# “Capítol 1: Introducció”

Fitxers i bases de dades

# Sistema de tractament de bases de dades (DBMS)

- DBMS conté informació sobre una empresa particular
  - Col·lecció de dades relacionades
  - Conjunt de programes per accedir a les dades
  - Un entorn *convenient* i *eficient* d'utilitzar
- Aplicacions de bases de dades
  - Finances: Transaccions bancàries
  - Aerolínees: Reserves, horaris
  - Universitats: Matrícules, horaris
  - Ventes: Clients, productes, compres
  - Minorista online: Seguiment de comandes, recomenacions personalitzades
  - Manufactures: Producció, inventari, comandes
  - Recursos humans: Registres d'empleats, salaris, impostos
- Les bases de dades poden ser molt grans
- Les bases de dades toquen tots els aspectes de les nostres vides

# Exemple base de dades universitària

- Exemples de programes
  - Afegir nous estudiants, professors i cursos
  - Registrar estudiants als cursos i generar llistes de classes
  - Assignar notes als estudiants, calcular les seves mitjanes i generar informes
- Al començament, les aplicacions de les bases de dades es van construir directament sobre els sistemes d'arxius

# Inconvenients d'utilitzar sistemes d'arxius per emmagatzemar dades

- Dades redundants i inconsistents
  - Multiples formats, duplicació de la informació en fitxers diferents
- Dificultat per accedir a les dades
  - Necessitat d'escriure un nou programa per dur a terme cada tasca nova
- Aïllament de les dades - fitxers i formats múltiples
- Problemes d'integritat
  - Integritat a les restriccions
    - Exemple: Nota ha de ser ¿ 0
  - Difícils d'afegir i canviar les existents

# Inconvenients d'utilitzar sistemes d'arxius per emmagatzemar dades (Cont.)

- Atomicitat d'actualitzacions
  - Les errades poden deixar la base de dades en un estat inconsistent amb actualitzacions parcials
  - Exemple: La transferència de fons des d'un compte ha de ser completa o que no es realitzi
- Accés simultani per diversos usuaris
  - Simultaneïtat d'accés necessari per construcció
  - Si no està controlada pot portar a inconsistències
    - Exemple: Dues persones llegint un compte amb un saldo (posem 100) i actualitzant-lo retirant diners (posem 50) al mateix temps
- Problemes de seguretat
  - Dificultat de proporcionar accés a tothom però no a totes les dades

Els sistemes de bases de dades donen solucions a tots els problemes anteriors

# Nivells d'abstracció

- **Nivell físic:** Descriu com un registre (exemple: client) és emmagatzemat.
- **Nivell lògic:** Descriu les dades emmagatzemades a la base de dades, i les relacions amb aquestes.

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

- **Nivell vista:** Els programes amaguen detalls del tipus de dades. Les vistes també poden amagar informació (tals com els salaris dels empleats) per motius de seguretat.

