

Database

By Dr. Taghinezhad

Scan for More Information:



SEVENTH EDITION

Database System Concepts



Website: ataghinezhad.github.io, Email: a0taghinezhad@gmail.com



Chapter 1: Introduction

Database System Concepts, 7th Ed.

©Silberschatz, Korth and Sudarshan

See www.db-book.com for conditions on re-use



Outline

- کاربردهای سیستم پایگاه داده
- هدف از سیستم‌های پایگاه داده
- نمایش داده (View of Data)
- زبان‌های پایگاه داده
- طراحی پایگاه داده
- موتور پایگاه داده
- معماری پایگاه داده
- کاربران و مدیران پایگاه داده
- تاریخچه سیستم‌های پایگاه داده

<https://ataghinezhad.github.io/>



سیستم‌های پایگاه داده

- **DBMS** شامل اطلاعات مربوط به یک کمپانی خاص است:
 - مجموعه‌ای از داده‌های مرتبط به هم.
 - مجموعه‌ای از برنامه‌ها برای دسترسی به داده‌ها.
 - محیطی که هم مناسب و هم کارآمد برای استفاده باشد.
- سیستم‌های پایگاه داده برای مدیریت مجموعه‌هایی از داده‌ها استفاده می‌شوند که:
 - بسیار ارزشمند هستند.
 - نسبتاً بزرگ هستند.
 - توسط چندین کاربر و برنامه، اغلب به طور همزمان، دسترسی می‌شوند.
- یک سیستم پایگاه داده مدرن یک سیستم نرم‌افزاری پیچیده است که وظیفه آن مدیریت یک مجموعه بزرگ و پیچیده از داده‌ها است.
- پایگاه‌های داده تمام جنبه‌های زندگی ما را تحت تأثیر قرار می‌دهند.



نمونه‌های کاربرد پایگاه داده

■ اطلاعات سازمانی

- فروش: مشتریان، محصولات، خریده‌ها.
- حسابداری: پرداخت‌ها، دریافت‌ها، دارایی‌ها.
- منابع انسانی: اطلاعات در مورد کارمندان، حقوق، مالیات بر حقوق.

■ تولید: مدیریت تولید، موجودی، سفارشات، زنجیره تأمین.

■ بانکداری و امور مالی

- اطلاعات مشتری، حساب‌ها، وام‌ها و تراکنش‌های بانکی.
- تراکنش‌های کارت اعتباری.
- امور مالی: خرید و فروش ابزارهای مالی (مانند سهام و اوراق قرضه)؛ ذخیره داده‌های بازار در زمان واقعی.

■ دانشگاه‌ها: ثبت‌نام، نمرات

<https://ataghinezhad.github.io/>



نمونه‌های کاربرد پایگاه داده (ادامه)

- خطوط هوایی: رزرو، برنامه‌ها.
- **مخابرات:** سوابق تماس‌ها، پیامک‌ها و مصرف داده، تولید صورت حساب‌های ماهانه، حفظ موجودی کارت‌های اعتباری پیش‌پرداخت.
- **خدمات مبتنی بر وب**
 - خرده‌فروشان آنلاین: پیگیری سفارش، توصیه‌های شخصی‌سازی شده.
 - تبلیغات آنلاین.
- **پایگاه‌های داده اسناد (Document databases)**
- **سیستم‌های ناوبری:** برای نگهداری مکان‌های مختلف مورد علاقه همراه با مسیرهای دقیق جاده‌ها، سیستم‌های قطار، اتوبوس‌ها و غیره.



هدف از سیستم‌های پایگاه داده

در روزهای اولیه، برنامه‌های پایگاه داده مستقیماً بر روی سیستم‌های فایل ساخته می‌شدند، که منجر به موارد زیر می‌شد:

- **تکرار و عدم سازگاری داده‌ها:** داده‌ها در چندین قالب فایل ذخیره می‌شوند که منجر به تکثیر اطلاعات در فایل‌های مختلف می‌شود.
- **دشواری در دسترسی به داده‌ها**
 - نیاز به نوشتن یک برنامه جدید برای انجام هر کار جدید.
- **انزوای داده (Data isolation)**
 - فایل‌ها و فرمت‌های متعدد.
- **مشکلات یکپارچگی (Integrity problems)**
 - محدودیت‌های یکپارچگی (مانند موجودی حساب < 0) به جای اینکه صراحتاً بیان شوند، "مدفون" در کد برنامه می‌شوند.
 - افزودن محدودیت‌های جدید یا تغییر محدودیت‌های موجود سخت است.



هدف از سیستم‌های پایگاه داده (ادامه)

■ اتمی بودن به‌روزرسانی‌ها (Atomicity of updates)

- ممکن است خطاها پایگاه داده را در یک حالت ناسازگار با به‌روزرسانی‌های جزئی انجام شده باقی بگذارند.
- مثال: انتقال وجه از یک حساب به حساب دیگر باید یا به طور کامل انجام شود یا اصلاً اتفاق نیفتد.

■ دسترسی همزمان توسط چندین کاربر

- دسترسی همزمان برای عملکرد مورد نیاز است.
- دسترسی‌های همزمان کنترل نشده می‌تواند منجر به عدم سازگاری شود.
- مثال: دو نفر به طور همزمان یک موجودی (مثلاً ۱۰۰) را می‌خوانند و آن را با برداشت پول (مثلاً ۵۰ برای هر کدام) به‌روزرسانی می‌کنند.

