



دانشگاه صنعتی اصفهان
دانشکده مهندسی برق و کامپیوتر

عنوان: تکلیف اول درس پایگاه داده ها ۱

نام و نام خانوادگی: علیرضا ابره فروش

شماره دانشجویی: ۹۸۱۶۶۰۳

نیم سال تحصیلی: پاییز ۱۴۰۰

مدرس: دکتر ناصر قدیری مدرس

دستیاران آموزشی: عارف آسمند - بهاره حاجی هاشمی - پردیس مرادیکی

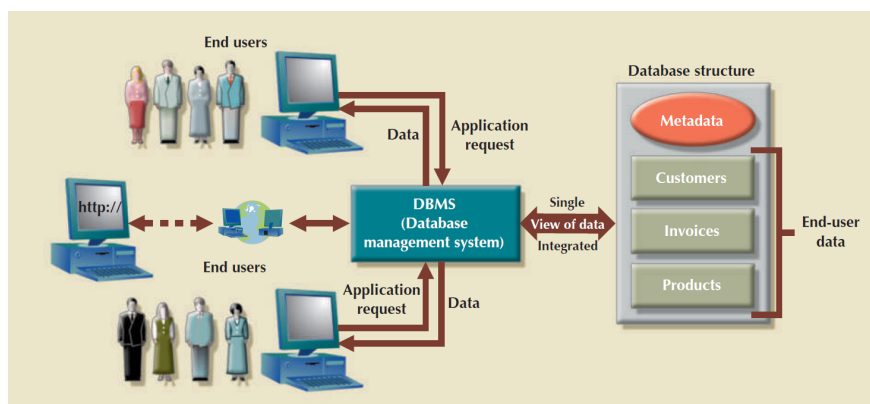
- سیدمهدی موسوی

۱

اگر سوال بخش بندی شده نباشد، پاسخ آن در این قسمت نوشته می شود.

۱.۱

• DBMS به عنوان واسطه بین کاربر و پایگاه داده عمل می کند. این ساختار پایگاه داده خود به عنوان مجموعه ای از فایل ها ذخیره می شود و تنها راه دسترسی به اطلاعات موجود در آن فایل ها از طریق DBMS است. شکل ۱.۱ بر این نکته تأکید می کند که DBMS به کاربر (یا برنامه کاربردی) یک نمای واحد و یکپارچه از داده های موجود در پایگاه داده ارائه می دهد. DBMS همه request های برنامه را دریافت و آن ها را به عملیات های پیچیده مورد نیاز برای پاسخ به این request ها ترجمه می کند. بسیاری از پیچیدگی های داخلی پایگاه داده به وسیله DBMS از برنامه های کاربردی و کاربران پنهان می شود. برنامه کاربردی ممکن است توسط یک برنامه نویس با استفاده از یک زبان برنامه نویسی مانند Visual Basic.NET، Java یا C# نوشته شود یا ممکن است توسط یک DBMS utility program ساخته شود. داشتن یک DBMS بین application کاربر و پایگاه داده مزایای مهمی را به ارمغان می آورد. اولاً DBMS به داده ها اجازه می دهد که بتوانند بین چندین برنامه به اشتراک گذاشته شوند. ثانیاً DBMS بسیاری از دیدگاه (view) های مختلف کاربران از داده ها را با هم ادغام می کند و در یک مخزن همه جانبه ارائه می دهد.



شکل ۱: DBMS تعاملات بین کاربر و پایگاه داده را مدیریت می کند.

• افزودنی داده ها و ناسازگاری (داده ها در چندین فرمت فایل ذخیره می شوند و در نتیجه افزودنی کمتری از اطلاعات در فایل های مختلف رخ می دهد)، سخت بودن دسترسی به داده ها (نیاز به نوشتن یک برنامه جدید برای مدیریت هر تسک جدید)، ایزولگی داده ها، مشکلات یکپارچگی، آپدیت اتمیک، دسترسی همزمان توسط چند کاربر، مشکلات امنیتی

۲.۱

گام های طی شده توسط DBMS جهت پاسخ به درخواست کاربر (query) در شکل ۲ مشخص شده است. گام های پایه عبارتند از:

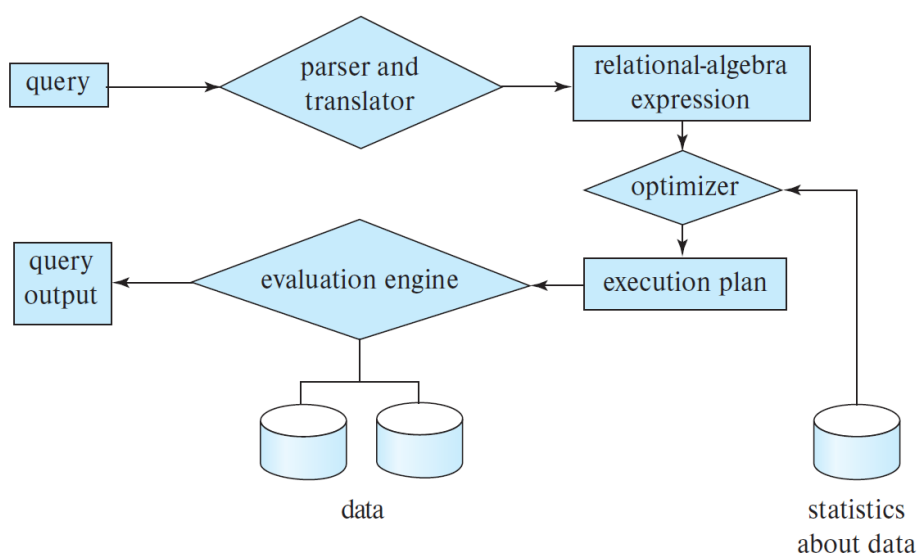
۱. پارس و ترجمه (Parsing and translation)

۲. بهینه سازی (Optimization)

۳. ارزیابی (Evaluation)

پیش از اینکه query processing آغاز شود، سیستم باید query را به یک فرمت قابل ترجمه تبدیل کند. یک زبان مانند SQL برای این کار مناسب می‌باشد، اما همچنان جهت نمایش داخلی query در یک سیستم مناسب نیست. یک راه مبتنی بر جبر رابطه‌ای وجود دارد که مناسب‌تر است.

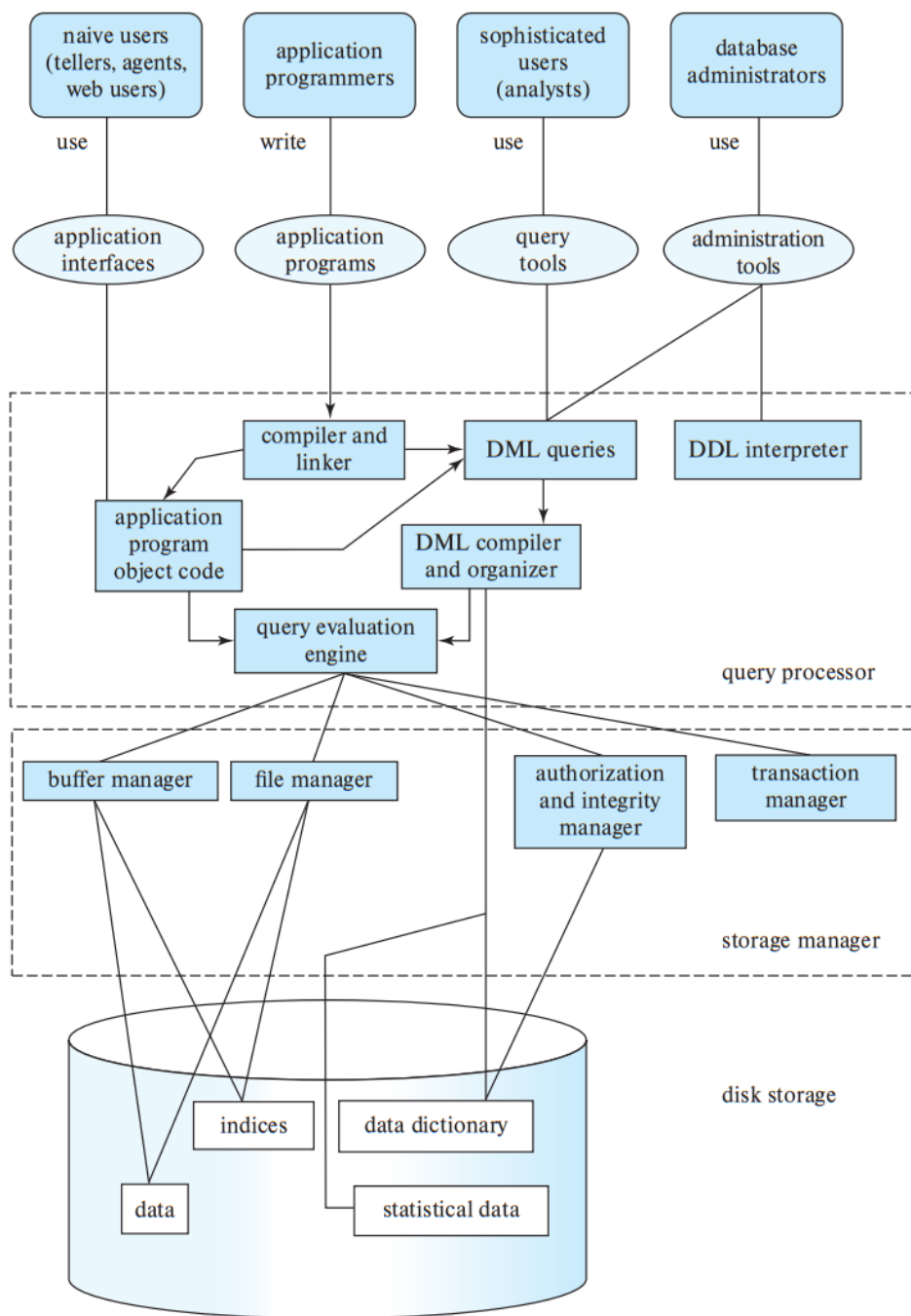
ابتدا سیستم درخواست را به یک query تبدیل می‌کند. از اینجا به بعد پردازش بر عهده DBMS است که query را به عبارتی تحت عنوان جبر رابطه‌ای پارس و ترجمه (این ترجمه شبیه به کاری است که پارسر کامپایلر انجام می‌دهد) می‌کند و سپس بهینه (چند روش ممکن) می‌کند و در انتخاب روش به آمار و سائز جداول توجه می‌کند. حال DBMS به یک execution plan رسیده است که چگونگی اجرای دستور وارد شده توسط کاربر را نشان می‌دهد. سپس در اختیار evaluation engine قرار می‌گیرد که در این گام به سراغ داده‌ها می‌رود و آن‌ها را کنار هم می‌چیند و خروجی را برمی‌گرداند.