# Vista de les dades. View of Data

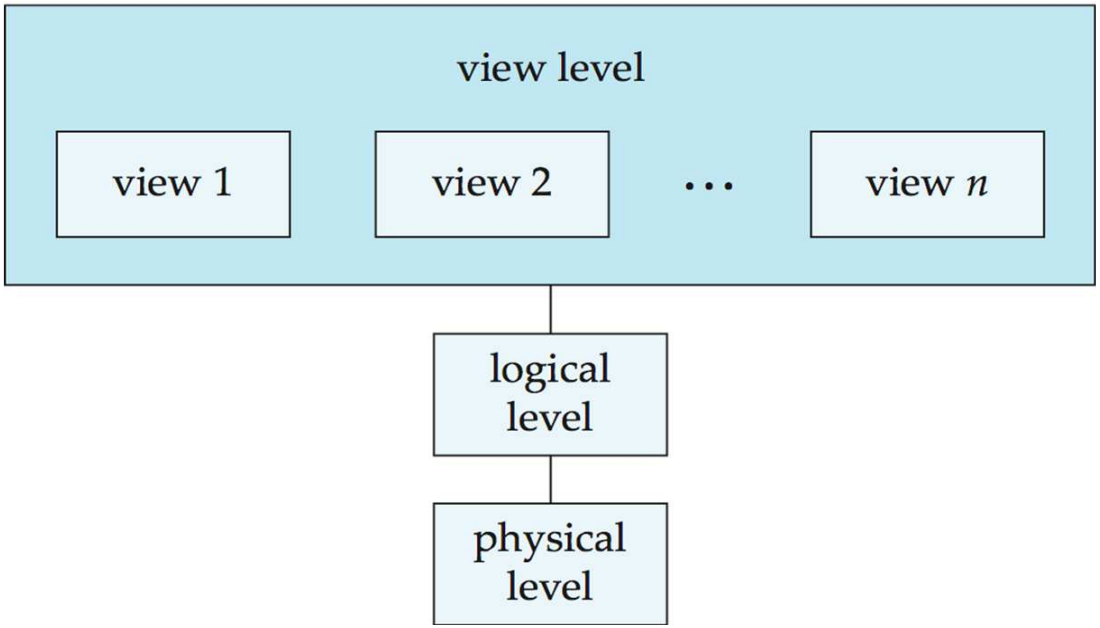


Figure: Exemple d'una arquitectura per a un sistema de base de dades.

# Casos i esquemes

- Similar als tipus i variables en llenguatges de programació
- **Esquema** - L'estructura lògica d'una base de dades
  - Exemple: Una base de dades d'informació sobre un conjunt de clients i els comptes i relacions entre ells
  - Anàleg al tipus d'informació d'una variable en un programa
  - **Esquema físic**: Diseny de la base de dades a nivell físic
  - **Esquema lògic**: Diseny de la base de dades a nivell lògic
- **Cas** - El contingut concret de la base de dades en un moment particular del temps
  - Anàleg al valor de la variable
- **Independènica Física de la dada** - L'habilitat de modificar l'esquema físic sense canviar l'esquema lògic
  - Les aplicacions depenen de l'esquema lògic
  - En general, les interfícies entre els nivells i components haurien d'estar ben definits de manera que els canvis a algunes parts no influeixin seriament en les altres.

# Models de dades

- Una col·lecció d'eines per descriure
  - Dades
  - Relacions de dades
  - Semàntica de dades
  - Restriccions de dades
- Model relacional
- Model de dada entitat-relació (principalment pel disseny de bases de dades)
- Models de dades basats en objectes (*Object-oriented* i *Object-relational*)
- Models semiestructurats de dades (XML)
- Altres models antics
  - Model de xarxa
  - Model jeràrquic

# Model relacional

- Model Rrlacional (Capítol 2)
- Exemple de dades tabulades en un model relacional

**Fitxer Pla**

| ID | name | dept_name | salary |
|---|---|---|---|
| 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 |

Table: Taula *instructor*

# Una mostra d'una base de dades relacional

| ID | name | dept_name | salary |
|---|---|---|---|
| 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 |

Table: Taula *instructor*

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

Table: Taula *department*

# Llenguatge de manipulació de dades (DML)

- Llenguatge per accedir i manipular les dades organitzades pel model apropiat
  - DML també conegut com llenguatge de consulta (query language)
  - SQL és el més utilitzat com a llenguatge de consulta
- Dues classes de llenguatges
  - **de Procediment** - Els usuaris especifiquen quines dades són necessàries i com aconseguir-les
  - **Declaratius (nonprocedural)** - Els usuaris especifiquen quines dades són necessàries sense especificar com aconseguir-les

# Llenguatge de Definició de Dades (DDL)

- Notació específica per definir l'esquema de la base de dades.
  Exemple:

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

- El compilador del DDL (*Data Definition Language*) genera un conjunt de formularis de taules guardades en un diccionari de dades
- El diccionari de dades conté metadades (és a dir, dades sobre dades)
  - Esquema de la base de dades
  - Restriccions d'integritat
    - Clau primària (ID únicament identifica cada casella de la taula instructors)
    - Integritat referencial (**references** constraint en SQL). Cada valor de *dept_name* ha de sortir a la seva taula.
  - Autorització