■ مشکلات امنیتی

- سخت است که به کاربر اجازه دسترسی به برخی از داده‌ها، و نه همه آن‌ها، داده شود.

سیستم‌های پایگاه داده راه‌حلهایی برای همه مشکلات فوق ارائه می‌دهند.



مثال پایگاه داده دانشگاه

- در این متن ما از یک پایگاه داده دانشگاهی برای توضیح همه مفاهیم استفاده خواهیم کرد.
- داده‌ها شامل اطلاعات در مورد:
 - دانشجویان
 - اساتید
 - کلاس‌ها
- نمونه‌های برنامه کاربردی:
 - افزودن دانشجویان، اساتید و دوره‌های جدید.
 - ثبت نام دانشجویان در دوره‌ها و تولید فهرست کلاس‌ها.
 - اختصاص نمرات به دانشجویان، محاسبه میانگین نمرات (GPA) و تولید کارنامه‌ها.



نمایش داده (View of Data)

- یک سیستم پایگاه داده مجموعه‌ای از داده‌های مرتبط به هم و مجموعه‌ای از برنامه‌ها است که به کاربران اجازه می‌دهد تا به این داده‌ها دسترسی پیدا کرده و آن‌ها را تغییر دهند.
- یک هدف اصلی از سیستم پایگاه داده ارائه یک نمای انتزاعی از داده‌ها به کاربران است.

○ مدل‌های داده (Data models)

- مجموعه‌ای از ابزارهای مفهومی برای توصیف داده‌ها، روابط داده‌ها، معنانشناسی داده‌ها و محدودیت‌های سازگاری.

○ انتزاع داده (Data abstraction)

- پنهان کردن پیچیدگی ساختارهای داده برای نمایش داده‌ها در پایگاه داده از کاربران از طریق چندین سطح.

<https://ataaghtezhad.github.io/>



مدل‌های داده

■ مجموعه‌ای از ابزارها برای توصیف:

- داده
- روابط داده
- معناشناسی داده
- محدودیت‌های داده

■ مدل رابطه‌ای (Relational model)

■ مدل داده موجودیت-رابطه (Entity-Relationship data model): عمدتاً برای طراحی پایگاه داده

■ مدل‌های داده مبتنی بر شیء (Object-based data models): شیء-گرا و شیء-رابطه‌ای

■ مدل داده نیمه‌ساخت‌یافته (XML) (Semi-structured data model)

■ سایر مدل‌های قدیمی‌تر:

- مدل شبکه‌ای (Network model)
- مدل سلسله‌مراتبی (Hierarchical model)

<https://ataghinezhad.github.io/>



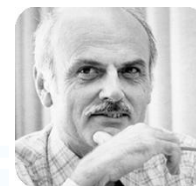
مدل رابطه‌ای

- تمام داده‌ها در جداول مختلف ذخیره می‌شوند.
- مثال داده‌های جدولی در مدل رابطه‌ای

Columns

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Rows



Ted Codd
Turing Award 1981

(a) The *instructor* table



یک نمونه پایگاه داده رابطه‌ای

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

<i>dept_name</i>	<i>building</i>	<i>budget</i>
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table



سطوح انتزاع

- **سطح فیزیکی (Physical level)** : توصیف می‌کند که یک رکورد (به عنوان مثال، استاد) چگونه ذخیره می‌شود.
- **سطح منطقی (Logical level)** : داده‌های ذخیره‌شده در پایگاه داده و روابط میان داده‌ها را توصیف می‌کند.

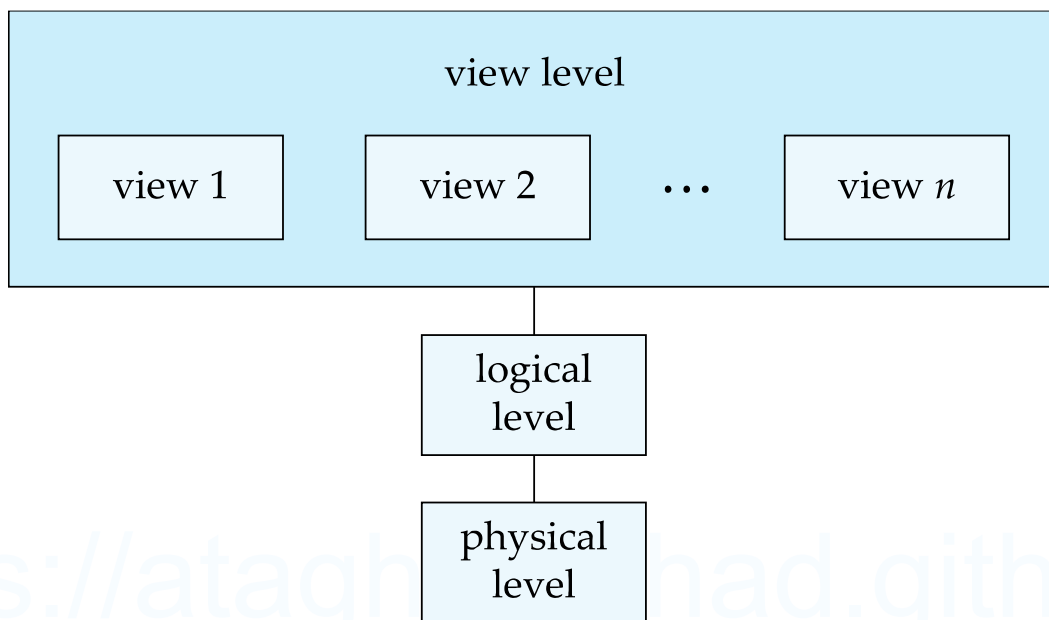
```
type instructor = record
    ID : string;
    name : string;
    dept_name : string;
    salary : integer;
end;
```