شکل ۲: گام‌های پردازش query

۳.۱

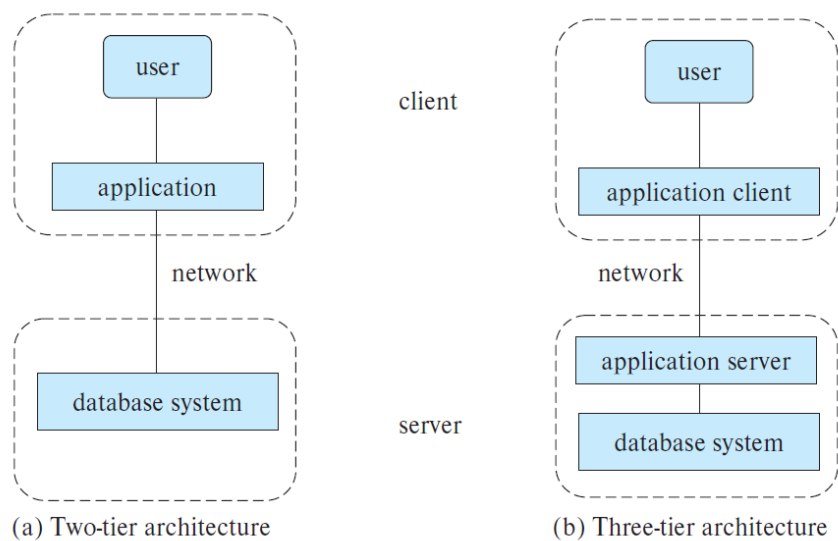
پاسخ بخش سوم سوال در این قسمت نوشته می‌شود.



شکل ۳

۲

ابجدھوز



شکل ۴

ویژگی‌ها	معماری دولایه	معماری سه‌لایه
سرعت	کمتر (کندتر)	بیشتر (سریعتر)
امنیت	کمتر (کلاینت می‌تواند مستقیماً با پایگاه داده تعامل داشته باشد)	بیشتر (کلاینت مجاز به تعامل مستقیم با پایگاه داده نمی‌باشد)
افزونگی	خانه شماره ۵	خانه شماره ۶
مقیاس‌پذیری	کمتر ()	بیشتر ()
انعطاف‌پذیری	خانه شماره ۵	خانه شماره ۶
یک‌پارچگی	خانه شماره ۸	خانه شماره ۹

جدول ۱: جدول شماره ۱

۳

اگر سوال بخش بندی شده نباشد، پاسخ آن در این قسمت نوشته می‌شود.