# SQL

- SQL: Llenguatge no procedimental més utilitzat
  - Exemple: Trobar el nom de l'instructor amb ID 22222

```
select   name
from     instructor
where    instructor.ID='22222'
```

  - Exemple: Trobar l'ID i els instructors de construcció del departament de Física

```
select   instructor.ID, department.building
from     instructor, department
where    instructor.dept_name=department.dept_name and
         department.dept_name='Physics'
```

- Normalment l'accés a les bases de dades mitjançant aplicatius són a través de
  - Extensions de llenguatge que permeten assignar SQL
  - Aplicatius d'interfície (i.e. ODBC/JDBC) els quals permeten cerques SQL per ser enviades a la base de dades
- Capítols 3,4 i 5.

# Disseny de la base de dades

El procés de dissenyar l'estructura general de la base de dades és:

- Disseny Lògic - Decidint l'esquema de la base de dades. El disseny de la base de dades requereix trobar una "bona" col·lecció de la relació d'esquemes.
  - Decisió empresarial - Quins atributs s'haurien de desar a la base de dades?
  - Decisió computacional - Quines relacions d'esquemes s'haurien de tenir i com els atributs s'haurien de distribuir a través de les relacions d'esquemes?
- Disseny físic - Decidint el disseny gràfic de la base de dades

# Disseny de la base de dades?

- Hi ha algun problema amb aquest disseny?

| ID | name | salary | dept_name | building | budget |
|---|---|---|---|---|---|
| 22222 | Einstein | 95000 | Physics | Watson | 70000 |
| 12121 | Wu | 90000 | Finance | Painter | 120000 |
| 32343 | El Said | 60000 | History | Painter | 50000 |
| 45565 | Katz | 75000 | Comp. Sci. | Taylor | 100000 |
| 98345 | Kim | 80000 | Elec. Eng. | Taylor | 85000 |
| 76766 | Crick | 72000 | Biology | Watson | 90000 |
| 10101 | Srinivasan | 65000 | Comp. Sci. | Taylor | 100000 |
| 58583 | Califieri | 62000 | History | Painter | 50000 |
| 83821 | Brandt | 92000 | Comp. Sci. | Taylor | 100000 |
| 15151 | Mozart | 40000 | Music | Packard | 80000 |
| 33456 | Gold | 87000 | Physics | Watson | 70000 |
| 76543 | Singh | 80000 | Finance | Painter | 120000 |

# Aproximació al disseny

- Teoria normalitzadora (Capítol 8)
  - Formalitzar quins dissenys són dolents, i provar-los
- Model Entitat-Relació (Capítol 7)
  - Models i empreses com a col·lecció d'*entitats* i *relacions*
    - Entitat: una "cosa" o "objecte" a l'empresa que es distingeix d'altres objectes

      - Descrits per un conjunt d'*atributs*
    - Relació: una associació a través de diverses entitats
  - Representar un gràfic mitjançant un diagrama *entitat-relació*

# El Model Entitat-Relació

- Models i empreses com a col·lecció d'*entitats* i *relacions*
  - Entitat: una "cosa" o "objecte" a l'empresa que es distingeix d'altres objectes
    - Descrits per un conjunt d'*atributs*
  - Relació: una associació a través de diverses entitats
- Representar un gràfic mitjançant un diagrama *entitat-relació*