- **سطح نما (View level)** : برنامه‌های کاربردی جزئیات انواع داده را پنهان می‌کنند.
○ نماها همچنین می‌توانند اطلاعات (مانند حقوق کارمند) را برای اهداف امنیتی پنهان کنند.



نمای داده (View of Data)

یک معماری برای یک سیستم پایگاه داده



<https://ataghinezhad.github.io/>



نمونه‌ها و طرح‌ها (Instances and Schemas)

- شبیه به انواع (types) و متغیرها (variables) در زبان‌های برنامه‌نویسی.
- طرح منطقی (Logical Schema) : ساختار منطقی کلی پایگاه داده.
 - مثال: پایگاه داده شامل اطلاعاتی در مورد مجموعه‌ای از مشتریان و حساب‌ها در یک بانک و رابطه بین آن‌ها است.
 - مشابه اطلاعات نوع (type information) یک متغیر در یک برنامه.
- طرح فیزیکی (Physical schema) : ساختار فیزیکی کلی پایگاه داده.
- نمونه (Instance) : محتوای واقعی پایگاه داده در یک نقطه زمانی خاص.
 - مشابه با مقدار (value) یک متغیر.

<https://staghinezhad.github.io/>



استقلال فیزیکی داده (Physical Data Independence)

■ **استقلال فیزیکی داده:** توانایی اصلاح طرح فیزیکی بدون تغییر طرح منطقی.

○ برنامه‌های کاربردی به طرح منطقی وابسته هستند.

○ به طور کلی، واسطه‌های بین سطوح و مولفه‌های مختلف باید به خوبی تعریف شوند تا تغییرات در برخی قسمت‌ها به طور جدی بر دیگر قسمت‌ها تأثیر نگذارد.

<https://ataghinezhad.github.io/>



زبان تعریف داده (Data Definition Language - DDL)

■ نماد مشخصات برای تعریف شمای پایگاه داده.

■ مثال:

```
create table instructor (  
    ID          char(5),  
    name        varchar(20),  
    dept_name   varchar(20),  
    salary      numeric(8,2)  
)
```

■ کامپایلر DDL مجموعه‌ای از الگوهای جدول (table templates) را تولید می‌کند که در یک دیکشنری داده (data dictionary) ذخیره می‌شوند.

■ دیکشنری داده حاوی فراداده (metadata) است (یعنی داده‌ها در مورد داده‌ها).

○ طرح پایگاه داده.

○ محدودیت‌های یکپارچگی (Integrity constraints).

• کلید اصلی (ID) (Primary key) به طور منحصربه‌فرد اساتید را شناسایی می‌کند.

○ احراز هویت (Authorization).

• چه کسی می‌تواند به چه چیزی دسترسی داشته باشد.



زبان دستکاری داده - Data Manipulation Language (DML)

■ زبانی برای دسترسی و بهروزرسانی داده‌هایی که توسط مدل داده مناسب سازماندهی شده‌اند.

○ DML به عنوان زبان پرس و جو (query language) نیز شناخته می‌شود.

اساساً دو نوع زبان دستکاری داده وجود دارد:

■ **DML رویه‌ای (Procedural DML)** : از کاربر می‌خواهد مشخص کند که چه داده‌هایی مورد نیاز است و چگونه آن داده‌ها را بدست آورد.

■ **DML اعلانی (Declarative DML)** : از کاربر می‌خواهد مشخص کند که چه داده‌هایی مورد نیاز است بدون تعیین چگونگی بدست آوردن آن داده‌ها.

■ DML‌های اعلانی معمولاً یادگیری و استفاده آسان‌تری نسبت به DML‌های رویه‌ای دارند.

■ DML‌های اعلانی به عنوان DML‌های غیررویه‌ای (non-procedural DMLs) نیز شناخته می‌شوند.

■ بخشی از یک DML که شامل بازیابی اطلاعات است، زبان پرس و جو نامیده می‌شود.



زبان پرس و جو SQL

■ زبان پرس و جو SQL غیرویه‌ای است. یک پرس و جو چندین جدول (احتمالاً فقط یکی) را به عنوان ورودی می‌گیرد و همیشه یک جدول واحد را برمی‌گرداند.

■ مثال برای پیدا کردن همه اساتید در دپارتمان علوم کامپیوتر:

```
select name  
from instructor  
where dept_name = 'Comp. Sci.'
```

• SQL معادل یک ماشین تورینگ نیست.

■ برای اینکه بتوان توابع پیچیده را محاسبه کرد، SQL معمولاً در یک زبان سطح بالاتر تعبیه می‌شود.

