌ها ممکن بوده و قابل تطبیق می‌باشند. شکل پایین هر دو عملیه را به صورت گرافیکی نشان می‌دهد.



شکل ۲-۶ تمثیل عملیه‌های اتحاد و تقاطع بالای دیتا ست‌ها به خاطر ترکیب دیتا

مثال حالت سوم از بحث قبلی اگر در نظر گرفته شود، معلومات اجناس فروشگاه‌هایی که آماده فروش است از جدول الف گرفته شده و قیمت همان اجناس از جدول ب نیز گرفته شده و با هم ترکیب می‌شوند. نتیجه این ترکیب تقاطع دو جدول الف و ب را نشان خواهد داد.

به عین ترتیب، اگر نام تمام اجناس از جدول الف و هم‌چنان از جدول ب گرفته شوند و از تکرار ریکوردها جلوگیری شود، اتحاد جدول‌های الف و ب به دست خواهد آمد. اگر فرض کنیم جدول‌های الف و ب هر کدام به ترتیب 100 و 150 ریکورد دارند، پس به تعداد 50 ریکورد در هر دو جدول مشابه بوده عین اجناس را نشان می‌دهند. در نتیجه تطبیق عملیه اتحاد بر آن‌ها، تعداد ریکوردهای به دست آمده عبارت از  $100 + 150$  منهای پنجاه یعنی 200 ریکورد دیتا خواهد بود؛ و نتیجه تطبیق عملیه تقاطع همان 50 ریکورد مشترک است. در بعضی حالات طوری اتفاق می‌افتد که تمام ریکوردهای یک دیتا ست گرفته شده و ریکوردهای مشابه که به اثر ترکیب دیتا به دست می‌آیند، به آن اضافه شده و نشان داده می‌شوند. شکل پایین این نوع عملیه را نشان می‌دهد.



شکل ۳-۶ نشان کرده ریکوردهای مکمل یک دیتا ست جمع ریکوردهای مشابه از دیتا ست دومی به کمک ترکیب دیتا

در مثال خریداری اجناس از فروشگاه اگر ضرورت باشد تا معلومات تمام کسانی که از یک منطقه مشخص اند، ارائه شود و در ضمن معلومات کسانی که از فروشگاه معین خریداری انجام داده اند نیز با معلومات متذکره مربوط اشخاص، ترکیب شود. در چنین حالتی فهرست مکمل از جدول اشخاص (تمام ریکوردهای جدول اشخاص) کیوری شده و با آن معلومات خریداران از فروشگاه نیز چک می شود و تنها ریکوردهای خریداری توسط اشخاص از جدول اشخاص ترکیب (ضمیمه) می گردد. در چنین نوع ترکیب دیتا، فیلدهای خریداری که از دیتا ست فروشگاه به جدول نتیجه اضافه شده اند، برای اشخاصی که خریداری انجام نداده اند، محتوای ناشناخته (Null Values) را نشان داده و برای کسانی که خریداری انجام داده اند، معلومات موجوده ارائه می شود؛ یعنی در هم چو موارد احتمال داشتن دیتاهای ناشناخته در جدول نتیجه بیش تر شده و به هر اندازه ای که تعداد ریکوردها در جدول اول بیش تر باشد، به همان اندازه محتوای ناشناخته بیش تر خواهد بود.

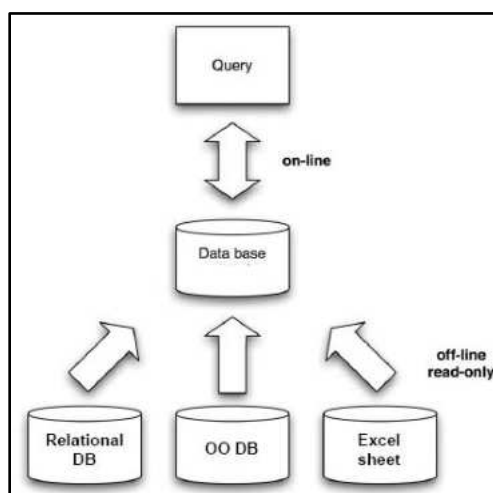
### ۶.۳ ترکیب فزیکی و ترکیب مجازی دیتا

زمانی که معلومات درباره یک موضوع مشخص و مورد علاقه، توسط یک استفاده کننده به کمک ترکیب دیتا از منابعی به دست می آیند، طریقه های مختلفی استفاده شده می توانند. به صورت عمومی طریقه های دسترسی به معلومات و ترکیب دیتا به دو کتگوری تقسیم می شوند:

- ترکیب فزیکی دیتا؛
- ترکیب مجازی دیتا.

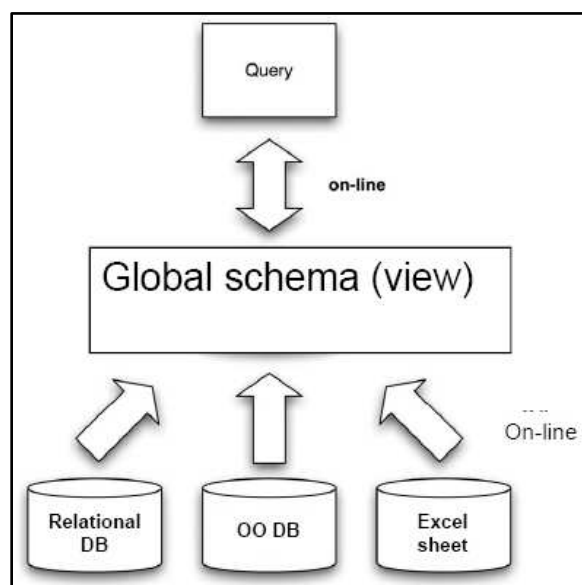
در ترکیب فزیکی، عناصر دیتا از دیتابیس و یا دیتابیس های موجود در یک سیستم گرفته شده و با هم ترکیب می شوند. بعد از ترکیب دیتا، ساختمان های جدید ایجاد شده و در یک دیتابیس جدید جا به جا می شوند؛ یعنی با استفاده از این نوع ترکیب دیتا، منابع اصلی شان کاپی شده و در منبع جدید (دیتابیس جدید) ذخیره و قابل دسترس می شود. بحث ترکیب فزیکی دیتا بیش تر بر می گردد به عناوین دیتا ویرهوس و دیتاماینگ که در فصل اخیر کتاب به آنها پرداخته خواهد شد.

شکل پایین نمای ترکیب فزیکی دیتا را نشان می دهد.



شکل ۶-۴ ترکیب فزیکی دیتا از منابع مختلف به شکل یک نمونه

ترکیب مجازی دیتا با استفاده از کد سیکویل جهت ایجاد کیوری‌ها صورت می‌گیرد. در ترکیب مجازی هر جزء دیتا در منبع اصلی خود بوده، تنها در زمان اجرای طرزالعمل‌های ترکیب دیتا، عناصر دیتا از منابع اصلی آن‌ها خوانده شده و معلومات مورد نظر تهیه می‌گردد. زمانی که منابع ترکیب دیتا از یک دیتابیس باشد، این نوع ترکیب به نام ترکیب محلی دیتا (Local Integration) یاد می‌شود. وقتی که منابع دیتا که با هم ترکیب شده از دیتابیس‌های مختلف باشند، آن‌را به نام ترکیب عمومی دیتا (Global Integration) یاد می‌کنند؛ یعنی منابع دیتا خارج از یک دیتابیس بوده و از سیستم‌های دیگر موجود در کمپیوتر و سیستم‌های مدیریت دیتابیس استفاده به عمل می‌آید. شکل پایین نمای ترکیب مجازی دیتا را نشان می‌دهد.



شکل ۵-۶ ترکیب مجازی دیتا با ذکر منابع اصلی دیتا

طوری که گفته شد، در ترکیب مجازی دیتا (Virtual Integration)، ریکوردهای دیتا در منابع اصلی خودشان باقی می‌مانند. منابع دیتا امکان دارد دیتابیس‌ها باشند، شیت‌های برنامه‌های میکروسافت اکسیل باشند و یا هم دیتابیس‌های آنلاین و صفحات وب باشند. چون دیتای اصلی در منابع خودشان تبدیل می‌شود، با هر بار خواندن (ترکیب کردن) آن، تازه‌ترین دیتا به دسترس قرار می‌گیرد؛ یعنی دیتا همیشه به‌روز استفاده می‌شود. چلنج کار یا به عبارت ساده‌تر مشکل کار همان دسترسی به دیتا و تنظیم درست کیوری‌ها جهت گرفتن دیتا از منابع اصلی آن است. اگر میتودهای گرفتن دیتا و ترکیب آن با ست‌های دیگر دیتا به صورت دقیق و درست انجام شود، نتایج حاصله با کیفیت و مطمئن خواهند بود. شکل پایین یک ترکیب دیتا را از دو دیتابیس که در سیستم‌های مدیریت دیتابیس جداگانه و از هم متفاوت دیزاین شده‌اند، نشان می‌دهد.

Multidatabase schema (e.g. DataJoiner)	
<pre> Sybase.PUBLICATIONS(PNR, TITLE, AUTHOR, JOURNAL) Sybase.AUTHORS(ANR, TITLE, NAME, AFFILIATION) Oracle.PAPERS(NUMBER, TITLE, WRITER, PUBLISHED) Oracle.WRITER(FIRSTNAME, LASTNAME, NROFPUBLICATIONS) </pre>	
<pre> PUBLICATIONS(PNR, TITLE, AUTHOR, JOURNAL) AUTHORS(ANR, TITLE, NAME, AFFILIATION) </pre>	<pre> PAPERS(NUMBER, TITLE, WRITER, PUBLISHED) WRITER(FIRSTNAME, LASTNAME, NROFPUBLICATIONS) </pre>
Database 1 (e.g. Sybase) schema	Database 2 (e.g. Oracle) schema

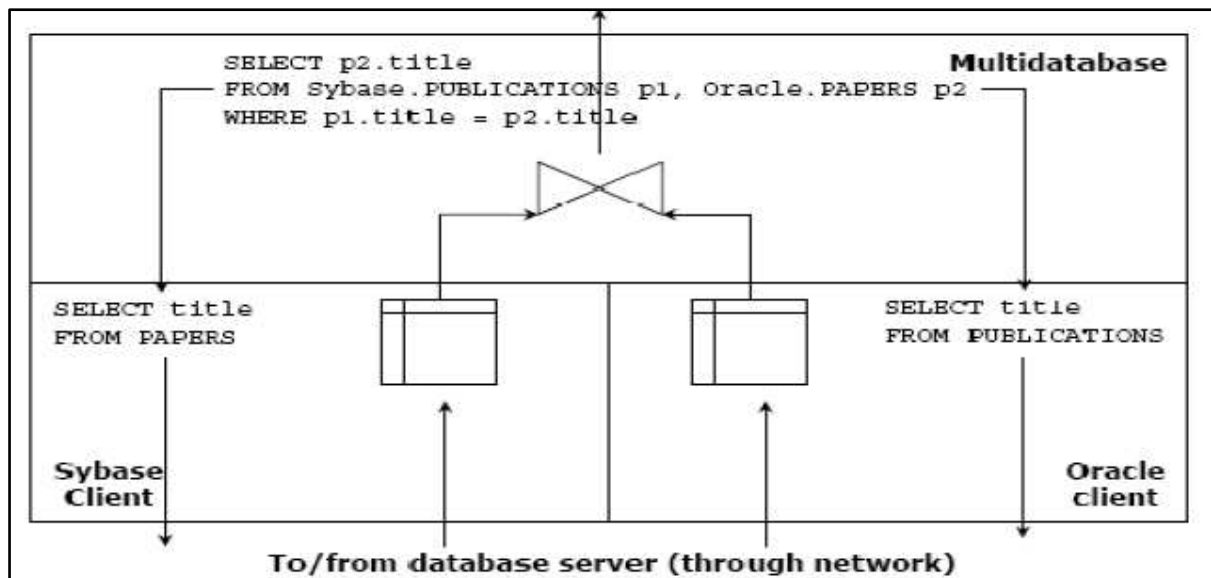
شکل ۶-۶ مثالی از ترکیب مجازی دیتا

### اجرای کیوری بالای دیتای ترکیب شده (Integrated Data)

طرز العمل‌های DML بالای دیتای ترکیب شده از منابع مختلف قابل اجراء است. وقتی بحث روی ترکیب دیتا باشد، معمولاً هدف همانا استفاده از واحدهای مختلف دیتا از منابع جداگانه است. در حالات عادی طرز العمل‌های کار با یوزر دیتا (Data Manipulation Languages) سیکویل در یک دیتابیس قابلیت اجرا را دارند. کیوری دیتا از یک دیتابیس و یا از دیتابیس‌های مختلف به حیث یکی از طرز العمل‌های اساسی کار کردن با یوزر دیتا استفاده می‌شود. طرز العمل کیوری دیتا وقتی بالای دیتای ترکیب شده از دو دیتابیس مختلف به راه انداخته شود، سیستم آن کیوری را به بخش‌های ذیل تقسیم می‌کند:

- تشخیص این که کدام جدول‌ها (واحدهای دیتا) از کدام دیتابیس‌ها (بخش‌های دیتا) ضرورت‌اند.
  - کدام قسمت‌های طرز العمل کیوری (اختیارها) بالای کدام یک از دیتابیس‌ها تطبیق شوند.
  - کدام قسمت‌های طرز العمل کیوری (اختیارها) بالای هر دو دیتابیس تطبیق شوند.
- شکل پایین نتیجه کیوری بالای دو دیتابیس را از محیط‌های مختلف اوریکل و سایبیس نشان می‌دهد. کد طرز العمل اصلی (کیوری) در قسمت بالایی نشان داده شده و تطبیق آن جهت گرفتن دیتا از منابع مختلف (دیتابیس‌های مختلف) در قسمت پایین شکل نشان داده شده است.





شکل ۶-۷ نمایشی از تطبیق کیوری بالای دیتا از دو منبع (دیتابیس) مختلف

#### ۶.۴ استفاده از نماها (Views)

نماها یا Views عبارت از ساختمان‌هایی‌اند که توسط آن‌ها بخش‌های یک دیتابیس و یا دیتابیس‌های مختلف به شکل فقط خواندنی (Read Only) با هم ترکیب شده و در نتیجه دیتای ترکیب‌شده، نشان داده می‌شود. یک نما قابلیت تطبیق بالای واحدهای دیتا از یک دیتابیس را دارد و یا هم یک نما می‌تواند به خاطر ترکیب واحدهای دیتا از دیتابیس مختلف ایجاد و استفاده شود. هر باری که یک نما اجراء می‌شود، آخرین دیتای به‌روزشده را از منابع تعریف‌شده در نما، استخراج می‌کند؛ به عبارت دیگر، یک نما عبارت از شکل ذخیره‌شدهٔ کود کیوری در ساحة مدیریت دیتا است.

یک کیوری هر باری که ضرورت به اجراء داشته باشد، باید کود آن نوشته و یا از مرجعی به سیستم لود شود. در حالی که نماها به حیث ساختمان‌های مستقل قابل تعریف در دیتابیس‌ها به یک نام ذخیره و استفاده شده می‌توانند. مزیت دیگر استفاده از نماها در مدیریت دیتا، تمثیل یک ساختمان جدید دیتا (مانند جدول) در دیتابیس‌ها است؛ یعنی یک نما وقتی با یک نام مشخص در یک دیتابیس تعریف شد، همان نما به شکل یک منبع دیتا در کیوری‌ها و نماهای دیگر قابل استفاده است.