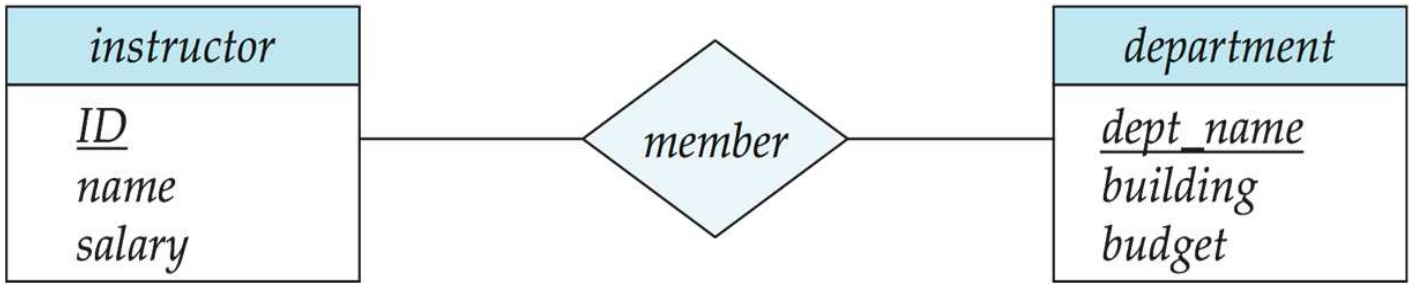


Figure: Diagrama *entitat-relació*.

Què ha passat amb dept_name de l'instructor i estudiant?

# Models de Dades Objecte-Relacionals

- Models relacional: plans, valors "atòmics"
- Models de dades Objecte Relacional
  - Extendre el model de dades relacional a partir d'incloure orientació d'objectes i construccions per tractar amb tipus de dades agregades
  - Permetre atributs de "tuples" per tenir tipus complexs, incloent valors no atòmics tal com relacions aniuades
  - Preservar les bases relacionals, en particular la declaració d'accés a les dades, mentre s'extendeix el poder de modelització
  - Proveir compatibilitat ascendent amb l'existència relacional de llenguatges

# XML: Extensible Markup Language

- Definit pel WWWConsortium (W3C)
- Originàriament previst com un llenguatge d'anotació de documents, no com un llenguatge de base de dades
- L'habilitat d'especificar noves etiquetes, i crear estructura d'etiquetes aniuades, feia de l'XML una bona manera d'intercanviar **dades**, no només documents
- L'XML s'ha convertit en la base per a les noves generacions d'intercanvi de formats de dades
- Una àmplia varietat d'eines són disponibles per analitzar, navegar i buscar documents/dades XLM

# Tractament d'emmagatzematge

- El gestor de magatzem és un programa que proveeix una interfície entre el baix nivell de dades emmagatzemades a la base de dades i els programes i consultes enviades al sistema
- El gestor de magatzem és el responsable de seguir les tasques següents:
  - Interactuar amb el gestor de fitxers
  - Emmagatzematge eficient, recuperant dades i actualitzant-les
- Problemes:
  - Accés a
  - Organització de fitxers
  - Indexant i "hashing"