■ برنامه‌های کاربردی عموماً از طریق یکی از موارد زیر به پایگاه‌های داده دسترسی پیدا می‌کنند:

○ افزونه‌های زبان برای اجازه دادن به SQL تعبیه‌شده (embedded SQL).

○ واسط برنامه کاربردی (Application program interface) مانند (ODBC/JDBC) که به پرس و جوهای SQL اجازه می‌دهد به پایگاه داده ارسال شوند.



دسترسی به پایگاه داده از برنامه کاربردی

- زبان‌های پرس و جو غیررویه‌ای مانند SQL به اندازه یک ماشین تورینگ جهانی قدرتمند نیستند.
- SQL از اعمالی مانند ورودی از کاربران، خروجی به نمایشگرها، یا ارتباط از طریق شبکه پشتیبانی نمی‌کند.
- چنین محاسبات و اعمالی باید در یک زبان میزبان (host language)، مانند C/C++، Java یا Python، با پرس و جوهای SQL تعبیه شده که به داده‌ها در پایگاه داده دسترسی پیدا می‌کنند، نوشته شوند.
- برنامه‌های کاربردی (Application programs) برنامه‌هایی هستند که برای تعامل با پایگاه داده به این شیوه استفاده می‌شوند.



طراحی پایگاه داده

فرآیند طراحی ساختار کلی پایگاه داده:

■ **طراحی منطقی (Logical Design):** تصمیم‌گیری در مورد طرح‌واره پایگاه داده. طراحی پایگاه داده مستلزم یافتن مجموعه‌ای "خوب" از طرح‌های رابطه (relation schemas) است.

○ تصمیم کسب و کار: چه صفاتی (attributes) را باید در پایگاه داده ثبت کنیم؟

○ تصمیم علوم کامپیوتر: چه طرح‌واره‌های رابطه‌ای باید داشته باشیم و چگونه صفات باید بین طرح‌واره‌های مختلف رابطه توزیع شوند؟

■ **طراحی فیزیکی (Physical Design):** تصمیم‌گیری در مورد چیدمان فیزیکی پایگاه داده.



موتور پایگاه داده (Database Engine)

- یک سیستم پایگاه داده به ماژول‌هایی تقسیم می‌شود که هر یک از مسئولیت‌های سیستم کلی را مدیریت می‌کنند.
- مولفه‌های عملیاتی (functional components) یک سیستم پایگاه داده را می‌توان به موارد زیر تقسیم کرد:
 - مدیر ذخیره‌سازی (The storage manager)
 - مولفه پردازشگر پرس و جو (The query processor component)
 - مولفه مدیریت تراکنش (The transaction management component)

<https://ataoghinezhad.github.io/>



مدیر ذخیره‌سازی (Storage Manager)

- یک ماژول برنامه‌ای است که واسطی بین داده‌های سطح پایین ذخیره شده در پایگاه داده و برنامه‌های کاربردی و پرس و جوهای ارسال شده به سیستم فراهم می‌کند.
- مدیر ذخیره‌سازی مسئول وظایف زیر است:
 - تعامل با مدیر فایل سیستم عامل (OS file manager).
 - ذخیره‌سازی، بازیابی و به‌روزرسانی کارآمد داده‌ها.
- مولفه‌های مدیر ذخیره‌سازی شامل:
 - مدیر احراز هویت و یکپارچگی (Authorization and integrity manager)
 - مدیر تراکنش (Transaction manager)
 - مدیر فایل (File manager)
 - مدیر بافر (Buffer manager)



مدیر ذخیره سازی (ادامه)

■ مدیر ذخیره سازی چندین ساختار داده را به عنوان بخشی از پیاده سازی سیستم فیزیکی پیاده سازی می کند:

○ فایل های داده (**Data files**) : خود پایگاه داده را ذخیره می کنند.

○ دیکشنری داده (**Data dictionary**) : فراداده در مورد ساختار پایگاه داده، به ویژه طرحواره پایگاه داده، را ذخیره می کند.

○ شاخص ها (**Indices**) : می توانند دسترسی سریع به اقلام داده را فراهم کنند.

○ یک شاخص پایگاه داده به آن اقلام داده ای اشاره می کند که یک مقدار خاص را نگه می دارند.



پردازشگر پرس و جو (Query Processor)

• مولفه‌های پردازشگر پرس و جو شامل:

○ **مفسر (DDL interpreter)** : دستورات DDL را تفسیر کرده و تعاریف را در دیکشنری داده ثبت می‌کند.

○ **کامپایلر (DML compiler)** : دستورات DML را در یک زبان پرس و جو به یک طرح ارزیابی (evaluation plan) ترجمه می‌کند که شامل دستورالعمل‌های سطح پایینی است که موتور ارزیابی پرس و جو (query evaluation engine) می‌فهمد.

• کامپایلر DML بهینه‌سازی پرس و جو (query optimization) را انجام می‌دهد؛ یعنی کم‌هزینه‌ترین طرح ارزیابی را از بین گزینه‌های مختلف انتخاب می‌کند.

○ **موتور ارزیابی پرس و جو (Query evaluation engine)** : دستورالعمل‌های سطح پایین تولید شده توسط کامپایلر DML را اجرا می‌کند.