نحوهٔ ایجاد نما در سیکویل به شکل ذیل است.

```
CREATE VIEW ViewName AS SELECT ... FROM ... WHERE ...
```

طور مثال یک نما به نام V1 از تمام جدول‌های دیتابیس مکتب (جدول کارمند، جدول مأموریت و جدول پروژه) بدون وضع شرایط طور ذیل ایجاد شده می‌تواند:

```
>> CREATE VIEW V1 AS SELECT * FROM EMPLOYEE, ASSIGNMENT, PROJECT
WHERE 1 = 1;
```

در مثال بالا تمام فیلدهای جدول‌های ذکر شده کیوری گردیده و در نمای جدید به نام V1 ذخیره می‌شود. البته مثال کیوری بالا مؤفقانه اجرا نمی‌شود. دلیل آن موجود بودن فیلدهای هم نام در جدول‌های دیتابیس مکتب است. به خاطر تطبیق طرزالعمل بالا در قسمت انتخاب جدول‌ها و فیلدهای جدول‌ها باید با دقت صورت گیرد. در صورت استفاده تمام فیلدهای جدول‌ها از اختیار تغییر نام فیلدها در طرزالعمل ایجاد نماها استفاده شده می‌تواند. به خاطر وضاحت بیش‌تر موضوع یک مثال دیگر از دیتابیس مکتب 2 که قبلاً ایجاد شده است و در آن جدولی به نام shirts با دیتا موجود است طور ذیل استفاده می‌گردد:

```
>> USE DATABASE SCHOOL2;
```

```
>> CREATE VIEW V2 AS SELECT * FROM shirts;
```

این طرزالعمل نمایی را به نام V2 در دیتابیس مکتب 2 طوری ایجاد می‌کند که با به راه‌انداختن آن تمام ریکوردهای جدول پیراهن‌ها نشان داده می‌شود. نمای V2 چون بدون کدام شرطی ایجاد شده است معادل با طرزالعمل کیوری ... SELECT از جدول پیراهن‌ها است. شکل پایین تطبیق طرزالعمل‌ها و مثال یاد شده را نشان می‌دهد.

```

C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin\mysql.exe

mysql> USE DATABASE SCHOOL2;
ERROR 1049 (42000): Unknown database 'database'
mysql> SHOW TABLES;
+-----+
| Tables_in_school2 |
+-----+
| assignment        |
| myset             |
| shirts            |
| t2                 |
| tb1               |
+-----+
5 rows in set (0.00 sec)

mysql> CREATE VIEW V2 AS SELECT * FROM shirts;
Query OK, 0 rows affected (0.08 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_school2 |
+-----+
| assignment        |
| myset             |
| shirts            |
| t2                 |
| tb1               |
| v2                |
+-----+
6 rows in set (0.00 sec)

mysql> SELECT * FROM V2;
+-----+-----+
| name      | size |
+-----+-----+
| dress shirt | large
| t-shirt   | medium
| polo shirt | small
+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT * FROM shirts;
+-----+-----+
| name      | size |
+-----+-----+
| dress shirt | large
| t-shirt   | medium
| polo shirt | small
+-----+-----+
3 rows in set (0.00 sec)

mysql>

```

شکل ۸-۶ تطبیق طرزالعمل‌های ایجاد نما و استفاده آن از یک دیتابیس

طوری که در شکل بالا دیده می‌شود، یک نما به نام V2 در دیتابیس مکتب 2 ایجاد شد. با استفاده از طرزالعمل SHOW TABLES نام نمای جدید در فهرست شامل شده است. دو طرزالعمل آخری در شکل بالا معادل‌اند. به خاطری که طرزالعمل آخری در نمای V2 ذخیره شده است. این مثال‌ها استفاده از یک منبع دیتا (یک دیتابیس) را نشان می‌دهد. عین کار به خاطر ترکیب دیتا از منابع مختلف نیز صورت گرفته می‌تواند.

طرزالعمل پایین نمایی به نام V3 را طوری ایجاد می‌کند که در آن دیتا از دو منبع مختلف گرفته و با هم ترکیب شده را نشان می‌دهد:

```
>>CREATE VIEW V3 AS SELECT * FROM school2.shirts, school.assignment WHERE 1=1;
```

نمایی که به اساس طرزالعمل بالا ایجاد می‌شود، دیتا را از دو منبع مختلف (دو دیتابیس مکتب و مکتب 2) گرفته، با هم ترکیب کرده و نتیجه را نشان می‌دهد. شکل پایین تطبیق مؤفقانه طرزالعمل را نشان می‌دهد.

```

C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin\mysql.exe

mysql> CREATE VIEW V3 AS SELECT * FROM school2.shirts, school.assignment
-> WHERE 1=1;
Query OK, 0 rows affected (0.06 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_school |
+-----+
| assignment        |
| employee          |
| project           |
| v3                 |
+-----+
4 rows in set (0.00 sec)

mysql> DROP VIEW V3;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_school |
+-----+
| assignment        |
| employee          |
| project           |
+-----+
3 rows in set (0.00 sec)

mysql>

```

شکل ۹-۶ تطبیق مؤفقانه طرزالعمل ترکیب دیتا از دو منبع مختلف (دیتابیس‌های مختلف)

همان‌طوری که نما به حیث یک ساختمان در دیتابیس‌ها به یک نام ایجاد شده می‌تواند، در صورت ضرورت دوباره از سیستم پاک هم شده می‌تواند. در شکل بالا توسط طرزالعمل ... CREATE VIEW یک نما به نام V3 مؤفقانه ایجاد شد. طرزالعمل SHOW TABLES نمای جدید را در فهرست نشان داد. با طرزالعمل DROP VIEW ... نمای جدید دوباره پاک شده و با به راه‌انداختن طرزالعمل SHOW TABLES نشان می‌دهد که طرزالعمل پاک کردن نما مؤفقانه انجام شده است. پس دیده شد که دیتا از منابع مختلف با هم ترکیب شده و در یک محیط استفاده می‌شوند.

**نوت:** یک نکته مهم قابل تذکر در مورد مثال‌های بالا، کم بودن وقت نوشتن این کتاب است. مثال‌های بسیار ساده و از نظر مفهوم پیش پا افتاده به حیث نمونه به کار گرفته شده است. در چاپ‌های بعدی کتاب ان شاء الله مثال‌های جامع‌تری استفاده خواهند شد تا در امر رساندن مطالب رُل بارزتری داشته باشند.

## ۶.۵ بازیافتن معلومات (Information Retrieval)

بازیافتن معلومات یکی از موضوعات قابل اهمیت در بخش‌های مختلف زندگی انسان‌ها به شمار می‌رود. معلومات همان سرمایه اصلی انسان‌ها بوده که به اساس آن معاملات، کارها، تصمیم‌ها، تحقیق، تجارت و غیره امور و کارها انجام می‌شوند. در ساحة کمپیوترساینس وسایل و پروگرام‌هایی که به خاطر ذخیره دیتا و دوباره استفاده کردن و تبدیل کردن دیتا به معلومات استفاده می‌شوند، حائز اهمیت بوده و از موضوعات قابل بحث به شمار می‌روند. با انکشاف کمپیوترها و سیستم‌های معلوماتی در چند دهه اخیر، تحلیل و تنظیم معلومات و تبدیل انواع مختلف معلومات از کاپی‌های سخت به کاپی‌های نرم بیش‌تر مورد توجه قرار گرفت. به خاطر بازیافتن معلومات ذخیره شده با فارمت‌های دیجیتل و دوباره ترکیب کردن بخش‌های دیتا مهارت‌های تخیلی ضرورت است؛ یعنی استفاده کنندگان سیستم‌های معلوماتی باید با تکنالوژی‌های روز بلدیت داشته باشند تا معلومات را از منابع مختلف گرفته و بازیابی کنند.

در صورت درست تنظیم نکردن معلومات مشکلاتی از قبیل عدم دسترسی به معلومات موجود، تکرار استفاده از بخش‌های معلومات، استفاده نادرست (به شکل دانسته و یا به شکل ندانسته) بروز کرده می‌تواند. با انکشاف کمپیوترها با سرعت بالا توقع می‌رفت که تمام مشکلات در قسمت بازیافتن معلومات حل خواهد شد. این مسئله تا اندازه‌یی درست واقع شد ولی در کل با درست تنظیم نکردن دیتا و نداشتن یک سیستم کامل و کارآمد بازیافتن معلومات، توقعاتی که بود به شکل کامل، نتایج مثبتی در پی نداشت.

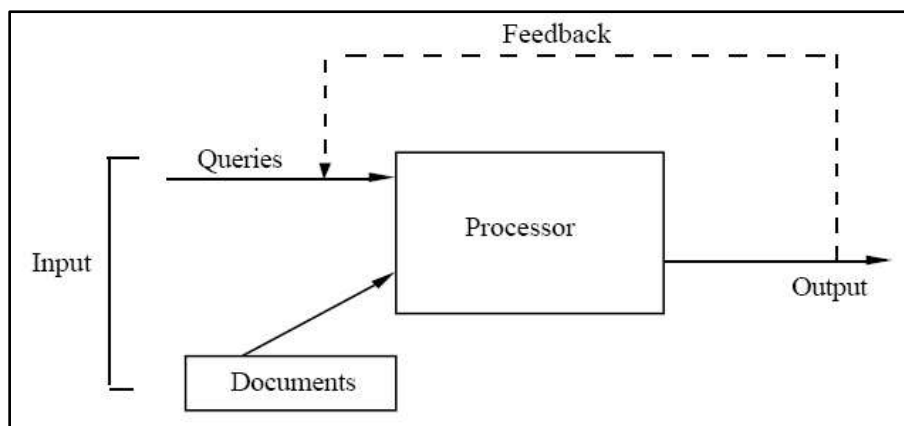
دیتا روی وب با حجم بسیار بالا موجود بوده و روز به روز به آن افزوده می‌شود. وقتی با معلومات روی وب و دیتای آنلاین فکر می‌شود، سیستم‌های استخراج و پیدا کردن معلومات باید با تریلیون‌ها واحد دیتا کار نمایند تا پروسه‌های بازیافتن معلومات، قناعت بخش و مفید واقع شوند. این مثال به خاطر اهمیت موضوع ارائه گردید. البته دست‌آوردهای بسیار در موارد یاد شده موجود بوده و پروژه‌ها روی استفاده از فارمت‌هایی که با دیتای بزرگ (Big Data) کار می‌نمایند و استفاده از الگورتم‌های مناسب را در این عرصه به کار می‌برند، موجوداند که خارج از بحث در این کتاب است.

### ۶.۵.۱ یک سیستم بازیافتن معلومات

سیستم‌ها به خاطر بازیافتن معلومات طرح و استفاده شده‌اند. درمجموع سیستم‌های بازیافتن معلومات دارای سه بخش ذیل‌اند:

- داخل کردن یا گرفتن معلومات (ریکورد‌های دیتا) توسط پروسیسور؛
- پروسس کردن معلومات؛

- خروجی معلومات (به اساس شرایط داده شده).  
شکل پایین بخش‌های یک سیستم بازیافتن معلومات را نشان می‌دهد.



شکل ۶-۱۰ بخش‌های تشکیل‌دهنده یک سیستم بازیافتن معلومات

طوری که در شکل بالا دیده می‌شود، یک سیستم بازیافتن معلومات به کمک یک پروسیسور که در مرکز دیگرام واقع شده است، تنظیم می‌گردد. پروسه‌هایی که بالای معلومات انجام می‌شوند، شاید بعضی ساختارها را به طور مؤقت ایجاد نمایند و معلومات را در آن دسته‌بندی کنند. تطبیق شرایطی که در کیوری‌های دخولی به خاطر بازیافتن و فیلترکردن معلومات وضع شده‌اند نیز در این قسمت صورت می‌گیرد.

معلومات اولیه‌یی که در سیستم به حیث دخولی (Input) استفاده می‌شوند عبارت از سندهای متنی (Text Documents) و شرایط وضع‌شده قابل تطبیق بالای معلومات داخل سندها است؛ یعنی متنی‌هایی که در سندها قابل پروسس‌اند یک بخش از دخولی به شمار رفته و کیوری‌ها (درخواست‌های معلومات) با در نظر گرفتن شرایط و موضوعات مشخص آن‌ها بخش دوم دخولی به حساب می‌آیند. دخولی می‌تواند از منابع مختلف در سیستم‌های محلی، عمومی و یا هم منابع آنلاین باشند. کیوری‌ها نیز می‌توانند از ترمینل‌های مختلف به شکل محلی، عمومی و یا هم آنلاین استفاده شوند.

بخش خروجی (Output) عبارت از همان معلومات بازیافتی‌اند که به درخواست استفاده‌کننده توسط سیستم بازیافتن معلومات نشان داده می‌شود. اگر معلومات بازیافته در خروجی به اساس ضرورت استفاده‌کننده نباشد، امکان دوباره خواستن معلومات موجود بوده یعنی فیدبک استفاده‌کننده می‌تواند کیفیت نتیجه خروجی را بهتر بسازد. در صورت عدم استفاده از شرایط مشخص در کیوری، طبعاً تمام معلوماتی که در سندهای دخولی‌اند به حیث معلومات خروجی (بازیافتی) ارائه خواهند شد.

## ۶.۶ بازیافتن معلومات و یا بازیافتن دیتا

معلومات با دیتا فرق دارد. معلومات زمانی که تحلیل و تجزیه شود به دیتا تبدیل می‌شود. به طور مثال در یک سناریو، شخصی به نام احمد در مکتبی واقع در شهر کابل به نام لیسه انصاری درس می‌خواند. احمد یک خانواده دارد که در آن با دو برادر، یک خواهر و والدینش زندگی می‌کند. ... احمد به کورس لسان انگلیسی

می‌رود. معلومات در مورد احمد به شکل یک فایل در کامپیوتر ذخیره است. همین معلومات اگر تحلیل و تجزیه شود و به شکل دیتای دوبعدی در جدول‌ها تنظیم شود، به نام دیتا یاد می‌شود.

به طور مثال یکی از جدول‌ها شاید معلومات مکتب احمد را داشته باشد. جدول مکتب به هیچ صورت معلومات مکمل در مورد احمد را که شامل سناریوی بالا است، در خود ندارد؛ و یا هم جدول دیگری که دیتای اعضای خانواده احمد در آن ذخیره شده باشد، معلومات مکمل سناریوی بالا را تمثیل نمی‌کند. ولی تحلیل و تنظیم معلومات سناریوی بالا در یک دیتابیس مزیت‌های دیگری را با خود دارد.

عناصر دیتا دوباره با هم یک‌جا شده و معلومات را تشکیل می‌دهد؛ یعنی به همان شکلی که معلومات در مورد یک موضوع به تحلیل گرفته شده و در دیتابیزی به شکل اجزای دیتا ذخیره می‌شود، به همان شکل از ترکیب دوباره دیتای معلومات به دست می‌آید. در نتیجه بازیافتن معلومات عبارت از کار کردن با دیتای متنی (سناریو در مثال بالا) است. بازیافتن دیتا عبارت از ترکیب دوباره عناصر دیتا و تهیه معلومات از آن است. پس گفته می‌شود معلوماتی که به شکل دخولی به سیستم بازیافتن معلومات داخل می‌شود، امکان دارد سندهای متنی باشند و یا هم عناصر دیتا از یک دیتابیس باشند. دقت در معلومات خروجی مستقیماً متأثر از این دو حالت می‌شود. شکل پایین موارد مشخصی را به اساس یافته‌ها و تجارب بین هر دو حالت به صورت خلاصه نشان می‌دهد.