۱.۳

پاسخ بخش اول سوال در این قسمت نوشته می‌شود.

۲.۳

پاسخ بخش دوم سوال در این قسمت نوشته می شود.

۴

اگر سوال بخش بندی شده نباشد، پاسخ آن در این قسمت نوشته می شود.

۱.۴

پاسخ بخش اول سوال در این قسمت نوشته می شود.

۲.۴

پاسخ بخش دوم سوال در این قسمت نوشته می شود.

۳.۴

پاسخ بخش دوم سوال در این قسمت نوشته می شود.

۴.۴

پاسخ بخش دوم سوال در این قسمت نوشته می شود.

۵

اگر سوال بخش بندی شده نباشد، پاسخ آن در این قسمت نوشته می شود.

۱.۵

$$\Pi_{Title, ReturnDate}(Borrow \bowtie_{MemberID = 1356 \wedge IsReturned = False} Book)$$

۲.۵

$$\Pi_{Name}(Member \bowtie_{Member.CategoryID = "Physics"} (Book \bowtie_{Book.BookID = Borrow.BookID \wedge CategoryID = "Physics"} Borrow))$$

۳.۵

$$\Pi_{Name, Title}(Member \bowtie_{Member.CategoryID = Book.CategoryID} Book) - \Pi_{Name, Title}(Borrow \bowtie_{Borrow.MemberID = Member.MemberID \wedge Borrow.BookID = Book.BookID} (Member \bowtie_{Member.CategoryID = Book.CategoryID} Book))$$

۴.۵

$\Pi_{Name, Title}(Member \bowtie_{Member.MemberID = Borrow.MemberID}$
 $(Borrow \bowtie_{CategoryID = "Drama" \wedge IsReturned = False \wedge Today - ReturnDate > 10 Days} Book))$

۵.۵

ابتدا عبارت را به شکل زیر تفکیک می کنیم:

$D \leftarrow Borrow \bowtie_{Borrow.BookID = Book.BookID} Book$

$C \leftarrow D \bowtie_{Borrow.MemberID = Member.MemberID} Member$

$B \leftarrow \sigma_{Borrow.NumDays \times Book.Penalty \geq 100000}(C)$

$A \leftarrow \Pi_{Member.Name, Book.Title}(B)$

D لیست اطلاعات کتاب های امانت گرفته شده را همراه با اطلاعات امانت آن ها برمی گرداند.
 C لیست اطلاعات اعضا و اطلاعات کتاب های امانت گرفته شده آن ها را همراه با اطلاعات امانت آن ها برمی گرداند.
 B سطرهایی از C را برمی گرداند که جریمه دیرکرد آن ها بزرگتر یا مساوی ۱۰۰۰۰۰ تومان باشد.
 A که همان عبارت نهایی است ستون های نام عضو و نام کتاب را از B برمی گرداند.
 پس نتیجه عبارت، نام عضو و نام کتاب هایی که امانت گرفته اند و جریمه دیرکرد آن ها بزرگتر یا مساوی ۱۰۰۰۰۰ تومان است می باشد.

۶.۵

پاسخ بخش دوم سوال در این قسمت نوشته می شود.

۶

در این قسمت با نحوه درج فرمول های ریاضی آشنا می شوید:

$$E = mc^2$$

۷

در این قسمت با نحوه درج اشکال آشنا می شوید:

۸

در این قسمت با نحوه درج جداول آشنا می شوید:



شکل ۵: شکل شماره ۱

خانه شماره ۳	خانه شماره ۲	خانه شماره ۱
خانه شماره ۶	خانه شماره ۵	خانه شماره ۴
خانه شماره ۹	خانه شماره ۸	خانه شماره ۷

جدول ۲: جدول شماره ۱

۹

در این قسمت با نحوه درج انواع لیست‌ها آشنا می‌شوید:

۱.۹

- مورد اول
- مورد دوم

۲.۹

۱. مورد شماره ۱
۲. مورد شماره ۲

۱۰

در این قسمت با نحوه ارجاع به سایر منابع آشنا می‌شوید:
به صفحه درس ارجاع داده می‌شود [۵۱].

۱۱ ضمیمه

برای آشنایی بیشتر با \LaTeX ، با جست و جو در اینترنت منابع مفیدی خواهید یافت.

منابع

[1] http://mrheidar.ir/courses/operating_system.html