مدیریت تراکنش (Transaction Management)

• یک تراکنش (transaction) مجموعه‌ای از عملیات است که یک تابع منطقی واحد را در یک برنامه پایگاه داده انجام می‌دهد.

• مولفه مدیریت تراکنش اطمینان حاصل می‌کند که پایگاه داده علی‌رغم خرابی‌های سیستم (مانند قطع برق و خرابی‌های سیستم عامل) و خرابی‌های تراکنش، در یک حالت سازگار (صحیح) باقی می‌ماند.

• مدیر کنترل همروندی (Concurrency-control manager) تعامل بین تراکنش‌های همزمان را کنترل می‌کند تا از سازگاری پایگاه داده اطمینان حاصل شود.



معماری پایگاه داده

■ پایگاه‌های داده متمرکز (Centralized databases)

- یک تا چند هسته، حافظه مشترک.

■ مشتری-سرویس‌دهنده (Client-server)

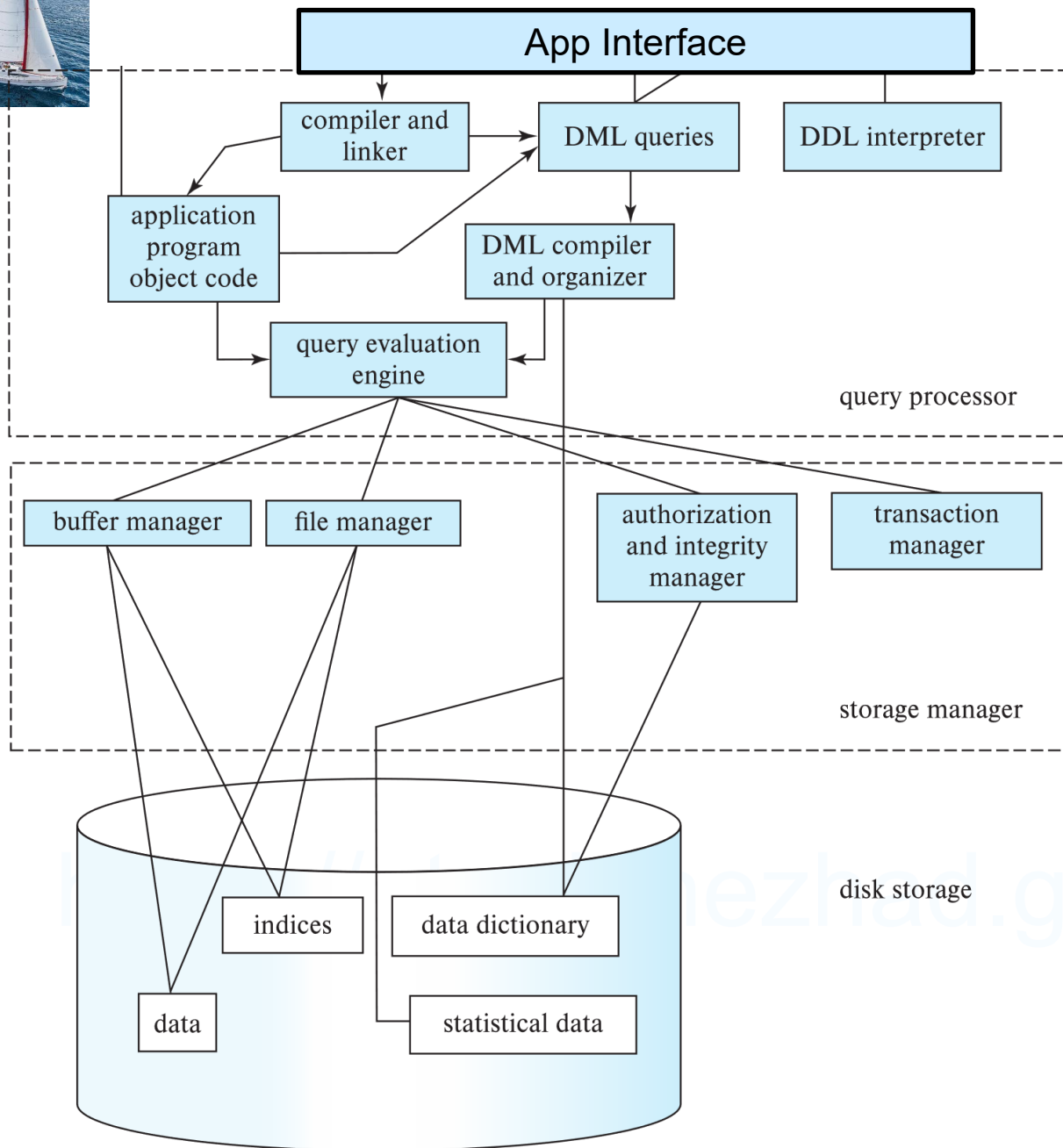
- یک دستگاه سرویس‌دهنده کار را به نمایندگی از چندین دستگاه مشتری اجرا می‌کند.

■ پایگاه‌های داده موازی (Parallel databases)

- حافظه مشترک چند هسته‌ای.
- دیسک مشترک (Shared disk).
- بدون اشتراک (Shared nothing).