Data Retrieval (DR)   Information Retrieval (IR)		
Matching	Exact match	Partial match, best match
Inference	Deduction	Induction
Model	Deterministic	Probabilistic
Classification	Monothetic	Polythetic
Query language	Artificial	Natural
Query specification	Complete	Incomplete
Items wanted	Matching	Relevant
Error response	Sensitive	Insensitive

شکل ۱۱-۶ مقایسه دو طریقه (اصطلاح) بازیافتن معلومات و بازیافتن دیتا

به خاطر حفظ مفاهیم اصلی یافته‌ها، جدول در شکل بالا به لسان انگلیسی گرفته شده از مرجع اصلی آن ارائه شده است. طوری که دیده می‌شود بعضی از تفاوت‌ها قابل بحث بوده که در ذیل به صورت مختصر به آن‌ها پرداخته شده است:

- پیدا کردن نمونه‌های هم‌خوان (Matching)

در استفاده از دیتابیس‌ها نمونه‌های دیتا دقیق می‌باشند در حالی که در بازیافتن معلومات از سندها، دقت در پیدا کردن نمونه‌های هم‌خوان کم‌تر است. این به خاطری است که در زمان تحلیل دیتا و

ذخیره آن به جدول‌های دیتابیس در زمان داخل کردن دیتا به سیستم صورت می‌گیرد. تصمیم در زمان بازیافتن معلومات بدون کدام تغییری بالای دیتا تطبیق شده و نتیجه ارائه می‌شود.

- نتیجه‌گیری (Inference)  
نتیجه‌گیری و استنباط از بازیافتن معلومات و دیتا بازهم با تفاوت از هم انجام می‌شود. زمانی که از دیتابیس استفاده می‌شود نتیجه آن از دیتای موجود کاسته می‌شود در حالی که با استفاده از معلومات متنی نتیجه‌ی با انگیزش بالا ارائه می‌شود.
- نمونه (Model)  
نمونه یا مدل دیتا زمانی که از دیتابیس استخراج می‌گردد به شکل قطعی بوده، در حالی که نمونه نشان داده شده به اساس بازیافتن سندها، معلومات احتمالی بوده؛ یعنی ارقام دیتا از دیتابیس طوری که هستند، در نتیجه ارائه می‌شوند. ولی معلومات ذکر شده در فایل‌های متنی قطعی نبوده و الگورتم‌های تطبیقی بر آن‌ها نتایج احتمالی را ارائه می‌کنند.
- طبقه‌بندی نتایج (Classification)  
طبقه‌بندی نتایج کیوری که بالای دیتابیس انجام شود، یک نتیجه داشته و عین مسئله در اجرای کیوری بالای معلومات متنی، امکان دارد چندین نتیجه مختلف را نشان دهد.
- لسان کیوری (Query Language)  
تفاوت دیگر در قسمت استفاده از لسان کیوری است. در دیتابیس‌ها زمانی که اجزای دیتا استفاده می‌شوند، لسان کیوری محدود به نحوه نوشتن سیستم مدیریت دیتابیس مربوطه است. در بازیافتن معلومات از منابع متنی محدودیت شکلی لسان کیوری موجود نبوده و خواستن معلومات بیش‌تر به لسان تکلم (جملات لسان تکلم) متکی است.
- مشخص ساختن خواسته‌ها در کیوری (Query Specification)  
به خاطر استخراج معلومات از دیتابیس، طرز‌العمل کیوری باید مکمل بوده و تمام شرایط آن مشخص شده باشد. در قسمت بازیافتن معلومات از منابع غیر دیتابیس مشخص کردن تمام جزئیات شرط نبوده، انعطاف بیش‌تر در اجرای طرز‌العمل و نشان کرده نتیجه موجود است.
- چیزهای خواسته‌شده (Items Wanted)  
عناوین چیزهایی که در کیوری دیتابیس خواسته می‌شود باید در جدول‌های دیتابیس موجود باشند. در صورت اشتباه یک کرکتر در یک نام، شاید کیوری نتیجه مطلوب را ارائه ننماید. در سیستم‌های بازیافتن معلومات این مسئله نیز با ارتباط موضوع خواسته‌شده و محتوای معلومات، حل گردیده و نتیجه نشان داده می‌شود؛ یعنی قرابت معلومات خواسته‌شده در یک کیوری با معلومات موجود کفایت می‌نماید تا نتیجه کیوری با معلومات نشان داده شود.
- پاسخ غلطی (Error Response)



زمانی که در طرزالعمل‌های بازیافتن دیتا و یا معلومات اشتباهی موجود باشد، سیستم‌های مدیریت دیتابیس پاسخ روشن می‌دهند در حالی که پاسخ سیستم‌هایی که با دیتای متنی کار می‌نمایند، بیش‌تر مشهود نیست.

طوری که در یافته‌های بالا دیده می‌شود، در بعضی موارد استفاده از دیتابیس‌ها برای بازیافتن عناصر دیتا مؤثریت بیش‌تر دارد؛ و در بعضی موارد نیز به کارگیری بازیافتن معلومات از منابع متنی بیش‌تر مؤثر است.

## ۶.۷ کیوری‌های ساختاری روی وب (Structured Queries over the Web)

معلومات بسیار زیاد در منابعی که از طریق اینترنت و وب قابل دسترس‌اند، موجود است. به طور مثال اگر در مورد عنوانی در صفحات وب جست‌وجو شود، در آن واحد به ده‌ها هزار و حتی صدها هزار صفحات وب فهرست می‌شوند. سیستم‌های جست‌وجوی معلومات که در اصطلاح به نام Search Engines یاد می‌شوند، زمینه‌ساز یافتن انواع مختلف معلومات روی وب می‌باشند. مشهورترین آن‌ها عبارت از Google است. هر نوع معلوماتی که به صورت عمومی ضرورت باشد از طریق صفحه گوگل قابل جست‌وجو بوده و پیدا کردن نتایج مطلوب در مورد موضوعی ممکن است. معلوماتی که به این طریق از آدرس‌های عمومی به دست می‌آیند، معمولاً با در نظر داشت کلمات کلیدی و اندکس‌های دیتا صورت می‌گیرد؛ یعنی کم‌تر به عمق دیتا و دیتابیس‌ها رجوع می‌شود.

طرزالعمل‌های جست‌وجوی معلومات روی وب امکان دارد معلومات را با شکل ساختاری منظم، پروسس نمایند و یا هم از طرزالعمل‌های جست‌وجو، معلومات غیر ساختاری را مورد پروسس قرار دهند. در حالت اول دیتا در دیتابیس‌ها به اساس مدل‌های تحلیل دیتا ذخیره بوده و نتایج معمولاً با دقت بالا استخراج می‌شوند. به طور مثال معلومات پروازهای یک شرکت هوایی در دیتابیسی تنظیم شده باشد، به خاطر خریدن تکت معلومات در مورد پروازهای مشخص از یک شهر به شهر دیگری در یک تاریخ و ساعت معین از دیتابیس کیوری شده و دیده می‌شوند. در چنین حالتی دیتای دقیق و قابل دسترس از قبل در دیتابیس مربوطه، ذخیره شده و در اختیار استفاده‌کنندگان قرار داده می‌شود. در حالت دوم در اکثر موارد عملیة جست‌وجوی دیتا تا اندازه‌ی سرسری صورت گرفته و به عمق معلومات ذخیره‌شده، پرداخته نمی‌شود. به خاطر کار بهتر با معلوماتی که تمام بخش‌های آن قابل تحلیل و در دسترس نیست، مدل‌هایی در نظر گرفته شده که در ذیل به آن‌ها پرداخته می‌شود.

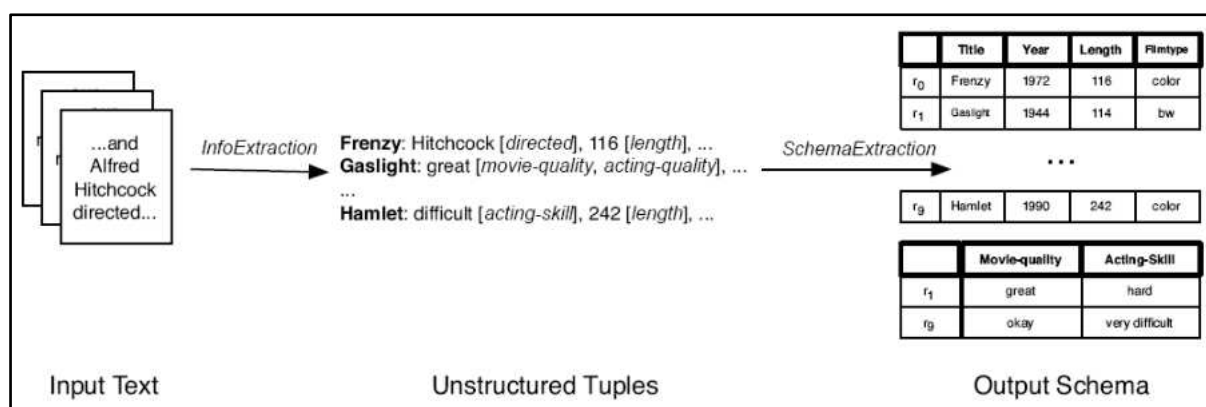
### ۶.۷.۱ مدل استخراج سکیم (The Schema Extraction Model)

مدل استخراج سکیم عبارت از جمع‌بندی معلوماتی است که در حوالی یک وب‌سایت از طرف استفاده‌کنندگان اضافه شده باشد. در این نوع مدل دیتای اصلی سیستم یا یک وب‌سایت ضرورت نیست. معمولاً به خاطر اهداف مدیریتی و پالیسی‌های اداری دیتای اصلی اکثر وب‌سایت‌ها به دسترس عموم قرار داده نمی‌شود. هدف از استفاده مدل استخراج سکیم نیز دسترسی به دیتای اصلی و محرمانه وب‌سایت‌ها و سیستم‌ها نیست. کوشش برای دسترسی به دیتای اصلی یک وب‌سایت در صورت نداشتن مجوز قانونی یک عمل جرمی پنداشته می‌شود.



با استفاده از مدل استخراج سکیم از یک سیستم معلوماتی چه وبسایت باشد و یا هم دیتابیس، معلومات جانبی به شکل قانونی گرفته شده و در فارمت دیتای ترکیبی (Structured Data) تنظیم می‌شود. بعد از استخراج دیتا و تنظیم آن به حیث یک منبع دیتا به شکل قانونی توسط یک سیستم پروسس دیتا مورد استفاده قرار گرفته می‌شود. جهت وضاحت بیش‌تر موضوع یک مثال ارائه می‌شود. وبسایت یک هتل، معلومات در مورد کل امکانات و خدمات هتل را دارا است. مشتریان هتل اختیار دسترسی به وبسایت هتل را داشته و به حیث استفاده‌کنندگان صفحه، نظریات و فیدبک‌شان را در موارد مختلف هتل ارائه می‌دارند.

نظریات مشتریان امکان دارد در خود وبسایت هتل انعکاس داده شوند و یا هم در صفحات دیگری مانند صفحات اجتماعی و خبری نشر شده باشند. وبسایت‌ها و صفحاتی که معلومات در مورد نظریات مشتریان را نشر کرده‌اند، می‌شوند به حیث دیتای دخیلی به خاطر استخراج سکیم استفاده شوند. جدول‌هایی با مشخصه‌های مناسب در سکیم ترتیب و معلومات گرفته شده در آن‌ها درج می‌شوند. بعد از گرفتن معلومات و تنظیم آن در ساختمان‌های قابل استفاده، می‌توان از آن به حیث یک مدلی که دیتای قانونی در مورد هتل یاد شده دارد، به کار گرفته شود. شکل پایین، جریان پروسه‌های استخراج سکیم را از معلومات متنی نشان می‌دهد.



شکل ۶-۱۲ پروسه‌های استخراج سکیم از معلومات داخل شده به شکل متن

در شکل بالا به خاطر مدل استخراج سکیم معلومات گرفته شده از منابع آنلاین به شکل متن دخیلی به تحلیل گرفته می‌شود. در مرحله بعدی اجزای متن به فارمت‌های از قبل تعریف‌شده تحلیل و تقسیم‌بندی می‌گردند. در مرحله اخیر دیتای تنظیم‌شده به ساختمان‌های مشابه به جدول‌های دیتابیس انتقال و ذخیره می‌گردد. از مجموعه ساختمان‌های ذخیره دیتا همراه با رابطه‌های لازم که توسط انکشاف‌دهنده سیستم نظر به ضرورت ایجادشده، سکیم استخراج گردیده و ریکوردهای به دست آمده از معلومات لازم را در بر می‌داشته باشد.

ایجاد مدل‌های دیتا به این شکل یکی از راه‌های استفاده از معلومات صفحات وب به شکل قانونی است. استفاده از مدل استخراج سکیم مانند هر سیستم دیگر دارای نواقص و مزایایی است. به دست آوردن معلومات بدون مصرف، یکی از بزرگ‌ترین مزایای این مدل به شمار می‌رود. به عین شکل دوباره به اشتراک گذاشتن و استفاده معلومات به دست آمده از طریق منابع آنلاین نیز از مزایایی مهم این مدل است.

در مقابل عدم دقت در انجام پروسه دیتا، ممکن این روش به حیث یکی از نقاط ضعف در این مدل به حساب بیاید. چون منبع معلومات استفاده کنندگان عام بوده و در مورد ارائه نظریات، آن‌ها آزاد می‌باشند. بناءً تکیه به معلوماتی که از منابع غیرمطمئن تهیه می‌شوند، یکی از چالش‌های عمده مدل استخراج سکیم‌ها به شمار می‌رود.

## ۶.۷.۲ مدل کیوری متنی (The Text Query Model)

با استفاده از این مدل یک نوع دیگر دسترسی به معلومات روی صفحات وب صورت می‌گیرد. در مدل قبلی دسترسی به معلومات (مدل استخراج کردن سکیم) کار بالای دیتای استفاده کنندگان صورت می‌گرفت؛ یعنی معلومات اصلی از نظریات و فیدبک استفاده کنندگان وبسایت‌ها به طریق‌های مختلف جمع‌آوری شده و به ساختمان‌های منظم ذخیره می‌شد. در این مدل، طوری که از نام آن پیداست، کیوری‌های متنی تشکیل داده می‌شوند، طوری که همان کیوری‌ها بالای معلومات صفحات مشخص شده وب تطبیق گردیده و نتایج حاصله به دست می‌آیند؛ یعنی این کار مترادف با استفاده کنندگان عادی وبسایت‌ها بوده و استفاده کنندگان وبسایت با مراجعه به آن‌ها معلومات خود را از طریق دیدن، خواندن و مطالعه به دست می‌آورند و مدل کیوری متنی به خاطر به دست آوردن معلومات از وبسایت‌ها، کیوری‌های را استفاده می‌کند.