# Processament de consultes

- Anàlisi i traducció
- Optimització
- Evaluació

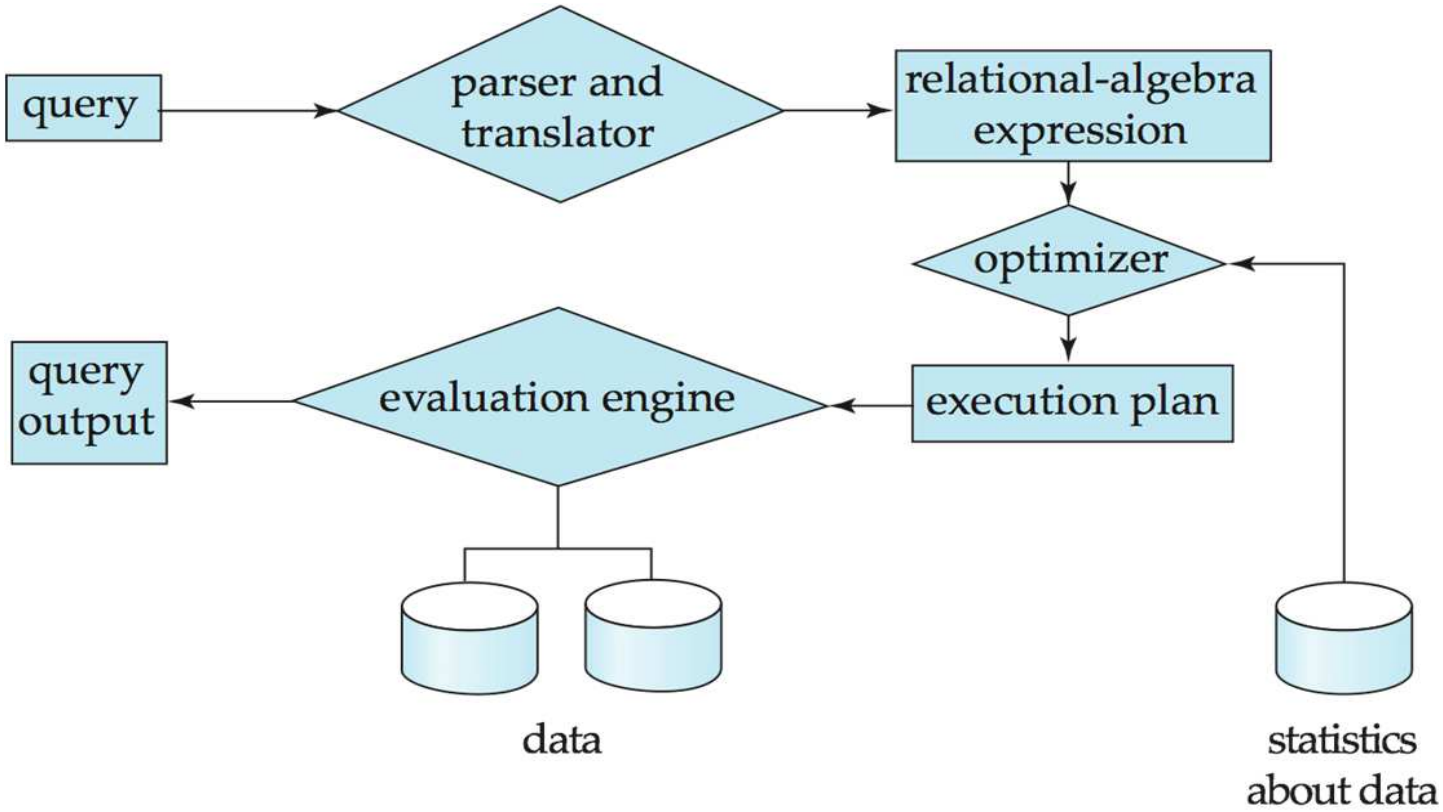


Figure: Exemple de processament de consultes.

# Processament de consultes (Cont.)

- Diferentes maneres d'evaluar una consulta feta
  - Expressions equivalents
  - Diferents algorismes per a cada operació
- El cost entre una bona manera i una manera dolenta d'evaluar una consulta pot ser enorme
- La necessitat d'estimar el cost operacional
  - Depèn d'una manera crítica de la informació estadística sobre relacions les quals la base de dades ha de mantenir
  - La necessitat d'estimar estadístiques per intermediar resultats a l'hora de calcular el cost d'expressions complexes

# Tractament transaccional

- Què passa si el sistema falla?
- Què passa si més d'un usuari està actualitzant alhora una mateixa dada?
- Una transacció és una col·lecció d'operacions que representa una única funció lògica en una aplicació de base de dades.
- El component de tractament transaccional assegura que la base de dades roman en un estat consistent (correcte) tot i les fallades del sistema (i.e. caigudes d'energia i caigudes del sistema operatiu) i les fallades transaccionals.
- El gestor Concurrency-control controla la interacció entre les transaccions concurrents, per assegurar la consistència de la base de dades.