■ پایگاه‌های داده توزیع شده (Distributed databases)

- توزیع جغرافیایی.
- عدم یکنواختی طرح‌واره/داده (Schema/data heterogeneity)



معماری پایگاه داده
(متمرکز/حافظه
مشترک)

ezhad.github.io/



برنامه‌های کاربردی پایگاه داده (Database Applications)

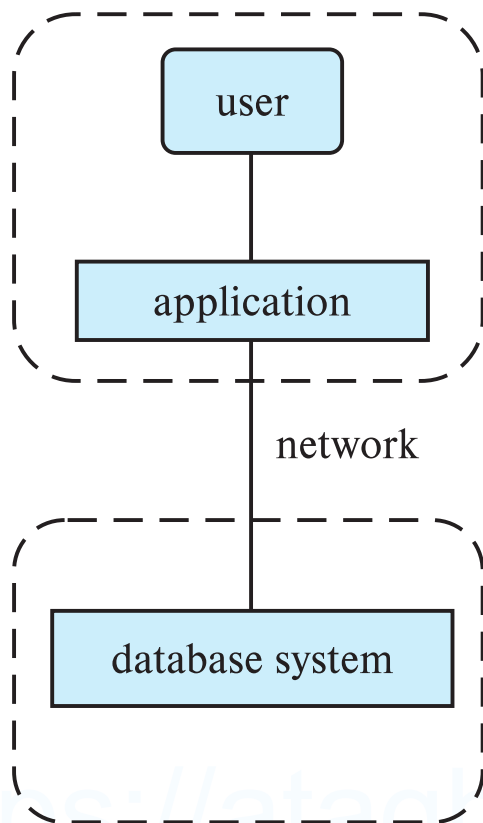
برنامه‌های کاربردی پایگاه داده معمولاً به دو یا سه قسمت تقسیم می‌شوند:

- معماری دو لایه (Two-tier architecture) : برنامه در دستگاه مشتری قرار دارد، جایی که عملکرد سیستم پایگاه داده را در دستگاه سرویس‌دهنده فراخوانی می‌کند.
- معماری سه لایه (Three-tier architecture) : دستگاه مشتری به عنوان یک جبهه (front end) عمل می‌کند و حاوی هیچ فراخوانی مستقیم پایگاه داده نیست.
 - مشتری با یک سرور برنامه کاربردی (application server) ، معمولاً از طریق یک واسطه فرم‌ها، ارتباط برقرار می‌کند.
 - سرور برنامه کاربردی به نوبه خود برای دسترسی به داده‌ها با یک سیستم پایگاه داده ارتباط برقرار می‌کند.