به خاطر وضاحت این مسئله یک مثال در نظر گرفته شده است. البته این مثال با روحیه فرهنگ افغانستان مساعد نبوده، فقط از سایت معلوماتی این مدل با حفظ توضیحات پروژه تحقیقی در این مورد گرفته شده است. فکر کنید یک شخص می‌خواهد تا معلومات کنسرت‌های یک گروه موسیقی را که در نزدیک شهر بود و باش آن شخص برپا می‌شوند را بیابد. در مثال شهر بود و باش شخص به نام Seattle که یکی از شهرهای کشور امریکا است، گفته شده است. تقسیم اوقات پروگرام‌های کنسرت گروه موسیقی در وبسایت آن‌ها موجود است. در یک کیوری همین شخص به حیث استفاده کننده می‌خواهد تا معلومات ذیل را دریافت کند:

- به دست آوردن معلومات شهر کنسرت و تاریخ کنسرت گروه موسیقی از وبسایت مورد نظر.
  - نشان کرده نام شهری که شخص در آن بود و باش دارد (در این جا شهر سیاتل).
  - محاسبه در مورد فاصله بین شخص مورد نظر و شهر که محل برگزاری کنسرت است که در یک فاصله مناسب قرار داشته باشد (در این مثال 100 میل در نظر گرفته شده است).
  - دیدن کنسرت‌های جدید (کنسرت‌هایی که تاریخ‌شان بعد از تاریخ کیوری باشند).
- شکل کیوری که با شرایط بالا، معلومات را از وبسایت مربوطه گرفته و با در نظر داشت شرایط استفاده کننده به دست می‌دهد، طور ذیل خواهد بود:

```
SELECT BandCity, BandDate
FROM (http://thebandsite.com/**,
["to appear in <string> on <data>", BandCity, BandDate])
WHERE BandDate > TIMESTAMP() AND
GeographicDistance(BandCity, "Seattle") =< 100
```

در مدل کیوری متنی استفاده‌کننده باید تمام بخش‌های ضروری کیوری را در کود آن داشته باشد تا نتیجه طرز‌العمل کامل شود. در قسمت FROM آدرس وبسایت‌هایی که دیتا از آن خوانده می‌شود، اضافه می‌گردد؛ یعنی مانند کیوری دیتابیس این کلمه ریفرنس است به منبع دیتا که در عملیه استفاده می‌شود. در این مثال معلومات از وبسایت گروپ موسیقی ضرورت است.

بناءً از آدرس الکترونیکی وبسایت متذکره استفاده شده است. در صورتی که اضافه از یک وبسایت به خاطر گرفتن معلومات ضرورت باشد، تمام وبسایت‌های مورد ضرورت در همین قسمت کیوری، آدرس داده می‌شوند. قبل از قسمت FROM فیلدهایی که خروجی کیوری را تشکیل می‌دهند، فهرست گردیده‌اند. در مثال بالا دو فیلد شهر و تاریخ برگذاری کنسرت از وبسایت متذکره استخراج می‌شوند.

شرایط اضافه شده در قسمت WHERE نتیجه معلومات گرفته شده از وبسایت را فیلتر می‌کند. در این قسمت دو شرط استفاده شده است. شرط اول تاریخ کنسرت را چک می‌کند که بعد از تاریخ اجرای کیوری باشد. در صورت موجود بودن معلومات کنسرت‌های قبلی، ریکوردهای مربوطه تاریخ‌های گذشته را نشان نمی‌دهد. به عین شکل قسمت دوم شرط همان اندازه فاصله بین شهر بود و باش استفاده‌کننده و شهر برگذاری کنسرت است. کنسرت‌هایی که در فواصل بیش‌تر از 100 میل برگذار می‌گردند از معلومات آن‌ها نیز صرف نظر شده و در نتیجه کیوری نشان داده نمی‌شود.

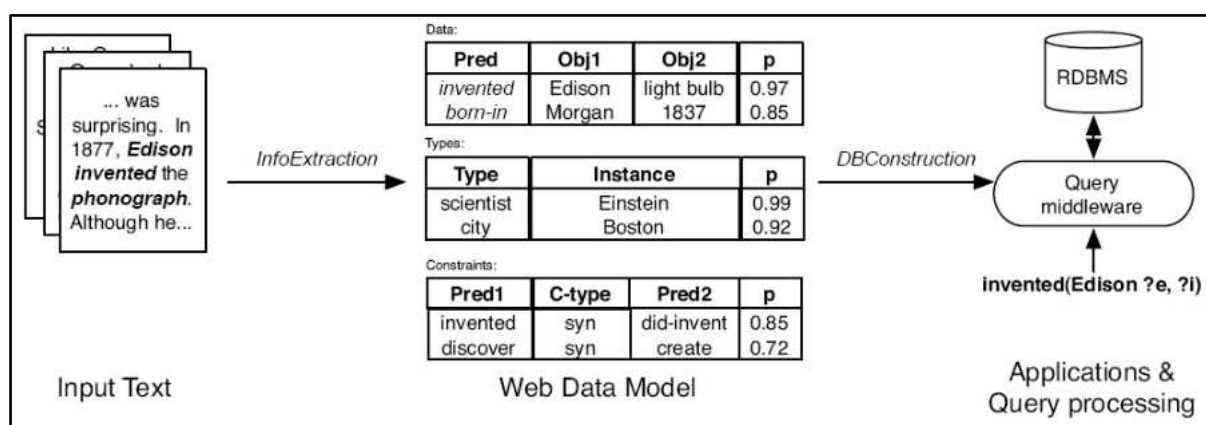
در مدل کیوری متنی معلومات از قبل ضرورت به ذخیره ندارد. چیزی که ضرورت است آن عبارت از کود کیوری است؛ یعنی با تحلیل اوضاع، گرفتن دیتا از منابع (وبسایت‌های فعال) و تطبیق شرایط مربوط به یک موضوع، از مدل کیوری متنی استفاده می‌شود. طبعاً به خاطر اطمینان از کارکردن درست مدل، کود آن باید به صورت امتحانی چک شود، ولی به خاطر استفاده به‌جا و مناسب، مدل کیوری متنی باید در زمان ضرورت آن اجراء شود. در مثال بالا اگر کیوری یک هفته قبل از ضرورت به راه انداخته شده باشد، کنسرت‌های یک هفته قبل نیز در فهرست نشان داده خواهند شد که این کار نتیجه مطلوب مدل کیوری را نادرست خواهد ساخت. چون در کیوری از تابع TIMESTAMP استفاده شده است و این تابع تاریخ و زمان اجراءشدن کیوری را به حیث شرط در نظر می‌گیرد. بناءً به خاطر درست بودن نتایج کیوری باید در زمان استفاده به راه انداخته شود تا از نمایش دیتای نادرست جلوگیری صورت گیرد.

### ۶.۷.۳ مدل استخراج دیتا برای دیتابیس (The Extraction Database Model)

مدل استخراج کردن (دیتا) برای دیتابیس یک طریقهٔ وسطی بین دو مدل قبلی به نام‌های مدل استخراج کردن سکیم و مدل کیوری متنی است. مشابهت نام این مدل با مدل استخراج سکیم در استفاده از اصطلاح استخراج کردن است. در مدل اول یک سکیم یا دسته‌یی از معلومات به اساس طرز‌العمل‌های جست‌وجوی معلومات در یک وبسایت توسط استفاده‌کنندگان آن تشکیل می‌شد، یعنی سکیم استخراج می‌شد. در این مدل (مدل استخراج کردن برای دیتابیس) معلومات از وبسایت‌ها استخراج شده و قسملاً تحلیل می‌شوند. بعد از آن به اساس معلومات به دست آمده، دیتابیس ارتباط (Relational Database) در روشنی قواعد ارتباط از قبیل موجودیت جدول‌های معیاری، ارتباط بین جدول‌ها و غیره ایجاد می‌شود. بعد از ایجادشدن دیتابیس

معلومات به دست آمده در جدول‌های دیتابیس ذخیره می‌شود؛ و در نهایت کیوری‌های استفاده از دیتای به دست آمده بالای دیتابیس جدید اجراء شده و نتایج مورد ضرورت به دست می‌آیند. طوری که گفته شد این مدل در بین دو مدل قبلی کار می‌کند. به این معنی که در مدل اول (مدل استخراج کردن سکیم) معلومات از طریقه‌های استفاده‌کنندگان جمع، نظریات و فیدبک‌شان گرفته شده و مورد استفاده قرار می‌گیرد.

در مدل دوم (مدل کیوری متنی) معلومات با استفاده از امکانات کیوری‌ها به صورت مستقیم از وب‌سایت‌های مورد نظر گرفته شده و استفاده می‌شود. مدل سه (مدل استخراج کردن برای دیتابیس) از هر دو طریقه کار مدل‌های قبلی تا اندازه‌ی استفاده کرده، معلومات از آدرس استفاده‌کنندگان را و نیز معلومات مستقیم از وب‌سایت‌ها را گرفته با هم ترکیب نموده و آن را استفاده می‌کند. شکل پایین طریقه کار مدل استخراج دیتا برای دیتابیس را نشان می‌دهد.



شکل ۶-۱۳ طریقه کار مدل استخراج دیتا برای دیتابیس (The Extraction Database Model)

طوری که در شکل بالا دیده می‌شود، به خاطر به دست آوردن دیتا، معلومات به دست آمده به کمک سیستم‌های مختلف استخراج کردن دیتا مورد استفاده قرار می‌گیرد؛ یعنی در سمت چپ تصویر متن دخولی امکان دارد از معلومات استفاده‌کنندگان گرفته شده باشد و یا هم به صورت مستقیم از محتوای وب‌سایت‌ها به کمک کیوری‌های متنی گرفته شده باشند. معلومات استخراج‌شده از منابع مختلف در قدم بعدی به خاطر ایجاد جدول‌های تحت عنوان دیتای مدل وب (Web Data Model) به کار گرفته می‌شوند.

جدول‌های تهیه شده به اساس معلومات اولیه استخراج‌شده از منابع آنلاین، معمولاً به فارمت جدول‌های قابل استفاده در دیتابیس‌های ارتباط نیست. به خاطر فارمت آن‌ها در این مرحله کیوری‌هایی به نام کیوری‌های وسطی (Middleware) تطبیق می‌شوند. نتیجه کیوری‌ها به حیث جدول‌های معیاری سیستم‌های ارتباط تهیه شده و به دیتابیس مورد نظر ذخیره می‌گردند. در نهایت تطبیق پروسه‌های بالا، دیتابیس نهایی برای استفاده دیتای استخراج‌شده از منابع آنلاین به حیث یک سیستم به کار گرفته می‌شود. مسئله‌ی که نباید فراموش شود این است که دیتای به دست آمده از منابع آنلاین در هر سه طریقه احتمالی است؛ یعنی نتایج به دست آمده صد فی صد دقیق نبوده، فیصدی دقت دیتا بستگی مستقیم به دقت کار انجام‌شده در طریقه‌های به دست آوردن دیتا می‌داشته باشد.



در این فصل به مسایل تخنیکی در مورد کارکردن با دیتای روی وب پرداخته شده است. در ابتداء موضوع ترکیب دیتا توضیح و سطوح یا لیول‌های ترکیب دیتا مشخص شدند. بعضی مفاهیم کلیدی موضوع ترکیب دیتا از قبیل تهیه معلومات احصائیوی، پیوندهای دقیق و احتمالی بین ریکوردهای دیتا، ارزیابی از کیفیت ترکیب دیتا و سناریوهای ترکیب دیتا با ذکر مثال ارائه شد. در ادامه دو طریقه ترکیب دیتا شامل ترکیب فزیکی، ترکیب مجازی و موارد استفاده از آن‌ها توضیح گردیده است. اجراء کیوری بالای دیتای ترکیب‌شده از جمله طرزالعمل‌های کار با دیتای (DML) با استفاده از یک شکل و دیتابیس‌های مختلف نشان داده شده است. ایجاد و استفاده از نماها (Views) به سطح محلی (Local) و عمومی (Global) با استفاده از مثال‌های آن بیان شد.

در ادامه فصل به عناوین بازیافتن معلومات از منابع اصلی آن‌ها اشاره شده است. تهیه معلومات از فایل‌های بزرگ متنی با استفاده از الگورتم‌ها به خاطر بازیافتن معلومات توضیح شد. سیستم‌های بازیافتن معلومات از مقادیر بزرگ معلومات متنی به صورت کلی نظر اندازی شده و توسط مثالی نشان داده شده است. بحث بازیافتن معلومات تحت عنوان مستقلى با بازیافتن دیتا به شکل مقایسوی به بحث گرفته شد. تفاوت‌های عمده بین هر دو حالت در یک جدول نشان داده شده و تفصیل لازم در موارد تفاوت‌ها به صورت نسبی توضیح داده شد.

کیوری‌های ساختاری روی وب از جمله عناوین مهم دیگر مربوط به دیتای روی وب به شمار رفته و توضیحات مورد نیاز در زمینه ارائه شده است. مدل‌ها به خاطر استخراج دیتا از صفحات وب موجود بوده و نظر به فعالیت‌شان به سه نوع دسته‌بندی می‌شوند. در قسمت نحوه کاری هر سه مدل توضیحات لازم ارائه شد. تفاوت‌ها و موارد استفاده از هر کدام این‌ها با میخانیکیت‌های کاری‌شان آذین‌بخش آخرین قسمت فصل می‌باشند.



## فعالیت های فصل ششم

۱. طرزالعمل عمومی ایجاد نما (View) را در سیکویل بنویسید.
۲. با استفاده از طرزالعمل SHOW DATABASES دیتابیس های موجود در سیستم را ببینید.
۳. با استفاده از طرزالعمل USE DATABASE ... یکی از دیتابیس های موجود در سیستم را فعال سازید.
۴. با استفاده از طرزالعمل SHOW TABLES فهرست جدول های موجود در دیتابیس فعال شده را ببینید.  
(در صورت موجود نبودن جدول ها با استفاده از طرزالعمل CREATE TABLE ... یک یا دو جدول را در دیتابیس ایجاد کنید.)
۵. با استفاده از طرزالعمل CREATE VIEW ... یک نما را به نام V1 ایجاد کنید که در آن معلومات یک جدول کیوری شده باشد.
۶. با استفاده از طرزالعمل SHOW TABLES فهرست جدول ها و نماها را ببینید تا اطمینان از ایجادشدن نما در فعالیت قبلی حاصل شود.
۷. با استفاده از طرزالعمل CREATE VIEW ... یک نما را به نام V2 ایجاد کنید که در آن معلومات اضافه از یک جدول در یک دیتابیس کیوری شده باشد.
۸. با استفاده از طرزالعمل SHOW TABLES FROM DatabaseName جدول های یکی از دیتابیس های دیگر موجود در سیستم را ببینید.
۹. با استفاده از طرزالعمل CREATE VIEW ... یک نما را به نام V3 ایجاد کنید که در آن معلومات اضافه از یک جدول و از دیتابیس های مختلف کیوری شده باشد.
۱۰. با استفاده از طرزالعمل CREATE VIEW ... یک نما را به نام V4 ایجاد کنید که در آن معلومات نمای ایجاد شده در فعالیت قبلی (V3) کیوری شده باشد.
۱۱. با استفاده از طرزالعمل CREATE VIEW ... یک نما را به نام V5 ایجاد کنید که در آن معلومات یک نما و یک جدول از یک دیتابیس با ذکر شرایطی کیوری شده باشد.
۱۲. با استفاده از طرزالعمل CREATE VIEW ... یک نما را به نام V6 ایجاد کنید که در آن معلومات یک نما و یک جدول از دیتابیس های مختلف با ذکر شرایطی کیوری شده باشد.
۱۳. با استفاده از طرزالعمل DROP VIEW ... آخرین نما را از سیستم پاک کنید.
۱۴. با استفاده از طرزالعمل SHOW TABLES فهرست جدول ها و نماها را ببینید تا اطمینان از پاک شدن آخرین نما حاصل شود.