# Usuaris i Administradors de la base de dades

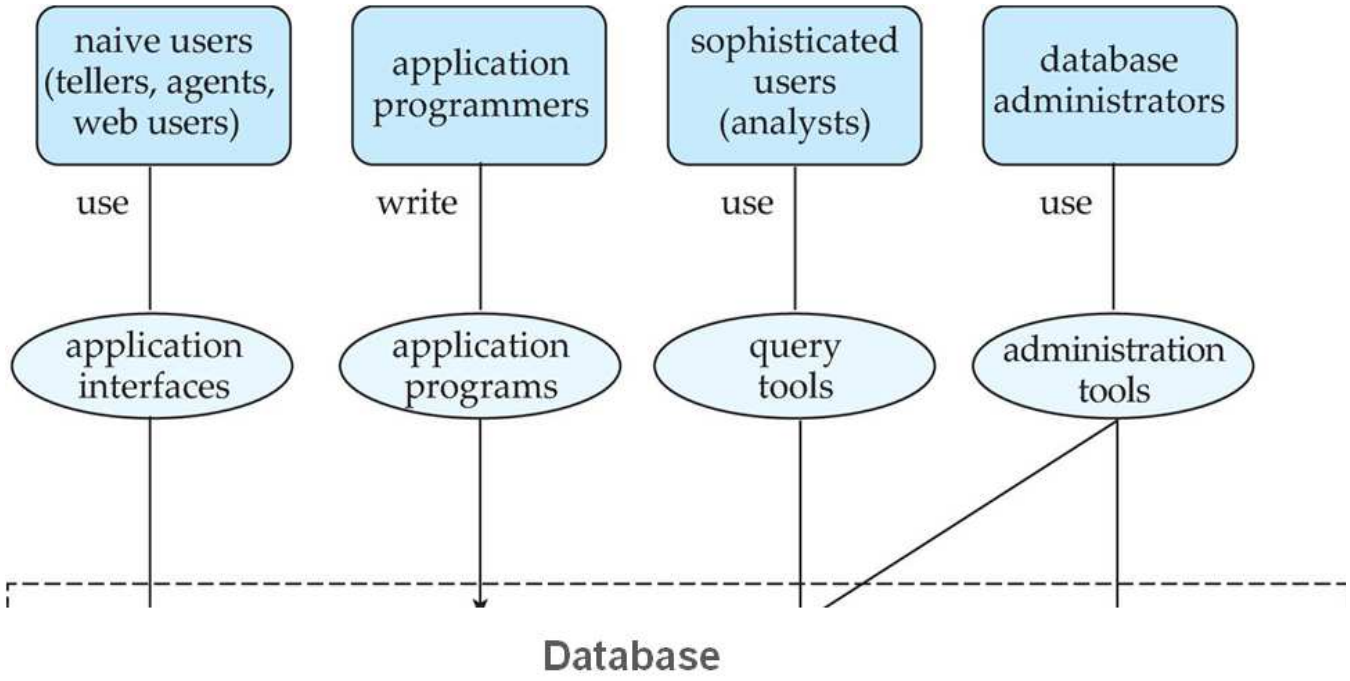


Figure: Exemple d'usuaris i administradors de bases de dades.