<https://ataghinezhad.github.io/>



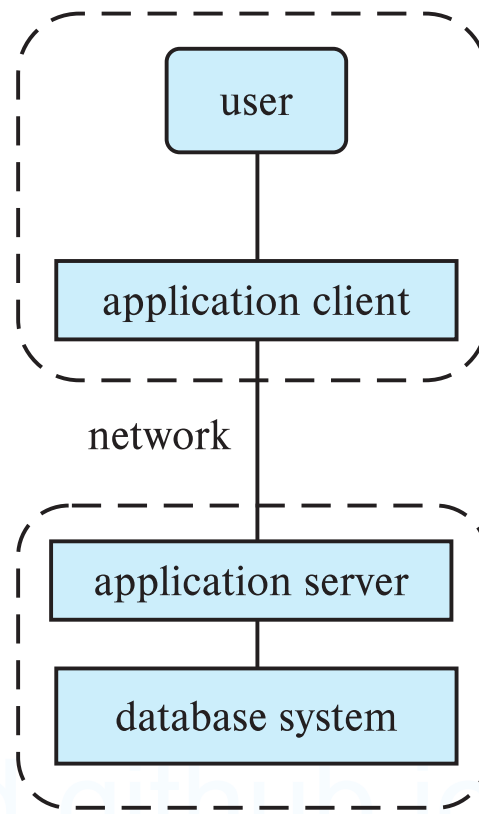
معماری‌های دو لایه و سه لایه



(a) Two-tier architecture

client

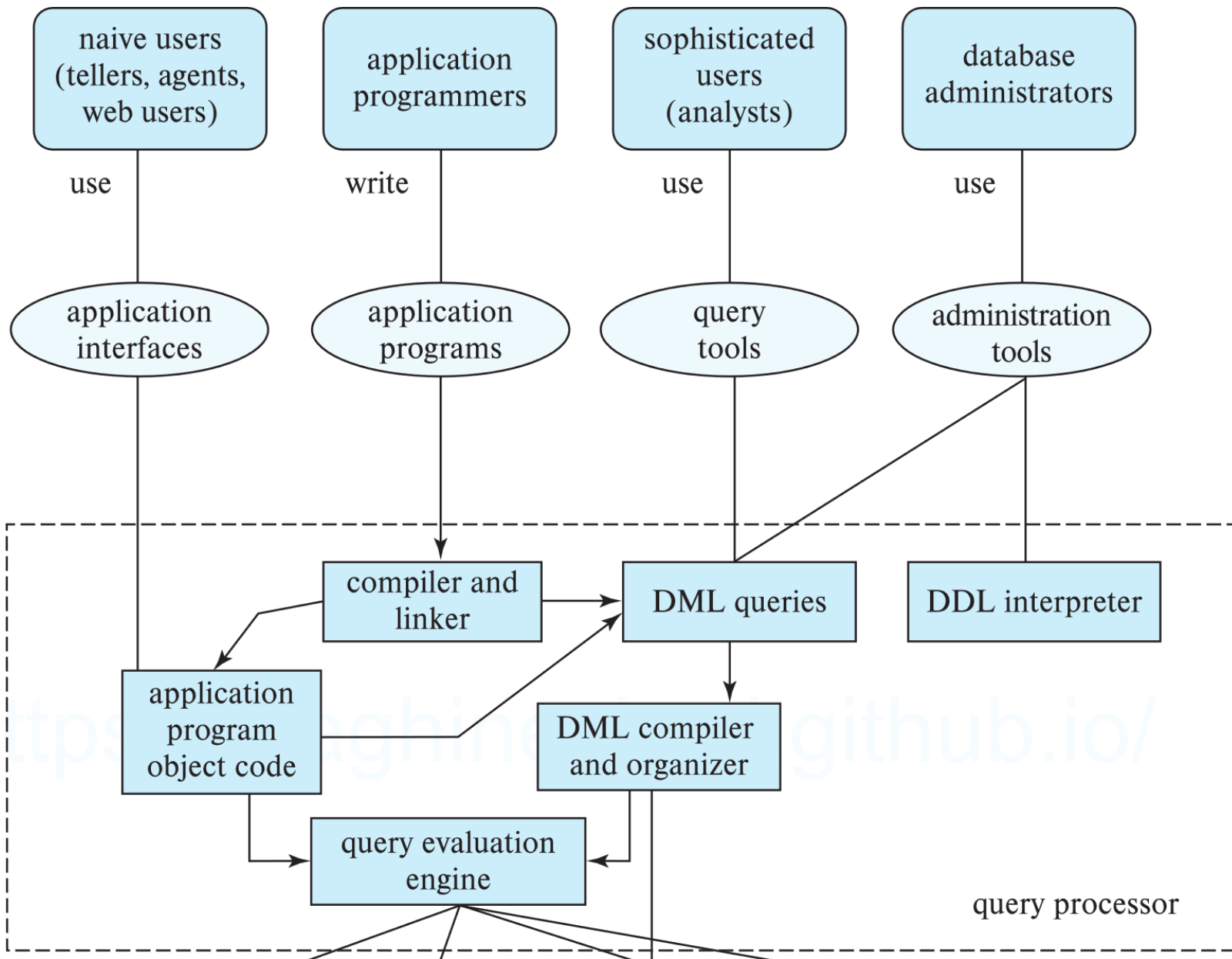
server



(b) Three-tier architecture



کاربران پایگاه داده





مدیر پایگاه داده (Database Administrator - DBA)

شخصی که کنترل مرکزی بر سیستم دارد، مدیر پایگاه داده (DBA) نامیده می شود. وظایف یک DBA شامل:

- تعریف طرح.
- تعریف ساختار ذخیره سازی و روش دسترسی.
- اصلاح طرح و سازمان فیزیکی.
- اعطای مجوز دسترسی به داده ها.
- نگهداری روتین:
 - پشتیبان گیری دوره ای از پایگاه داده.
 - اطمینان از در دسترس بودن فضای کافی دیسک آزاد برای عملیات عادی و ارتقاء فضای دیسک در صورت نیاز.
 - نظارت بر کارهایی که در پایگاه داده در حال اجرا هستند.



تاریخچه سیستم‌های پایگاه داده

■ دهه ۱۹۵۰ و اوایل دهه ۱۹۶۰:

- پردازش داده با استفاده از نوارهای مغناطیسی برای ذخیره‌سازی.
- نوارها فقط دسترسی ترتیبی (sequential access) را فراهم می‌کردند.
- کارت‌های سوراخ‌دار (Punched cards) برای ورودی.

■ اواخر دهه ۱۹۶۰ و دهه ۱۹۷۰:

- دیسک‌های سخت اجازه دسترسی مستقیم به داده‌ها را دادند.
- مدل‌های داده شبکه‌ای و سلسله‌مراتبی به طور گسترده‌ای مورد استفاده قرار گرفتند.
- تد کاد (Ted Codd) مدل داده رابطه‌ای را تعریف کرد.
- برای این کار برنده جایزه تورینگ ACM شد.
- IBM Research نمونه اولیه System R را آغاز کرد.
- UC Berkeley (Michael Stonebraker) نمونه اولیه Ingres را آغاز کرد.
- Oracle اولین پایگاه داده رابطه‌ای تجاری را منتشر کرد.
- پردازش تراکنش با عملکرد بالا (برای آن دوران).



تاریخچه سیستم‌های پایگاه داده (ادامه)

■ دهه ۱۹۸۰:

- نمونه‌های اولیه رابطه‌ای تحقیقاتی به سیستم‌های تجاری تکامل یافتند.
- SQL به استاندارد صنعتی تبدیل شد.
- سیستم‌های پایگاه داده موازی و توزیع شده.
- Teradata, IBM, Wisconsin.
- سیستم‌های پایگاه داده شیء-گرا.