۱. مفهوم ترکیب دیتا (Data Integration) را به صورت ساده بیان کنید.
۲. سطوح ترکیب دیتا (Data Integration Levels) کدام‌ها اند؟ مختصراً توضیح دهید.
۳. کوچک‌ترین واحد به خاطر ذخیره دیتا در سیستم‌های دیتابیس ارتباط (Relational Databases) به چه نام یاد می‌شود؟ مختصراً توضیح دهید.
۴. پیوند دقیق (Exact Linkage) را در بحث ترکیب دیتا (Data Integration) بیان کنید.
۵. پیوند احتمالی (Probabilistic Linkage) را در بحث ترکیب دیتا (Data Integration) بیان کنید.
۶. در طریقه پیوند واحدهای دیتا از منابع مختلف کدام اشتباهات متصوراند؟ نام بگیرید.
۷. ترکیب فزیک دیتا را مختصراً توضیح دهید.
۸. ترکیب مجازی دیتا را مختصراً توضیح دهید.
۹. ترکیب محلی دیتا (Local Integration) به چه مفهوم است؟
۱۰. ترکیب عمومی دیتا (Global Integration) به چه مفهوم است؟
۱۱. نما (View) را توضیح دهید.
۱۲. هدف استفاده از شرط  $WHERE\ 1 = 1$  در یک کیوری و نتیجه آن چیست؟
۱۳. یک سیستم بازیافتن معلومات (Information Retrieval) دارای کدام بخش‌ها است؟ شرح دهید.
۱۴. فرق بین معلومات و دیتا را با ذکر یک مثال جامع واضح سازید.
۱۵. مدل استخراج سکیم (The Schema Extraction Model) را به طور خلاصه بیان کنید.
۱۶. مدل کیوری متنی (The Text Query Model) را به طور خلاصه بیان کنید.
۱۷. مدل استخراج کردن دیتا برای دیتابیس (The Extraction Database Model) را به طور خلاصه بیان کنید.

## فصل هفتم

# انبارکردن دیتا (Data Warehousing) و بهره‌برداری از دیتا (Data Mining)



**هدف کلی:** محصلان با مفاهیم انبار دیتا (Data Warehouse) و بهره‌برداری از دیتا (Data Mining) آشنا شوند.

**اهداف آموزشی:** در پایان این فصل محصلان قادر خواهند شد تا:

۱. انبار کردن دیتا (Data Warehousing) را توضیح کنند.
۲. پروسه‌های دیزاین انبار دیتا را بفهمند.
۳. با ساختمان‌های انبار دیتا آشنا شوند.
۴. بهره‌برداری از دیتا (Data Mining) را توضیح داده و هدف استفاده از آن را بفهمند.
۵. وظایف و ساختار دیتاماینینگ را توضیح کرده بتوانند.
۶. تفاوت بین انبار دیتا و دیتاماینینگ را بیان کنند.
۷. خصوصیات و فواید هر دو سیستم انبار دیتا و دیتاماینینگ را نام بگیرند.



دیتا بعد از تحلیل و ذخیره در منابع مختلف از قبیل دیتابیس‌ها، فایل‌های معلومات، صفحات وب و غیره توسط سیستم انبار دیتا (Data Warehousing) طوری تنظیم می‌شود که به آسانی قابل دسترسی باشد. انبار دیتا عبارت از به کارگرفتن تکنالوژی کمپیوترساینس و سیستم‌های معلوماتی بوده که توسط آن دیتا از منابع اصلی دیتابیس‌ها، فایل‌ها، صفحات وب و غیره گرفته شده و معلومات در مورد یک موضوع مشخص از آن‌ها استخراج می‌گردد. اگر وضاحت بیش‌تر داده شود، در استفاده از سیستم انبار دیتا عناصر دیتا و معلومات به شکل فیزیکی در این سیستم‌ها ذخیره نشده بلکه دیتا و معلومات در موقعیت‌های اصلی خود است. انبار دیتا استفاده‌کننده را کمک می‌کند تا معلومات مورد نظر خود را جست‌وجو کرده و به کمک این سیستم از منابع اصلی آن استخراج کند. توضیحات بیش‌تر در قسمت انبار دیتا، پروسه‌های دیزاین آن و ساختمان‌های استاندارد انبار دیتا در این فصل توضیح داده شده است.

بهره‌برداری از دیتا که در انگلیسی آن را Data Mining می‌گویند، عبارت از به کارگیری قسمت‌های دیتای ذخیره‌شده در سیستم است. سیستمی که دیتا را ذخیره و قابل استفاده می‌سازد، می‌تواند یک دیتابیس ارتباط باشد، می‌تواند فایل‌هایی از معلومات باشند، می‌تواند صفحات وب به صورت آنلاین باشند و یا هم می‌تواند فایل‌های متنی معلومات باشد. در این فصل به توضیحات لازم در قسمت دیتامیننگ، هدف و وظایف آن و نیز در قسمت ساختار و طبقه‌بندی دیتامیننگ توضیحات لازم ارائه شده است. در ادامه فصل به مقایسه نسبی و تفاوت‌های عمده بین انبار دیتا (Data Warehousing) و بهره‌برداری از دیتا (Data Mining) پرداخته شده است. این فصل با خلاصه موضوعات توضیح شده و با سؤالات و فعالیت شاگردان پایان می‌یابد.

## ۲.۱ انبارکردن دیتا (Data Warehousing)

انبار دیتا که در اصطلاح انگلیسی آن را به نام Data Warehouse یاد می‌کنند، به اساس تعریف عالمی به نام William H. Inmon عبارت از مجموعه‌های عناصر دیتا بوده که به خاطر استفاده کردن در مسایل مختلف تصمیم‌گیری در یک موضوع به کار برده می‌شوند. این مجموعه‌های دیتا خصوصیات مشترک ذیل را دارا می‌باشند:

- متمایل بودن به موضوع (Subject Oriented): یک انبار دیتا به یک موضوع تمایل می‌داشته باشد. هر بخش دیتا باید در بر گیرنده معلومات در مورد یک موضوع مشخص باشد؛ یعنی در انبار دیتا معلومات در ارتباط به موضوعات مدیریت می‌شوند نه این‌که معلومات در رابطه به فعالیت‌های روزمره یک اداره در آن ذخیره و مدیریت شود. مثال‌های موضوعاتی که در سیستم‌های انبار دیتا استفاده می‌شوند، عبارت‌اند از دیتا در مورد محصولات، مشتریان، تمویل‌کنندگان، فروشات، خدمات و غیره. در یک انبار دیتا تمرکز بیش‌تر بالای مدل‌سازی و تحلیل دیتا به هدف تصمیم‌گیری است.
- کامل بودن (Integrated): دیتا و ارقامی که امکان دارد از منابع مختلف گرفته شده باشد، باید به طور یک پارچه، کامل و منسجم قابل استفاده باشد. یک انبار دیتا متشکل از دیتای ترکیب‌شده از منابع مختلف مانند دیتابیس‌های ارتباط، فایل‌های دیتا، معلومات از صفحات وب، انترنت، کتاب‌ها و غیره است.

- زمان‌های گوناگون (Time Variant): تمام دیتا شامل انبار دیتا به اساس دوره‌های زمانی مشخص قابل شناسایی است؛ یعنی هر قسمت دیتا دارای معلومات (میتادیتا) است که در آن مدت (زمان و تاریخ) مشخص، تنظیم‌شدن دیتا ثبت باشد. همین دوره ترتیب‌شدن دیتا (زمان و تاریخ آن) باید به خاطر شناسایی دیتا استفاده شده بتواند.
- پابرجا بودن (Non-Volatile): دیتا در انبار دیتا به صورت ثابت و پابرجا است. دیتا بیش‌تر به یک انبار دیتا اضافه می‌شود ولی هیچ دیتا نباید از انبار دیتا کاسته (دور) شود. این صفت انبار دیتا باعث اطمینان بیش‌تر برای بخش‌های مدیریتی می‌شود. در ادامه تعریف عمومی بالا، در انبارهای دیتا معلومات بخش‌های مختلف قابلیت ترتیب، ذخیره و دسترسی را در روشنایی با صفات ذکر شده دارد. دیتای قابل دسترس از انبارهای دیتا به هدف تصمیم‌گیری‌های مدیریتی قابل استفاده است.
- تعریف دیگری از انبار دیتا توسط آقای Ralph Kimball در کتابش به نام (The Data Warehouse Toolkit) با یک اندازه اختصار نظر به تعریف قبلی طور ذیل ارائه گردیده است:
- یک انبار دیتا عبارت از نسخه‌یی از دیتای اجرایی است که به خاطر کیوری و تحلیل کردن ساخته شده باشد.
- تعریف دومی از انبار دیتا به نسبت تعریف آقای Inmon ساده‌تر بوده ولی با آن هم مسئله‌یی را بیان می‌کند. با وسعت استفاده از کمپیوترها و وسایل الکترونیکی و تکیه بیش‌تر به منابع دیتای آنلاین، اهمیت ساحات مدیریت دیتا روز به روز بیش‌تر می‌شود. با در نظر داشت این موضوع، نظریه‌های مختلف در قسمت استفاده‌های عملی از تکنالوژی انبار دیتا به وجود آمده و تعریفات بالا را بیش‌تر غنی ساخته است. یک انبار دیتا ساختمان‌هایی را شامل می‌شود که در آن بر علاوه یوزردیتا، میتادیتا نیز انبار می‌شود. در انبار دیتا بر علاوه تنظیم ریکوردهای دیتا از منابع مختلف دیگر، دیتا از صفحات وب نیز قابل مدیریت بوده و در اختیار استفاده‌کنندگان قرار داده می‌شود. دیتای قابل دسترس برای استفاده‌کنندگان به شکل‌های چندبعدی، تاریخی و یک پارچه (گرفته شده از منابع مختلف و قابل استفاده مانند یک واحد ثابت) است.

### ۷.۱.۱ پروسه‌های دیزاین انبار دیتا (Data Warehouse Design Process)

یک انبار دیتا به سه طریقه دیزاین شده می‌تواند: طریقه اول بالا به پایین، طریقه دوم پایین به بالا و طریقه سوم ترکیبی از هر دو طریقه قبلی. این طریقه‌ها در اکثر موارد استفاده می‌شوند که در آن‌ها دیزاین‌ها به اساس تحلیل دیتا صورت می‌گیرند. استفاده‌کننده می‌تواند یکی از این طریقه‌ها را انتخاب نموده و به کار گیرد.

- طریقه اول: بالا به پایین (Top Down)  
در طریقه بالا به پایین معمولاً دیزاین از ساختمان عمومی شروع شده و به تعریفات ساختمان‌های فرعی می‌انجامد. این طریقه در حالاتی مؤثر است که در آن‌ها تکنالوژی بیش‌تر مطرح بوده و امکانات استفاده از آن بهتر باشد. هم‌چنان زمانی که مشکلات موضوع مورد نظر که باید توسط انبار دیتا حل شوند، واضح و قابل فهم باشند، استفاده از مدل بالا به پایین دارای مؤثریت بیش‌تر است. مؤثریت

استفاده از طریقه دیزاین بالا به پایین در شرایط یادشده، شامل زود تکمیل شدن دیزاین با استفاده از منابع کمتر و مصارف کمتر می شود.

- طریقه دوم: پایین به بالا (Bottom Up)

دیزاین به طریقه پایین به بالا با انجام دادن تجربه ها و طرح های اولیه (Prototypes) آغاز می شود. این طریقه در پروژه های تازه شروع شده و کارهای تجارتي که در مراحل ابتدایی باشند، استفاده می شود. هم چنان کارهای تجارتي و مدیریتی که تازه به استفاده از تکنالوژی روی می آورند، از جمله بخش هایی به شمار می روند که به کارگیری از طریقه پایین به بالا برای شان توصیه می شود. این طریقه اداره های استفاده کننده را قادر می سازد تا قبل از تکمیل پروژه ایجاد انبار دیتا، در مراحل مختلف پیشرفت کار را دیده و تجربه کنند. به خاطر جدید بودن کار و بارشان، تجارب قسمت هایی که درست دیزاین نشده باشند، با مصارف کمتر دوباره اصلاح شده می توانند. به هر اندازه ای که در این طریقه به جزئیات بیش تر در دیزاین پرداخته شود و طرح های اولیه مورد بحث قرار بگیرند، به همان اندازه پروژه های کامل و مؤفق دیزاین خواهند شد.

- طریقه سوم: ترکیبی از دو طریقه قبلی (Top Down & Bottom Up)

در صورت استفاده از ترکیب هر دو طریقه، یک اداره می تواند با در نظر داشت بخش های کاری شان از هر دو طریقه (بالا به پایین و پایین به بالا) کار بگیرند؛ یعنی موضوعاتی که مشخص اند، آن ها را با دیزاین بالا به پایین کار کنند و موضوعات جدید کاری شان را با طریقه دومی دیزاین کنند. خوبی استفاده از طریقه سوم این است که در دیزاین عمومی با در نظر گرفتن امکانات و معلومات دست داشته اداره، طراحان دست آزاد در قسمت کارکردن روی دیزاین پروژه می داشته باشند. هر بخش با هر طریقه ای که کار می شود، مزیت های مربوط به همان طریقه در آن ها استفاده می شوند. انبارهای دیتا در بخش های مختلف ایجاد شده می توانند. در بخش تعلیم و تربیت یک انبار دیتا امکان دارد معلومات عینی و تاریخی یک ممکلت را و یا هم چندین ممکلت را در یک حوزه علمی در بر داشته باشد. کنوانسیون های بین المللی در ساحات تحصیلات عالی و مسلکی در دنیا موجوداند که هر کدام آن ها در ساحه ای از جهان قابل اجراء است. طور مثال کنوانسیون کشورهای قاره های آسیا و استرالیا که در سال 2011 میلادی در شهر توکیو جاپان انکشاف داده شده است، ممالک عضو آن از هر دو قاره باید تمام معلومات و دیتای تحصیلات عالی خود را شریک سازند. اگر تحصیلات عالی افغانستان (دولتی و خصوصی) می خواهد شامل این کنوانسیون شود<sup>9</sup>، یکی از شرایط آن شریک سازی معلومات به روز با کشورهای عضو کنوانسیون است. به عین شکل یک انبار دیتا برای یک شرکت تجارتي ایجاد شده می تواند.

---

<sup>9</sup> این متن در زمستان 1397 نوشته شده است و تا این تاریخ، افغانستان عضویت کنوانسیون آسیا برای تحصیلات عالی را کسب نکرده است. در آینده شاید موضوع تغییر نموده و جمله بالا ضرورت به تعدیل داشته باشد. (به خاطر معلومات به سایت وزارت تحصیلات عالی افغانستان <http://www.mohe.gov.af> مراجعه شود.)

در ذیل پروسه‌های دیزاین یک انبار دیتا به صورت فشرده توضیح شده است:

- یک پروسه تجارتي را انتخاب کنید که مدل آن تهیه شود. مثال‌های پروسه تجارتي می‌تواند شامل پیشنهادها، صورت حساب‌ها، انتقال اجناس، خریداری‌ها، مدیریت حساب‌ها و یا هم فروشات باشند. اگر پروسه‌های تجارتي به سطح اداره بوده و در آن کارها انجام شده و اسنادی تبادل می‌شوند، پس برای آن باید مدل انبار دیتا دیزاین شود. در صورتی که بخش‌های کاری و دیپارتمنت‌های مسئول معلوم باشند، یک مدل کوچک دیتای مرکز تجارتي باید انتخاب شود.
- در ادامه جزئیات یک پروسه تجارتي را انتخاب کنید. هر جزء کوچک در یک پروسه، معلومات در مورد تفصیل بخش کاری را افاده می‌نماید و در جدولی به نام جدول حقیقت یک پروسه (Fact Table) ذخیره می‌شود. مثال اجزای یک پروسه تجارتي عبارت از هر ترانزکشن در یک پروسه بوده می‌تواند.
- تعداد بُعدهای را مشخص کنید که به ریکوردهای جدول حقیقت یک پروسه لازم‌اند. بُعدهای معمول (Typical Dimensions) عبارت از زمان، مشتری، تهیه‌کننده، انبارخانه، نوع ترانزکشن و حالت بوده می‌توانند. به خاطر وضاحت نام‌های بُعدهای نمونه دیتا به انگلیسی نیز ارائه می‌شوند:

Typical dimension includes: Time, Item, Customer, Supplier, Warehouse, Transaction Type, and Status.