# Sistema intern de la base de dades

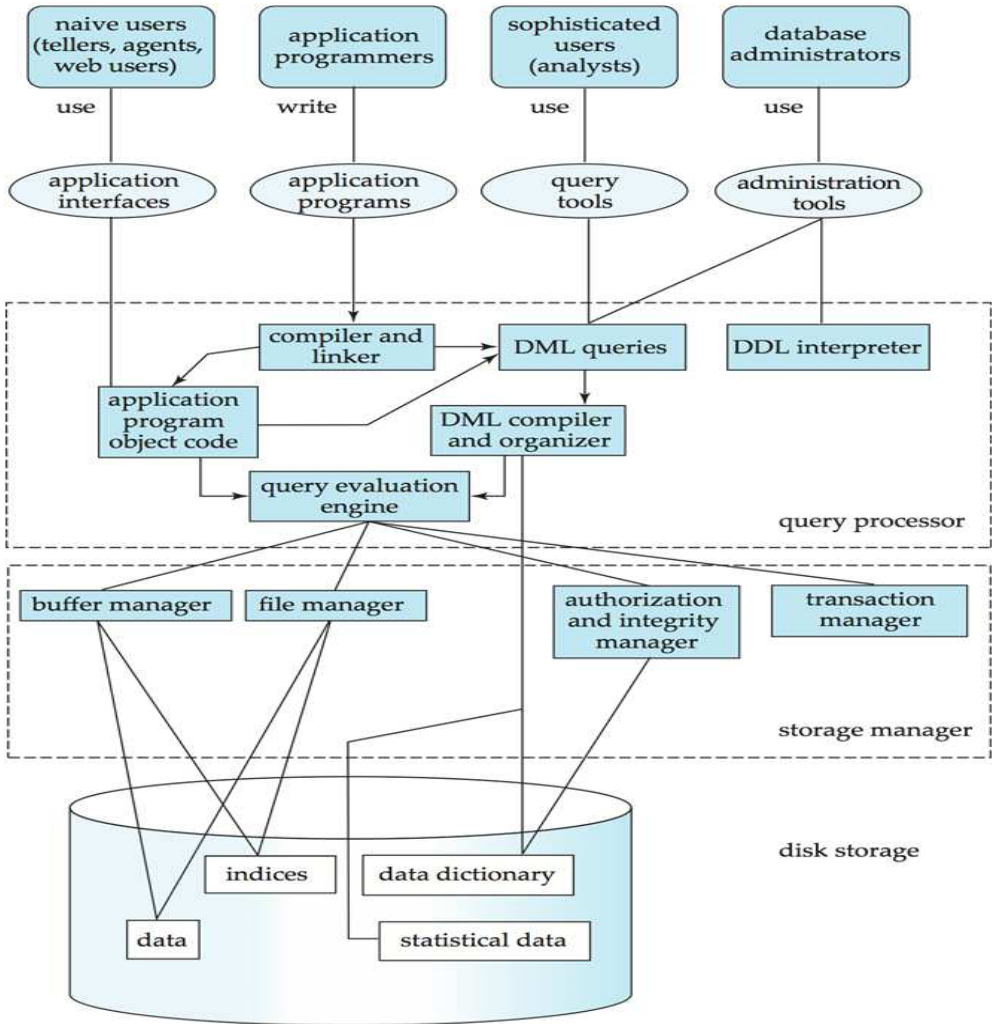


Figure: Exemple de sistemes interns de la base de dades.

# Arquitectura de la base de dades

L'arquitectura del sistema d'una base de dades és altament influenciat pel subjacent sistema computacional el qual la base de dades està funcionant:

- Centralitzat
- Servidor-Client
- Paral·lel (multiprocessador)
- Distribuït

# Història del sistema de les bases de dades

- La dècada del 1950 i principis del 1960:
  - Ãšs de cintes magnètiques per emmagatzemar dades
    - Les cintes només usen accés seqüencial
  - Targetes perforades per introduir les dades
- Finals del 1960 i dècada del 1970:
  - Els discs durs permeten l'accés directe a les dades
  - Xarxa i models de dades jeràrquics en usos extesos
  - Ted Codd defineix el model de dades relacional
    - Guanyaria el premi ACM Turing per aquesta feina
    - Inverstigadors d'IBM comencen amb el prototip System R
    - UC Berkeley comença el prototip Ingres
  - Alt compliment (per l'època) dels processos transaccionals

# Història del sistema de les bases de dades (Cont.)

- La dècada del 1980:
  - Prototips de cerca relacional desenvolupa en sistemes comercials
    - SQL es converteix en l'estàndard industrial
  - Sistemes de bases de dades paral·les i distribuïts
  - Sistemes de bases de dades d'objectes-orientats
- Dècada del 1990:
  - Suport de grans decisions i aplicacions data-mining
  - Grans magatzems multi-terabites de dades
  - Emergència de webs comercials
- Principis del 2000:
  - Estàndards XML i XQuery
  - Administració autònoma de la base de dades
- Finals del 2000:
  - Sistemes d'emmagatzematge de dades gegants
    - Google BigTable, Yahoo PNuts, Amazon, ...