■ دهه ۱۹۹۰:

- برنامه‌های کاربردی بزرگ پشتیبانی از تصمیم‌گیری (decision support) و داده کاوی (data-mining).
- انبارهای داده (data warehouses) بزرگ چند ترابایتی.
- ظهور تجارت وب.

<https://ataghinezhad.github.io/>



تاریخچه سیستم‌های پایگاه داده (ادامه)

■ دهه ۲۰۰۰:

- سیستم‌های ذخیره‌سازی داده‌های بزرگ (Big data storage systems).
- Google BigTable, Yahoo PNuts, Amazon.
- سیستم‌های "NoSQL".
- تجزیه و تحلیل داده‌های بزرگ: فراتر از SQL.
- Map reduce و موارد مشابه.

■ دهه ۲۰۱۰:

- SQL بارگذاری مجدد شد (SQL reloaded).
- جبهه (SQL front end) SQL به سیستم‌های Map Reduce.
- سیستم‌های پایگاه داده موازی گسترده (Massively parallel database systems).
- پایگاه‌های داده درون حافظه (main-memory databases) چند هسته‌ای.



End of Chapter 1

<https://ataghinezhad.github.io/>



تمرینات عملی (Practice Exercises)

- ۱.۱ این فصل چندین مزیت عمده سیستم پایگاه داده را توصیف کرده است. دو مورد از معایب آن چیست؟
- ۱.۲ پنج روش را فهرست کنید که در آن‌ها سیستم اعلام نوع (type declaration system) یک زبان مانند جاوا یا C++ با زبان تعریف داده‌ای که در پایگاه داده استفاده می‌شود، تفاوت دارد.
- ۱.۳ شش مرحله اصلی را که برای راه‌اندازی یک پایگاه داده برای یک بنگاه خاص انجام می‌دهید، فهرست کنید.
- ۱.۴ فرض کنید می‌خواهید یک سایت ویدیویی شبیه به یوتیوب بسازید.
هر یک از نکات ذکر شده در بخش ۱.۲ را به عنوان معایب نگهداری داده‌ها در یک سیستم پردازش فایل (file-processing system) در نظر بگیرید.
در مورد ارتباط هر یک از این نکات با ذخیره‌سازی داده‌های ویدیویی واقعی، و با فراداده‌ها در مورد ویدیو، مانند عنوان، کاربری که آن را بارگذاری کرده است، برچسب‌ها، و کاربرانی که آن را مشاهده کرده‌اند، بحث کنید.
- ۱.۵ پرس و جوهای کلمه‌کلیدی (Keyword queries) که در جستجوی وب استفاده می‌شوند، کاملاً با پرس و جوهای پایگاه داده متفاوت هستند.
- تفاوت‌های کلیدی بین این دو را، از نظر نحوه مشخص شدن پرس و جوها و از نظر نتیجه یک پرس و جو،



Exercises

- ۱.۶ چهار برنامه کاربردی که استفاده کرده‌اید و به احتمال زیاد از یک سیستم پایگاه داده برای ذخیره داده‌های ماندگار (persistent data) استفاده کرده‌اند، فهرست کنید .
- ۱.۷ چهار تفاوت مهم بین یک سیستم پردازش فایل و یک DBMS را فهرست کنید .
- ۱.۸ مفهوم استقلال داده فیزیکی (physical data independence) و اهمیت آن در سیستم‌های پایگاه داده را توضیح دهید .
- ۱.۹ پنج مسئولیت عمده یک سیستم مدیریت پایگاه داده (database-management system) را فهرست کنید . برای هر مسئولیت، مشکلاتی را که در صورت عدم انجام آن مسئولیت ایجاد می‌شود، توضیح دهید .
- ۱.۱۰ حداقل دو دلیل را فهرست کنید که چرا سیستم‌های پایگاه داده از دستکاری داده‌ها با استفاده از یک زبان پرس و جو اعلانی مانند SQL ، به جای فقط ارائه یک کتابخانه از توابع C یا C++ برای انجام دستکاری داده‌ها، پشتیبانی می‌کنند .
- ۱.۱۱ فرض کنید دو دانشجو در تلاش برای ثبت‌نام در یک درسی هستند که تنها یک صندلی خالی دارد . کدام مولفه از یک سیستم پایگاه داده مانع از این می‌شود که به هر دو دانشجو آن صندلی آخر داده شود؟
- ۱.۱۲ تفاوت بین معماری‌های برنامه دو لایه و سه لایه را توضیح دهید. کدام یک برای برنامه‌های کاربردی وب مناسب‌تر است؟ چرا؟
- ۱.۱۳ دو ویژگی توسعه یافته در دهه ۲۰۰۰ را فهرست کنید که به سیستم‌های پایگاه داده کمک می‌کنند تا حجم کاری تجزیه و تحلیل داده (data-analytics workloads) را مدیریت کنند .
- ۱.۱۴ توضیح دهید که چرا سیستم‌های NoSQL در دهه ۲۰۰۰ ظهور کردند و به طور خلاصه ویژگی‌های آن‌ها را با سیستم‌های پایگاه داده سنتی مقایسه کنید .
- ۱.۱۵ حداقل سه جدولی را توصیف کنید که ممکن است برای ذخیره اطلاعات در یک سیستم شبکه‌های اجتماعی مانند فیس‌بوک استفاده شود.