- در قدم بعدی واحداث اندازه‌گیری را مشخص کنید که توسط آن‌ها در هر ریکورد جدول حقیقت توضیح شود. واحداث اندازه‌گیری معمول، عبارت از مقدارهای جمع‌پذیر نمبرها مانند واحد پول فروش اجناس و یا تعداد فروش اجناس می‌باشند.

## ۲.۱.۲ ساختمان‌های انبار دیتا (Data Warehouse Architectures)

در ساختمان‌های انبار دیتا خصوصیت‌هایی نهفته‌اند که هر سیستم انبار دیتا باید آن‌ها را داشته باشند. این خصوصیت‌ها در ذیل مختصراً توضیح می‌شوند:

- تفکیک پروسه‌ها (Separation)  
پروسه‌های تحلیلی بالای دیتا و پروسه‌های ترانزکشن دیتا باید از هم‌دیگر جدا باشند. تفکیک این دو کتگوری پروسه‌ها شاید به صورت صد فی صد مشکل باشد ولی تا اندازه ممکن باید پروسه‌های هر دو بخش از هم‌دیگر متمایز گردند.
- صعودپذیری (Scalability)  
بعضاً در سیستم‌های انبار دیتا مقدار دیتا زیاد می‌شود (بیش‌تر از گنجایش یک سیستم می‌شود) و یا هم ضرورت‌های استفاده‌کنندگان که ایجاب به مدیریت می‌کنند، بیش‌تر از حد معمول می‌شود. در چنین حالاتی باید ساختمان‌های سخت‌افزار و نرم‌افزار سیستم به آسانی تبدیل و بهتر شده بتوانند تا جوابگوی ضرورت‌های استفاده‌کنندگان و اداره باشند.
- توسعه‌پذیری (Extensibility)

یک انبار دیتا شاید ضرورت به توسعه در قسمت ساختمان‌ش پیدا کند؛ یعنی تکنالوژی‌ها و اپلیکیشن‌های جدید ضرورت به نصب و استفاده در سیستم را پیدا کنند. ساختمان انبار دیتا باید قابلیت توسعه را در هم‌چون موارد داشته و جوابگوی ضرورت‌های استفاده‌کنندگان باشد. در حالات یاد شده ادارهٔ مربوطه باید ضرورت به دوباره دیزاین کردن تمام سیستم را نداشته باشد.

- امنیت (Security)

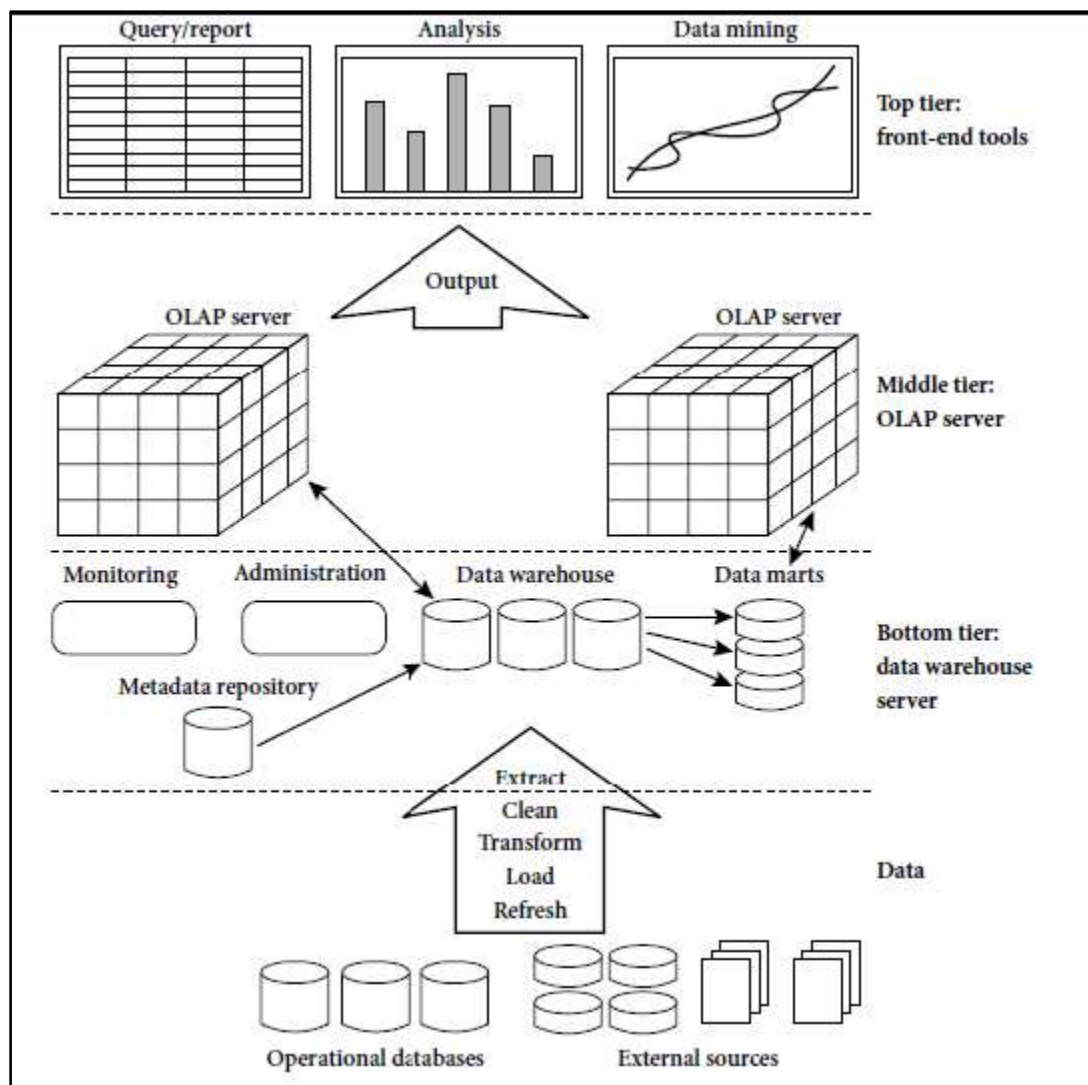
در انبارخانه‌ها معمولاً دیتای استراتژیک یک ارگان ذخیره است. بناءً نظارت بر دسترسی استفاده‌کنندگان به انبار دیتا دارای اهمیت بوده و لازمی است.

- قابلیت مدیریت (Administer-ability)

یکی دیگر از خصوصیات یک سیستم انبار دیتا عبارت از مدیریت آن است. یک سیستم انبار دیتا باید به آسانی و با رهنمایی‌های ساده، قابلیت مدیریت را داشته باشد. یک سیستم مدیریت انبار دیتا که دارای خصوصیات توضیح‌شده در بالا را داشته باشد، به صورت خوب‌تر به استفاده گرفته می‌شود. مسئلهٔ مهمی که در قسمت ساختمان‌های مدیریت دیتا مطرح بوده عبارت از دسته‌بندی این سیستم‌ها به انواع یک طبقه‌یی، دو طبقه‌یی و سه طبقه‌یی اند. انبارهای دیتای یک طبقه‌یی و دو طبقه‌یی عبارت از سیستم‌های ساده بوده و موارد استفاده از این دو نوع نیز به ندرت دیده می‌شود. بناءً در این کتاب از توضیحات در مورد دو نوع اول صرف نظر شده و مستقیماً به توضیحات در مورد نوع سوم پرداخته شده است.

### ۲.۱.۳ ساختمان سه طبقه‌یی انبار دیتا

ساختمان سه طبقه‌یی انبار دیتا یکی از کامل‌ترین و پر استفاده‌ترین ساختمان‌های انبار دیتا به شمار می‌رود. شکل پایین بخش‌های شامل در ساختمان سه طبقه‌یی انبار دیتا را نشان می‌دهد.



شکل ۷-۱ دیاگرام ساختمان سه طبقه‌یی انبار دیتا

طوری که در شکل بالا دیده می‌شود، قسمت بالاتر از بخش دیتا، عبارت از طبقه پایین بوده و آن سرور انبار دیتا است. طبقه وسطی بالاتر از طبقه پایین سرور به نام OLAP<sup>10</sup> جا دارد و طبقه سه مربوط به بخش‌های Frontend می‌شود که استفاده‌کنندگان از آن مجراها، سیستم را به کار می‌گیرند. توضیحات بیش‌تر در مورد طبقه‌های انبار دیتا در ادامه ارائه شده‌اند.

<sup>10</sup> OLAP اختصار کلمات Online Analytical Processing بوده و به خاطر تحلیل دیتای چندبعدی (Multidimensional) استفاده می‌شود.

- طبقه اول: طبقه پایین عبارت از سرور انبار دیتا است که تقریباً همیشه با یک سیستم دیتابیس در ارتباط است. برنامه‌ها و وسایلی در عقب پرده سیستم (Backend) سهولت‌های داخل کردن دیتا به دیتابیس را مهیا می‌سازند. دیتا از منابع مختلف شامل دیتابیس‌های اجرایی، فایل‌ها، وبسایت‌ها و غیره داخل دیتابیس می‌شود. پروگرام و وسایل داخل کردن دیتا، کارهای اساسی دیگر مانند استخراج دیتا مطابق فارمت دیتابیس اصلی، پاکیزه کردن دیتا (دورکردن دیتای زائد و نادرست قبل از داخل کردن به دیتابیس) و انتقال دادن دیتا از منبع اصلی آن به دیتابیس را اجراء می‌کنند. یکی از خصوصیات انبار دیتا همانا یکی ساختن دیتا از منابع مختلف در مورد یک موضوع مشخص بوده به همین لحاظ، برنامه‌ها و وسایل داخل کردن دیتا، تنظیم یک‌جا کردن یا حد اقل نزدیک ساختن ریکوردهای دیتای مربوط به یک موضوع را قبل از داخل کردن به دیتابیس در همین طبقه اول انجام می‌دهند. ریکوردهای دیتا قبل از داخل شدن به دیتابیس به یک فارمت در آورده شده و به سیستم داخل می‌شوند. کارهای دیگر طبقه پایین مانند لود کردن و تازه کردن (Refresh) دیتا به خاطر به‌روز کردن انبار دیتا نیز در این جا انجام می‌شوند.
- دیتایی که با استفاده از اپلیکیشن‌ها استخراج می‌شوند به نام گیت‌وی‌ها (Gateways) یاد می‌شوند. یک گیت‌وی به خاطر ایجاد کد سیکویل و به راه‌انداختن آن روی کمپیوتر سرور استفاده شده و به واسطه آن دیتا استخراج می‌شود. مثال‌های گیت‌وی‌ها عبارت از <sup>11</sup>ODBC، <sup>12</sup>OLEDDB و <sup>13</sup>JDBC اند. در این طبقه بر علاوه بخش‌هایی که توضیح شد، یک ساختمان دیگر به خاطر ذخیره میتادیتا نیز است. در ذخیره‌گاه میتادیتا معلومات بخش سیستم در مورد انبار دیتا و محتویات آن ثبت می‌شود.
- طبقه دوم: طبقه وسطی عبارت از سرور OLAP است که به خاطر تحلیل آنلاین دیتا استفاده می‌شود. این سرور می‌تواند یکی از دو نوع OLAP ارتباط (Relational Online Analytical Processing) باشد و یا هم OLAP چندبعدی (Multidimensional OLAP) باشد.
- a. مدل OLAP یک نمونه انکشاف یافته‌یی از سیستم مدیریت دیتابیس ارتباط است که به خاطر تبدیل کردن عملیه‌ها بالای دیتای چندبعدی به عملیه‌های استاندارد ارتباط به کار برده می‌شود.
- b. مدل MOLAP یک نوع خاص سرور است که از طریق آن عملیه‌های چندبعدی به شکل مستقیم بالای دیتای چندبعدی (Multidimensional) اجراء می‌شود.
- طبقه سوم: طبقه سوم یا بالایی طوری که در شکل نیز دیده می‌شود، مربوط بخش خروجی سیستم می‌شود و در آن دسته‌یی از برنامه‌های انترفیس استفاده‌کننده نهایی (Frontend) نشان داده می‌شود.

<sup>11</sup> ODBC اختصار کلمات Open Database Connection بوده به خاطر ارتباط دادن دیتابیس با کمپیوتر سرور استفاده می‌شود و مربوط شرکت مایکروسافت می‌باشد.

<sup>12</sup> OLEDDB اختصار کلمات Open Linking and Embedding for Databases بوده به خاطر ارتباط دادن دیتابیس با کمپیوتر سرور استفاده می‌شود و مربوط شرکت مایکروسافت می‌باشد.

<sup>13</sup> JDBC اختصار کلمات Java Database Connection بوده و به همین منظور وصل کردن دیتابیس با کمپیوتر سرور استفاده می‌شود.

در این طبقه وسایل کیوری دیتا، تهیه کردن راپور از دیتا، تحلیل دیتا و بهره‌برداری از دیتا (Data Mining) گنجانیده شده است.

## ۷.۲ بهره‌برداری از دیتا (Data Mining)

بهره‌برداری از دیتا یا دیتامیننگ (به خاطر آشنایی کم‌تر به اصطلاح بهره‌برداری از دیتا از این به بعد کلمه دیتامیننگ در متن کتاب استفاده می‌شود.) به معنی پیدا کردن معلومات در مورد یک موضوع مشخص از اثر کاوش ریکوردهای دیتا است. دیتا با استفاده از تکنالوژی‌های مؤثر در کمپیوترها، دیتابیس‌ها، صفحات وب، فایل‌ها، انبارخانه‌های دیتا و غیره موقعیت‌های ذخیره کردن دیتای دیجیتل که قابل دسترسی باشند، ذخیره می‌شود. نام استفاده شده یعنی دیتامیننگ، مفهوم کار را افاده نمی‌کند. اصطلاحات دیگری به خاطر توضیح همین هدف در ساحة کمپیوترساینس و سیستم‌های معلوماتی استفاده شده است.

دیتامیننگ در حقیقت خارج کردن دیتا از منابع آن نبوده بلکه به دست آوردن یا خارج کردن معلومات از منابع ریکوردهای دیتا است؛ یعنی زمانی که دیتا در مورد یک موضوع مشخص از منابعی گرفته شود، دیگر ریکوردهای دیتا نه بلکه اجزای معلومات در مورد همان موضوع مشخص است. به عبارت دیگر، دیتامیننگ عبارت از پروسه‌های دریافتن نمونه‌های دیتا در بین دسته‌های بزرگی از ریکوردهای دیتا است.

همین دریافتن دیتا با استفاده از میتودهای ذکاوت مصنوعی (Artificial Intelligence)، آموختن ماشین (Machine Learning)، احصائیه و سیستم‌های دیتابیس صورت می‌گیرد. هدف کلی استفاده از پروسه‌های دیتامیننگ به دست آوردن و استخراج معلومات از یک دیتا ست و انتقال آن به یک شکل قابل فهم به خاطر استفاده‌های بعدی است.