# FINAL CAPÍTOL 1

# Figures

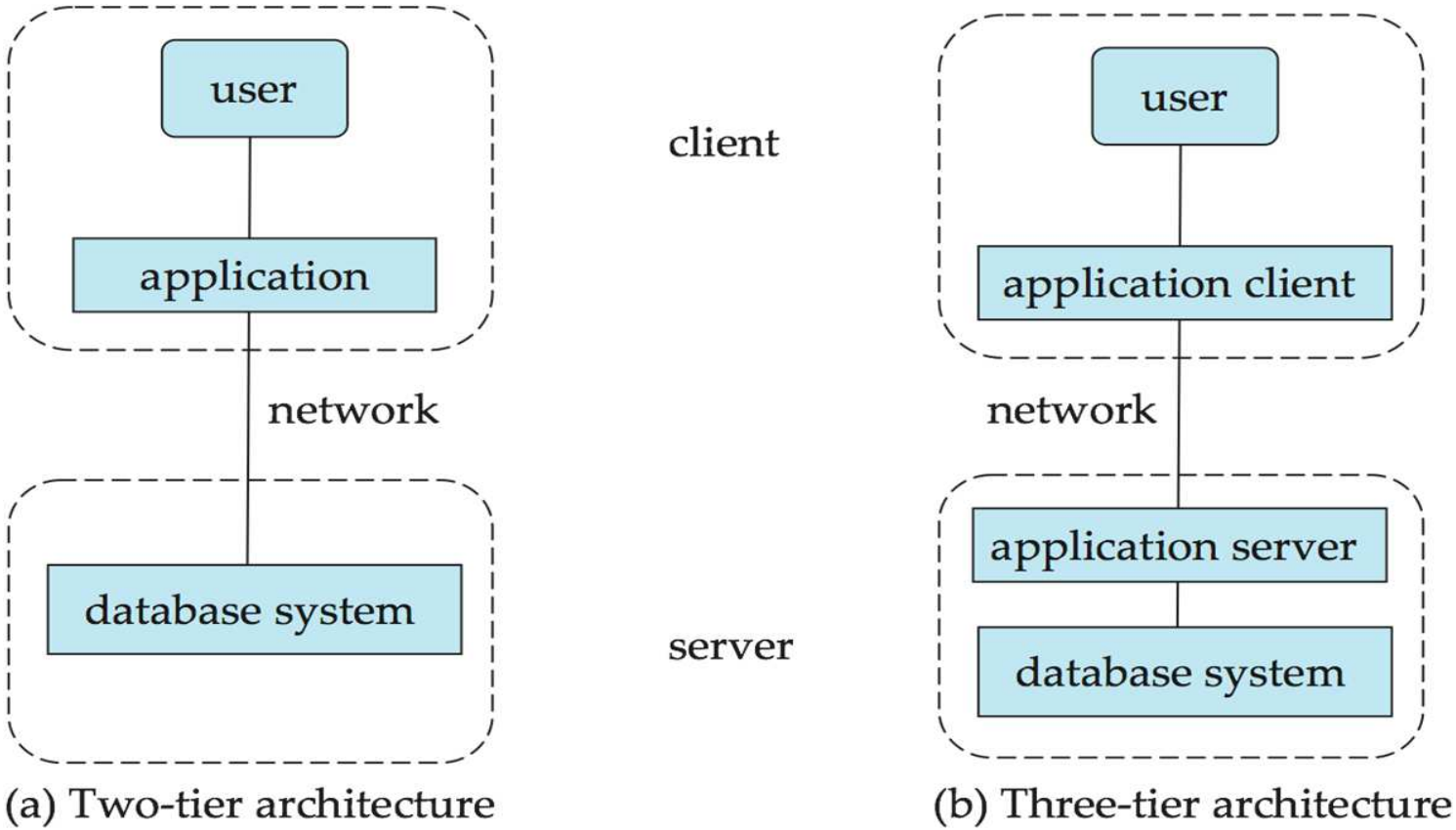


Figure: 1.01