### خصوصیت‌های کلیدی دیتامیننگ عبارت‌اند از:

- دریافتن نمونه‌های دیتا به شکل خودکار؛
- پیش‌بینی خروجی‌های احتمالی؛
- ایجاد معلومات قابل استفاده؛
- تمرکز بالای دیتا ست‌های وسیع در دیتابیس‌ها.

### ۷.۲.۱ هدف بهره‌برداری از دیتا (دیتامیننگ)

نام دیتامیننگ برگرفته شده از دو مفهوم عمده است. یکی آن جست‌وجوی معلومات تجارتي راجع به یک موضوع مشخص در یک دیتابیس بزرگ است و مفهوم دومی برای به کارگیری اصطلاح استخراج معدن سنگ‌های قیمتی از یک کوه در یک ساحة جغرافیایی است. مثال‌هایی برای مفهوم اول، کلمه دیتامیننگ پیدا کردن معلومات در مورد اجناس مشخص از میان میلیون‌ها ریکورد دیتا بوده می‌تواند. در هر دو مفهوم کارهایی که باید انجام شوند عبارت از پیدا کردن و حل‌اجی کردن مقدارهای بزرگ مواد است؛ و یا هم با استفاده از تکنالوژی‌های موجود و طریقه‌های هوش مصنوعی پیدا کردن عناصر فیزیکی در معادن و دیتا در دیتابیس‌ها



به صورت دقیق‌تر انجام شده می‌تواند. اگر موضوع بحث به سمت عنوان دیتاماینینگ و هدف از آن برگردانده شود، در خواندن و استفاده دیتا از دیتابیس‌ها با احجام بزرگ، تکنالوژی دیتاماینینگ امکانات جدید تجارتی را در این ساحه با توانایی‌های ذیل به میان آورده است:

- پیش‌بینی خودکار در مورد جهت‌ها و حرکت‌ها (Automated Prediction of Trends And Behaviors): دیتاماینینگ پیدا کردن معلومات پیش‌بینی شده در دیتابیس‌های بزرگ را به شکل خودکار زمینه‌سازی می‌کند. سؤال‌هایی که معمولاً ضرورت به تحلیل‌های تفصیلی را دارا بودند، حالا با سرعت زیاد مستقیماً جواب داده می‌شوند. مثال خوب در این مورد، پیش‌بینی کردن خودکار سیستم در قسمت مارکیت هدفمند بوده می‌تواند. در چنین موارد، دیتاماینینگ به استفاده‌های قبلی دیتا توسط استفاده‌کنندگان مراجعه کرده و بلندترین استفاده قسمت‌های دیتا را شناسایی و به استفاده‌کننده به شکل موارد پیش‌بینی شده پیشنهاد می‌کند. در صورت در نظر گرفتن نتایج تهیه شده توسط سیستم، یک تاجر می‌تواند در بخش‌های بیش‌تر استفاده شده سرمایه گذاری نماید.
- دریافتن خودکار نمونه‌های ناشناخته قبلی (Automated Discovery of Previously Unknown Patterns): وسیله‌های دیتاماینینگ داخل دیتابیس‌ها را دیده و نمونه‌های (Patterns) قبلی استفاده شده را که به شکل پنهان (Hidden) موجوداند، در یک قدم شناسایی می‌کنند. به طور مثال، دریافتن نمونه عبارت از تحلیل دیتای فروشات پرچون در یک فروشگاه بوده که در آن اجناسی شناسایی شوند که در اصل مختلف بوده ولی معمولاً توسط مشتریان با هم خریداری می‌شوند. به طور مثال، مشتریانی که آرد خریداری می‌نمایند، همراه با آن خمیرمایه و نمک نیز می‌خرند. این نتیجه برای کسانی که مواد خوراکی می‌فروشند این را می‌رساند تا خمیرمایه و نمک را در نزدیک آرد بگذارند. وقتی مشتری به خاطر خرید آرد می‌آید، با دیدن اجناس دیگری که معمولاً توسط مشتریان با هم خریداری می‌شوند، اجناس اضافی را نیز خواهند خرید. مثال‌های دیگر در این کتگوری شناسایی امتیاز کارت‌های ساختگی با ویژگی‌هایی که توسط سیستم‌های دیتاماینینگ پیدا می‌شوند، بوده می‌تواند.

## ۷.۲.۲ وظایف دیتاماینینگ

در استفاده از دیتاماینینگ معمولاً کارهای ذیل انجام می‌شوند:

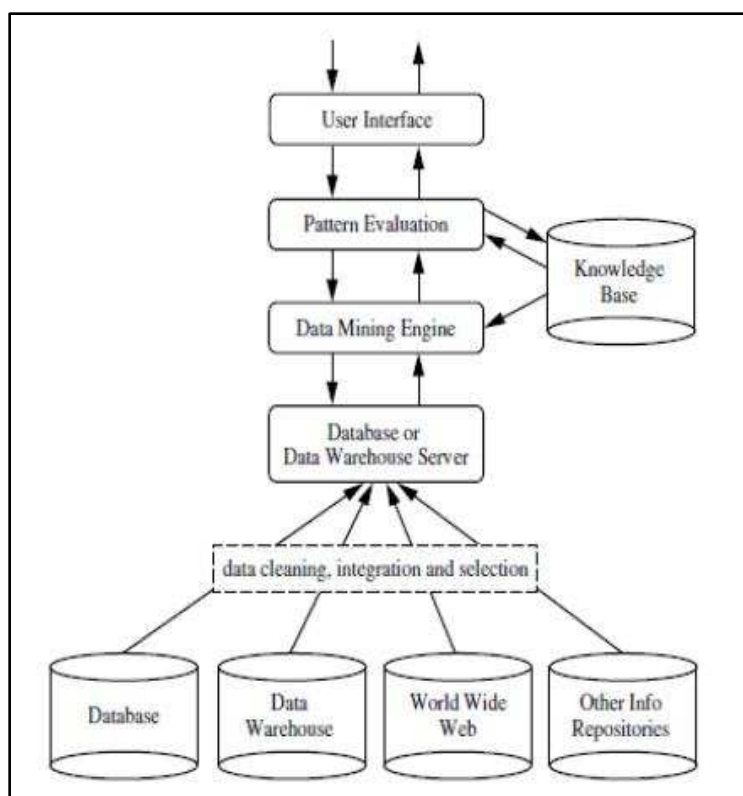
- شناسایی خلاف قاعده‌بودن (Anomaly Detection): خلاف قاعده‌بودن عناصر دیتا به خاطر تفکیک ریکوردهای غیرعادی در مجموعه‌های دیتا را شامل می‌شوند. همین ریکوردهای غیرعادی شاید باعث به میان آمدن اشتباه در دیتا گردد. وظیفه دیتاماینینگ پیدا کردن دیتای منزوی (Outlier)، تغییر داده شده (Changed) و منحرف شده (Deviated) می‌باشد.
- آموزش قاعده هم‌پیوند (Association Rule Learning): از جمله وظایف دیگر دیتاماینینگ جست‌وجوی ارتباط‌ها بین متحولین است. طوری که پیش‌تر نیز گفته شد، در فهرست فروشات یک سوپرمارکیت، مشتریان کدام اجناس را با هم خریداری می‌کنند. دیتاماینینگ وسیله‌هایی را دارد تا

این نوع دیتا را تحلیل نموده و پیشنهاداتی نظر به ضرورت مارکیت به استفاده کننده در زمینه ارائه دارد. این نوع فعالیت بعضاً به نام تحلیل خرید مارکیت نیز یاد می شود.

- کلاستر سازی (Clustering): کلاستر سازی یا خوشه یی کردن دیتا یکی دیگر از وظایف دیتاماینینگ به شمار رفته که در آن ریکوردهای دیتا که به نحوی با هم وجوه مشترک (خصوصیات مشابه) دارند، دریافت می شوند و به صورت دسته های (کلاسترهای) قابل استفاده تنظیم می شوند. البته این کار علاوه از ساختمان های خود دیتابیس ها بوده و به خاطر تنظیم و دسته بندی کردن دیتا استفاده می شوند.
- صنف بندی کردن (Classification): از طریق دیتاماینینگ ساختمان های موجود دیتا، عمومی سازی شده و دیتای جدید از آن به دست آمده می تواند. طور مثال یک برنامه ایمیل بخواهد یک ایمیل را به دسته ایمیل های Spam تنظیم کند.
- مراجعت (Regression): کوشش به خاطر پیدا کردن یک تابع که توسط آن دیتا با کمترین خطا مدل سازی شود.
- خلاصه سازی (Summarization): دیتاماینینگ شکل های نمایش دیتا ست ها را به صورت بسیار خلاصه تجسم سازی و ایجاد راپور را زمینه سازی کرده می تواند.

### ۷.۳ ساختار دیتاماینینگ (Architecture of Data Mining)

یک سیستم دیتاماینینگ به صورت عمومی دارای بخش هایی است که در شکل پایین نشان داده شده اند. البته بخش های یک سیستم دیتاماینینگ به صورت مختصر در ادامه توضیح می گردد.



شکل ۷-۲ ساختار یک سیستم دیتاماینینگ

## اساس معلومات (Knowledge Base)

این بخش در قسمت پیدا کردن و ارزیابی کردن نمونه‌های معلومات کمک می‌کند تا معلومات به دست آمده در مورد یک موضوع بیش‌تر جالب شود. معلومات می‌تواند بر بنیاد مفاهیم تنظیم شده و توسط این بخش مشخصه‌های معلومات در سطوح مختلف قرار داده شوند. بر علاوه، معلومات مشخص در مورد یک موضوع، معلومات دیگری که فکر می‌شود برای یک استفاده‌کننده از یک دومین مشخص جالب بوده نیز در این بخش در نظر گرفته می‌شود. مثال‌های دیگری که در قسمت معلومات به اساس یک دومین یا ساحة مشخص نشان داده می‌شود، عبارت از معیارهای اضافی به خاطر جالب‌ساختن معلومات پیدا شده و میتادیتا از منابع متنوع بوده می‌تواند.

بخش اساسی معلومات (Knowledge Base) تبادلۀ نمونه‌ها را با بخش دیگری به نام Pattern Evaluation یا ارزیابی نمونه‌های دیتا به صورت دو جانبه (قرار شکل بالا) انجام می‌دهد. همین بخش قیده‌های دیتا را به ماشین دیتامایننگ می‌فرستد. ماشین دیتامایننگ در فقرۀ بعدی توضیح شده است.

- ماشین دیتامایننگ (Data Mining Engine): ماشین دیتامایننگ یکی از بخش‌های مهم و کلیدی در سیستم دیتامایننگ به شمار می‌رود. ماشین دیتامایننگ متشکل از مادیول‌های وظیفوی به خاطر اجراء و تنظیم کارها است. کارهایی که در این بخش انجام می‌شود عبارت از متمایز ساختن عناصر دیتا، تحلیل‌های ارتباط و تناظر دیتا، طبقه‌بندی، پیش‌بینی، تحلیل دسته‌یی، تحلیل قطعات دورافتاده و تحلیل ارزیابی دیتا است. به خاطر وضاحت بیش‌تر و ریفرنس معلومات اصطلاحات انگلیسی کارهای قابل اجراء توسط ماشین دیتامایننگ در ذیل فهرست می‌گردد.

a. Characterization

b. Association and Correlation Analysis

c. Classification

d. Prediction

e. Cluster Analysis

f. Outlier Analysis

g. Evaluation Analysis

- بخش ارزیابی نمونه (Pattern Evaluation Module): در این بخش اساساً پیدا کردن و اندازه‌گیری جالب‌بودن عناصر دیتا صورت گرفته و تمرکز پیدا کردن نمونه‌های قابل توجه برای استفاده‌کنندگان انجام می‌شود. به خاطر تعیین صفت‌های جالب‌بودن نمونه‌های معلومات از فلت‌های تعیین‌شده، استفاده می‌شود. در بعضی موارد بخش‌های ارزیابی نمونه و بخشی که قبلاً توضیح شد (ماشین دیتامایننگ) به شکل یک واحد عمل می‌کنند. چنین حالت بستگی به استفاده از نوعیت میتود دیتامایننگ می‌داشته باشد. برای به دست آوردن نتایج مؤثر در دیتامایننگ، رُل جالب‌بودن نمونه‌ها مهم بوده، طوری که به هر اندازه‌یی که در این مورد به تفصیل بیش‌تر پرداخته شود به همان اندازه مؤثریت آن بیش‌تر می‌شود.

- بخش انترفیس برای استفاده‌کننده (The User Interface): طوری که از نام این بخش پیداست، یک انترفیس به خاطر ارتباط بین استفاده‌کننده و سیستم دیتاماینگ به کار برده می‌شود. انترفیس به استفاده‌کننده سیستم اجازه می‌دهد تا کیوری یا کاری را که از سیستم می‌خواهد به نحوی وارد سیستم دیتاماینگ نماید. معلومات کمکی به خاطر جست‌وجوی بهتر دیتا و اجرای کامل سیستم نیز از طریق انترفیس به استفاده‌کننده ارائه می‌شود. بر علاوه، این بخش استفاده‌کنندگان سیستم را اجازه می‌دهد تا دیتابیس و ساختمان‌های انبار دیتا (Data Warehouse) را مشاهده کرده و نمونه‌های ذخیره‌شده را ارزیابی نماید. این مشاهده‌ها و دسترسی به ساختمان‌های دیتا با امکانات دیدن آن‌ها به اشکال مختلف مهیا شده است.

## ۷.۴ طبقه‌بندی سیستم‌های دیتاماینگ

یک سیستم دیتاماینگ به صورت عمومی می‌تواند به اساس خصوصیات ذیل تقسیم بندی شود:

- تکنالوژی دیتابیس (Database Technology)
  - احصائیه (Statistics)
  - آموختن ماشینی (Machine Learning)
  - دانش معلومات (Information Science)
  - تجسم فکری (Visualization)
  - و یا هم بعضی موارد دیگر.
- بر علاوه تقسیم‌بندی لیست‌شده قبلی، یک سیستم دیتاماینگ می‌تواند به اساس معیارهای دیگری که در ذیل فهرست شده‌اند، نیز تفکیک شود:

- نوع دیتابیس‌های استفاده‌شده؛
- نوع معلومات استفاده‌شده؛
- نوع تخنیک‌های استفاده‌شده؛
- اپلیکیشن‌های استفاده‌شده.

طبقه‌بندی سیستم‌های دیتاماینگ به اساس فهرست دومی به صورت مختصر توضیح داده می‌شود.

### ۷.۴.۱ تقسیم‌بندی سیستم‌های دیتاماینگ به اساس نوع دیتابیس‌ها

یک سیستم بهره‌برداری از دیتا (Data Mining) می‌تواند با در نظر داشت انواع دیتابیس‌های استفاده شده در آن مشخص شود. سیستم‌های دیتابیس به اساس خصوصیات مختلف از قبیل مدل‌های دیتا، انواع دیتا و غیره طبقه‌بندی شده می‌تواند؛ و همین طبقه‌بندی دیتابیس‌های استفاده شده در سیستم دیتاماینگ، مشخص‌کننده نوعیت سیستم‌های دیتاماینگ نیز می‌باشد. به طور مثال، اگر نوعیت یک سیستم دیتاماینگ با در نظر داشت مدل دیتای آن تعیین شود، یک سیستم دیتاماینگ می‌تواند از نوع ارتباط باشد، می‌تواند از

نوع ترازکشنل باشد، می‌تواند از نوع شی‌گرا (Object Oriented) باشد و یا هم از نوع انبار دیتا (Data Warehouse) باشد.

#### ۷.۴.۲ تقسیم‌بندی سیستم‌های دیتاماینگ به اساس نوع معلومات

طوری که در عنوان دیده می‌شود، سیستم‌های دیتاماینگ به اساس استخراج معلومات طبقه‌بندی شده می‌توانند؛ یعنی این سیستم‌ها با در نظر داشت کارهایی که روی پیدا کردن دیتا اجراء می‌شوند، تقسیم‌بندی شده می‌توانند. مثال در مورد کارهای قابل اجراء روی سیستم‌های دیتاماینگ قرار ذیل است:

- متمایز ساختن عناصر دیتا (Characterization)
- نکته‌سنجی در عناصر دیتا (Discrimination)
- تحلیل‌های ارتباط و تناظر دیتا (Association and Correlation Analysis)
- طبقه‌بندی (Classification)
- پیش‌بینی عناصر دیتا (Prediction)
- کلسر سازی دیتا (Clustering)
- تحلیل قطعات دور افتاده دیتا (Outlier Analysis)
- تحلیل تکامل دیتا (Evolution Analysis)

#### ۷.۴.۳ تقسیم‌بندی سیستم‌های دیتاماینگ به اساس تخنیک‌ها

سیستم‌های دیتاماینگ به اساس تخنیک‌های به کار برده شده طبقه‌بندی شده می‌توانند. تخنیک‌های استفاده شده به خاطر گرفتن دیتا مربوط به رُل استفاده کننده در استخراج دیتا از سیستم و میتودهای به کار گرفته شده به خاطر تحلیل دیتا است.

#### ۷.۴.۴ تقسیم‌بندی سیستم‌های دیتاماینگ به اساس نوع اپلیکیشن‌های استفاده شده

استفاده از اپلیکیشن‌های مختلف و سازگاری آن‌ها با سیستم‌های استخراج دیتا یکی دیگر از مسایلی به شمار می‌رود که به اساس آن‌ها یک سیستم دیتاماینگ مشخص شده می‌تواند. مثال‌های این نوع اپلیکیشن‌ها عبارت از انجام کارهای مالی، ارتباط از دور، (Telecommunication)، مارکیت‌های ستاک (Stock Markets)، سیستم‌های ایمیل و غیره‌اند.

#### ۷.۵ مقایسه انبار دیتا با بهره‌برداری از دیتا

دیتا و ارقام در دیتابیس‌ها تنظیم و ذخیره می‌شوند. تحلیل معلومات و تجزیه آن به خاطر به دست آوردن عناصر دیتا یکی از مزیت‌های استفاده از دیتابیس‌ها است. زمانی که دیتا درست تحلیل شده و در بخش‌های مشخصی به نام جدول‌ها یا رابطه‌ها قابلیت ذخیره شدن را پیدا می‌کند، مسایل مهم از قبیل جلوگیری از پاک شدن دیتا، جلوگیری از تکرار دیتا و کم کردن پروسه‌های کاری استفاده از دیتا را مهیا می‌سازد. ذخیره

دیتا با در نظر گرفتن معیارهای ثابت برای مدل‌های ارتباط به حیث یک روی سکه توسط انبار کردن دیتا (Data Warehousing) صورت می‌گیرد.

دوباره استفاده کردن از مقادیر بزرگ دیتای ذخیره شده به خاطر اهداف مشخص با طریقه‌های استاندارد به حیث دو روی سکه از اهمیت خاصی برخوردار است؛ یعنی به هر اندازه‌یی که دیتا ارزش دارد به همان اندازه استفاده از دیتا نیز دارای اهمیت است. در صورت داشتن سیستم‌ها و دیتای عالی اگر دیتا استفاده نشود، به این معنی است که انبار کردن دیتا یک کار زائد بوده و در آن صرف منابع و زمان به مصرف رسیده است. اگر مسئله دو روی سکه یا استفاده کردن از دیتا به صورت خوب و مؤثر به کار گرفته شود، در حقیقت دیتای انبار شده ارزش پیدا می‌کند. همین استفاده از دیتای انبار شده که در اصطلاح آن را به نام استخراج کردن دیتا از معدن آن می‌نامند عبارت از یک ساحة کاری در علوم مدیریت دیتا و سیستم‌های معلوماتی بوده و به نام بهره‌برداری از دیتا (Data Mining) یاد می‌شود. اگر با وضاحت بیش‌تر گفته شود، انبار دیتا به خاطر تنظیم، مدیریت، ذخیره و یک جاسازی دیتا از منابع مختلف استفاده می‌شود و دیتاماینگ به خاطر استخراج معلومات ضروری و نمونه‌های دیتا به خاطر اهداف مشخص به کار گرفته می‌شود. به کمک دیتاماینگ، معلومات از دیتابیس‌ها استخراج می‌شود، ولی چون سیستم‌های انبار دیتا مقادیر بزرگ دیتا را با کیفیت بهتر از منابع مختلف به شمول دیتابیس‌ها از طریق خود، قابل دسترس به سایر استفاده‌کنندگان می‌سازد، پس استفاده از انبار دیتا توسط سیستم‌های دیتاماینگ بیش‌تر توصیه می‌شود. تفاوت عمده بین هر دو سیستم (انبار دیتا و بهره‌برداری از دیتا) به اساس معلومات بالا کاملاً واضح است. به خاطر ادامه بحث به بعضی موضوعات مربوط به هر دو نوع سیستم‌ها در ذیل پرداخته می‌شود.

### خصوصیت‌های عمده انبار دیتا (Data Warehouse Key Feature)

علاوه بر توضیحات در قسمت‌های اول فصل اینک به صورت خلاصه خصوصیات کلیدی انبار دیتا یک‌بار دیگر در ذیل توضیح داده می‌شود.

- متمایل بودن به موضوع (Subject Oriented): یک انبار دیتا به یک موضوع تمایل می‌داشته باشد؛ یعنی در انبار دیتا معلومات در ارتباط به موضوعات مدیریت می‌شوند نه این‌که معلومات در رابطه به فعالیت‌های روزمره یک اداره در آن ذخیره و مدیریت شود. مثال‌های آن موضوعات، عبارت‌اند از دیتا در مورد محصولات، مشتریان، تمویل‌کنندگان، فروشات، خدمات و غیره.
- کامل بودن (Integrated): دیتا و ارقامی که امکان دارد از منابع مختلف گرفته شده باشد، باید به طور یک پارچه، کامل و منسجم قابل استفاده باشد. یک انبار دیتا متشکل از دیتای ترکیب شده از منابع مختلف مانند دیتابیس‌های ارتباط، فایل‌های دیتا، معلومات از صفحات وب و غیره است.
- زمان‌های گوناگون (Time Variant): تمام دیتای شامل انبار دیتا، به اساس دوره‌های زمانی مشخص قابل شناسایی می‌باشند. همین دوره ترتیب شدن دیتا (زمان و تاریخ آن) باید به خاطر شناسایی دیتا استفاده شده بتواند.

- پابرجا بودن (Non-Volatile): دیتا در انبار دیتا به صورت ثابت و پابرجا است. دیتا بیش تر به یک انبار دیتا اضافه می شود ولی هیچ دیتا نباید از انبار دیتا کاسته (دور) کرده شود.
- فواید استفاده از انبار دیتا (Data Warehouse):**

- دیتای ثابت و باکیفیت (Consistent and Quality Data)
- تنزیل قیمت (Cost Reduction)
- دسترسی به وقت به دیتا (More Timely Data Access)
- اجراء و تولید انکشاف یافته (Improved Performance and Productivity)

### **خصوصیت های عمده بهره برداری از دیتا (Data Mining Key Features)**

علاوه بر توضیحات در قسمت های وسط فصل، اینک به صورت خلاصه خصوصیات کلیدی بهره برداری از دیتا یکبار دیگر در ذیل فهرست می شوند.

- دریافتن نمونه های دیتا به شکل خودکار (Automatic Discovery of Patterns)
- پیش بینی کردن خروجی های احتمالی (Prediction of Likely Outcomes)
- ایجاد معلومات قابل استفاده (Creation of Actionable Information)
- تمرکز بالای دیتا ست های وسیع و دیتابیس ها (Focus on Large Datasets and Databases)

### **فواید استفاده از دیتا ماینینگ (Data Mining)**

- بازاریابی مستقیم (Direct Marketing): توانایی پیش بینی این که چه کسی بیش تر علاقه مند کدام محصولات است؛ یعنی کی ها بیش تر کدام اجناس را خریداری و یا استفاده کرده اند.
- تحلیل گرایش ها (Trend Analysis): دانستن این که یک مارکیت (بازار) سوداگری به کدام سمت روان است. کی ها کدام معاملات را بیش تر انجام می دهند.
- شناسایی تقلب و فریب کاری (Fraud Detection): تخنیک های دیتا ماینینگ می توانند در شناسایی کارهای تقلبی و نادرست کمک کنند. ادعای نادرست از شرکت بیمه (Insurance) یکی از مثال های خوب در این زمینه بوده می تواند.
- پیشگویی کردن در مارکیت های معاملات پولی (Forecasting in Financial Markets): تخنیک های دیتا ماینینگ در این زمینه نیز کمک زیاد نموده در قسمت نوسان نرخ اسعار، اشخاص مسلکی می توانند مستقیماً از آن سود ببرند.

طوری که در این بحث دیده شد، سیستم های انبار دیتا و بهره برداری از دیتا به شکل دو روی یک سکه یکی به دیگری کمک می نمایند تا استفاده کنندگان به حد اعظمی از آن ها سود ببرند. یک سیستم انبار دیتا خوب است ولی اگر دیتای آن استفاده نشود، کار آبی سیستم به صفر ضرب شده و یا حتی به خاطر استفاده منابع به جهت منفی سوق داده می شود. به عین شکل یک سیستم بهره برداری از دیتا زمانی مؤفق است که دیتای با کیفیت و خوب در اختیار آن گذاشته شود.



این فصل با معلومات در مورد موضوعات قابل بحث و توضیح آغاز شد. در آغازین کلمات فصل به مسایل مربوط به جمع‌آوری، تحلیل، ترتیب، ذخیره و دوباره استفاده از معلومات اشاره شده است. عناوین عمده تشکیل‌دهنده در این فصل عبارت از انبار دیتا (Data Warehousing)، بهره‌برداری از دیتا (Data Mining) و تفاوت‌ها بین هر دو موضوع را در بر می‌گیرند.

در قسمت انبار کردن دیتا، پروسه‌های دیزاین انبار دیتا، ساختمان‌های انبار دیتا به شمول ساختمان سه طبقه‌یی دیتا به اساس استندرها و با مثال‌ها ارائه گردیده است. خصوصیات مشترک سیستم‌های انباردیتا تحت عنوان انبار کردن دیتا با ذکر مثال‌های مناسب توضیح شده‌اند. در عنوان بعدی (پروسه‌های دیزاین)، طریقه‌های مختلف از قبیل طریقه بالا به پایین، طریقه پایین به بالا و یا هم ترکیب هر دو طریقه به خاطر دیزاین کردن یک سیستم انبار دیتا به توضیح گرفته شد. این سه حالت در قسمت استفاده از طریقه‌های دیزاین انبار دیتا عام بوده و در بیش‌تر کارهای دیزاین مانند دیزاین دیتابیس ارتباطات نیز به کار گرفته می‌شود.

عنوان عمومی دیگر عبارت از بهره‌برداری از دیتا یا همان عنوان مشهور (Data Mining) بوده که از عناوین فرعی تحت آن بحث شد. در عناوین فرعی هدف استفاده از دیتاماینگ، وظایف دیتاماینگ، ساختار دیتاماینگ و طبقه‌بندی سیستم‌های دیتاماینگ هر کدام به ترتیب مورد بحث قرار گرفته است. در اکثر عناوین این بخش موضوعات مهمی توضیح گردید. از جمله در عنوان طبقه‌بندی سیستم‌های دیتاماینگ دو دیدگاه مختلف در نظر گرفته شده و مثال‌ها و توضیحات لازم در زمینه ارائه شده است.

در ادامه فصل، مقایسه دو عنوان عمده یعنی انبار دیتا با دیتاماینگ از ابعاد مختلف به بحث گرفته شد. هدف اصلی انبار دیتا و نیز هدف اصلی استفاده از دیتاماینگ و اهمیت هر کدام در قسمت استفاده کردن مؤثر از معلومات موجود نشان داده شده‌اند. خصوصیات هر دو عنوان به صورت جداگانه در این قسمت دوباره فهرست گردیده و توضیحات قسمی به آن افزوده شده است. در نهایت در کنار توضیح خصوصیات، فواید هر کدام از سیستم‌ها به صورت نسبتاً مختصر توضیح داده شد.





### فعالیت گروهی

با استفاده از مراحل پایین یک انبار دیتا (Data Warehouse) را ایجاد کنید:

۱. یک پروسه تجارتي مربوط به یک شرکت را انتخاب کنید که مدل آن تهیه شود. مثال های پروسه تجارتي می تواند شامل پیشنهادها، صورت حساب ها، انتقال اجناس، خریداری ها، مدیریت حساب ها و یا هم فروشات باشند.
۲. جزئیات یک پروسه تجارتي را پیدا کنید. هر جزء کوچک در یک پروسه، معلومات در مورد تفصیل بخش کاری را افاده می نماید و در جدولی به نام جدول حقیقت یک پروسه (Fact Table) ذخیره می شود. مثال اجزای یک پروسه تجارتي عبارت از هر ترانزکشن و غیره در یک پروسه بوده می تواند.
۳. تعداد بعدهایی را مشخص کنید که به ریکوردهای جدول حقیقت یک پروسه لازم اند. بعدهای معمول (Typical Dimensions) عبارت از زمان، شیء، مشتری، تهیه کننده، انبارخانه، نوع ترانزکشن و حالت بوده می توانند.

## مأخذ (References)

---

1. Cafarella, M. J. et al. (2007). *Structured Querying of Web Text*. 3<sup>rd</sup> Biennial Conference on Innovative Data Systems Research (CIDR). CA, USA.
2. Elmasri, R. and Navathe, S. B. (2011). *Fundamentals of Database Systems – 6<sup>th</sup> Edition*. Addison-Wiskey, NY. USA.
3. Han, J. Kamber, M. Pei, J. (2012). *Data Mining Concepts and Techniques – 3<sup>rd</sup> Edition*. Morgan Kaufmann Publishers of Elsevier, Waltham, MA, USA.
4. Inmon, W. H. (2005). *Building the Data Warehouse – 4<sup>th</sup> Edition*. Wiley Publishing, Inc. Indianapolis, Indiana, USA.
5. Manning, C. D. Raghavan, P. and Schutze, H. (2009). *An Introduction to Information Retrieval*. Cambridge University Press, Cambridge, England.
6. Ray, E. T. (2001). *Learning XML*. O'Reilly & Associates, Inc. US.
7. Tan, P. N. Steinbach, M. and Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wesley, NJ, USA.
8. Tibor, R. – PHD. (2011). *Advanced DBMS*. Kiado. Kezirat Lezarva.
9. Vaughan, T. (2010). *Multimedia: Making it Work – 8<sup>th</sup> Edition*. Mc Graw Hill. NY.
10. e-Book: *DBMS Tutorial*. by Tutorials Point Pvt. Ltd. <http://tutorialspoint.com>. (2015).
11. e-Book: *XML Tutorial*. by Tutorials Point Pvt. Ltd. <http://tutorialspoint.com>. (2018).