CMPT 354
Database Systems I

Chapter 1 – Introduction

Course Textbooks

- Required Text:
  - Database Systems the Complete Book / Molina, Ullman and Widom, Prentice Hall 2002.
- Recommended Text:
  - Database Systems an application-oriented approach (second edition) / Kifer, Bernstein and Lewis, Addison Wesley 2004.
  - Database Management Systems (third edition) / Ramakrishnan and Gehrke, McGraw-Hill 2003.

Course Outline

- E-R Diagrams, UML Class modelling
- Relational Model, Normalization
- Relational Algebra
- SQL Language
- Constraints and Triggers
- Database Application Development
- Object Oriented and Object Relational Databases
- XML, XPath, XQuery, XML Schema
- Advance topics:
  - Data Mining
  - Multimedia Databases

Database Examples

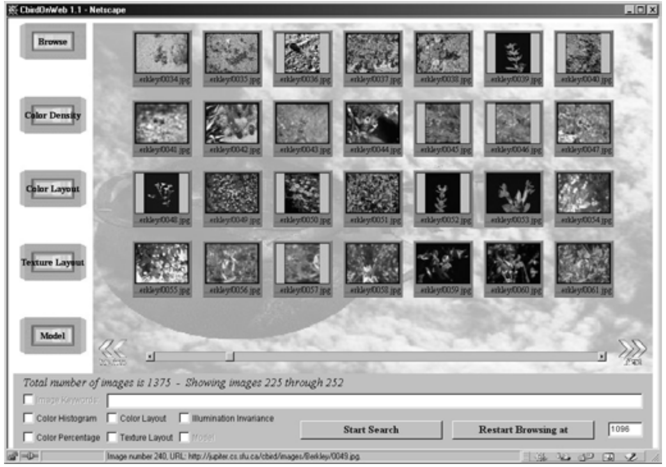
Simon Fraser University

123456789
Jane Doe

29345001054392

University systems

Bank transactions



CbirdOnWeb 1.1 - Netscape

Total number of images is 1375 - Showing images 225 through 252

□ Image Keywords:

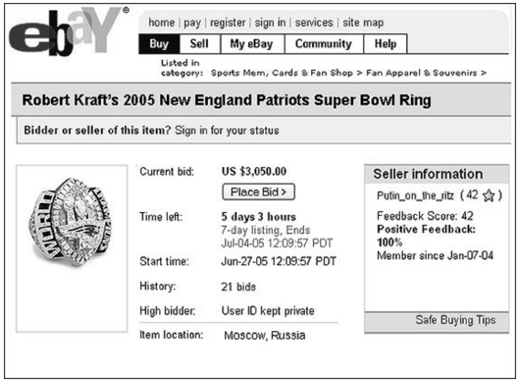
□ Color Histogram □ Color Layout □ Illumination Invariance

□ Color Percentage □ Texture Layout □ Model

1096

Image number 240, URL: http://jupiter.cs.sfu.ca/cbird/images/Berkley/0049.jpg

Multimedia web search



Listed in category: Sports Mem, Cards & Fan Shop > Fan Apparel & Souvenirs >

Robert Kraft's 2005 New England Patriots Super Bowl Ring

Bidder or seller of this item? Sign in for your status

Current bid: **US $3,050.00**

Time left: **5 days 3 hours**
7-day listing, Ends
Jul-04-05 12:09:57 PDT

Start time: Jun-27-05 12:09:57 PDT

History: 21 bids

High bidder: User ID kept private

Item location: Moscow, Russia

Seller information

Putin_on_the_ritz (42 ☆)

Feedback Score: 42
Positive Feedback: 100%
Member since Jan-07-04

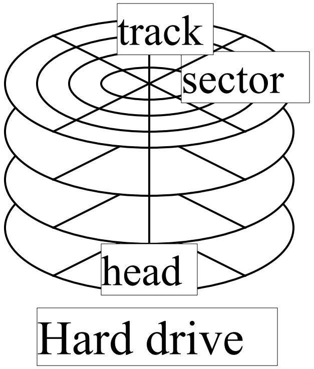
E-bay

The Need for Databases

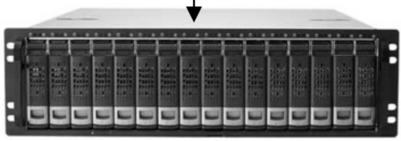
- Database - any conceptual collection of data that persists over long time.
- DBMS (Database Management System) – A system that facilitates Database operations.
  1. Persistent Storage: improving upon OS concepts for storing and accessing a large amount of data.
  2. Programming Interface: ADTs, and higher level languages.
  3. Transaction management: Ensure data modifications are allowed and handled properly.

Persistent Storage

- How are files stored on disks?
- How to search for database information on disk?



- Seek time is expensive.
- All probable data should be read together.
- FAT helps find files but not data, need to index data.


RAID Array

NAS (Network Attached Storage)

Programming Interface

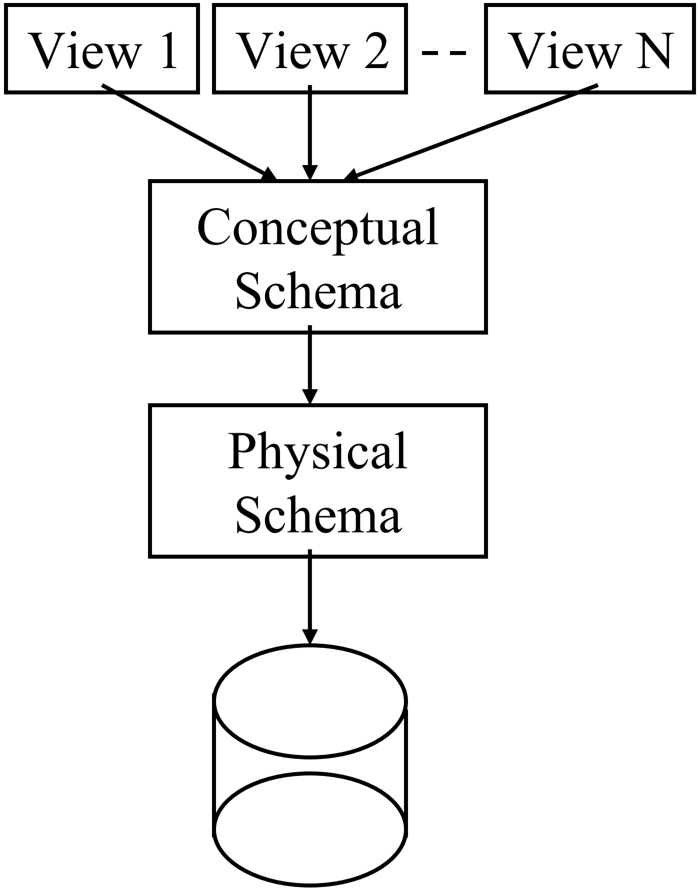
- What if a database object is added another attribute? What if the database system needs to be replaced altogether?
- How to let application designer reuse the DBMS?
- How to facilitate easy search of the database?
  - DDL (Data Definition Language)
  - DML (Data Manipulation Language) / Query Language
  - SQL Example:

```
Select balance
From bank_accounts
Where account_no = 10293847
```

Transaction Processing

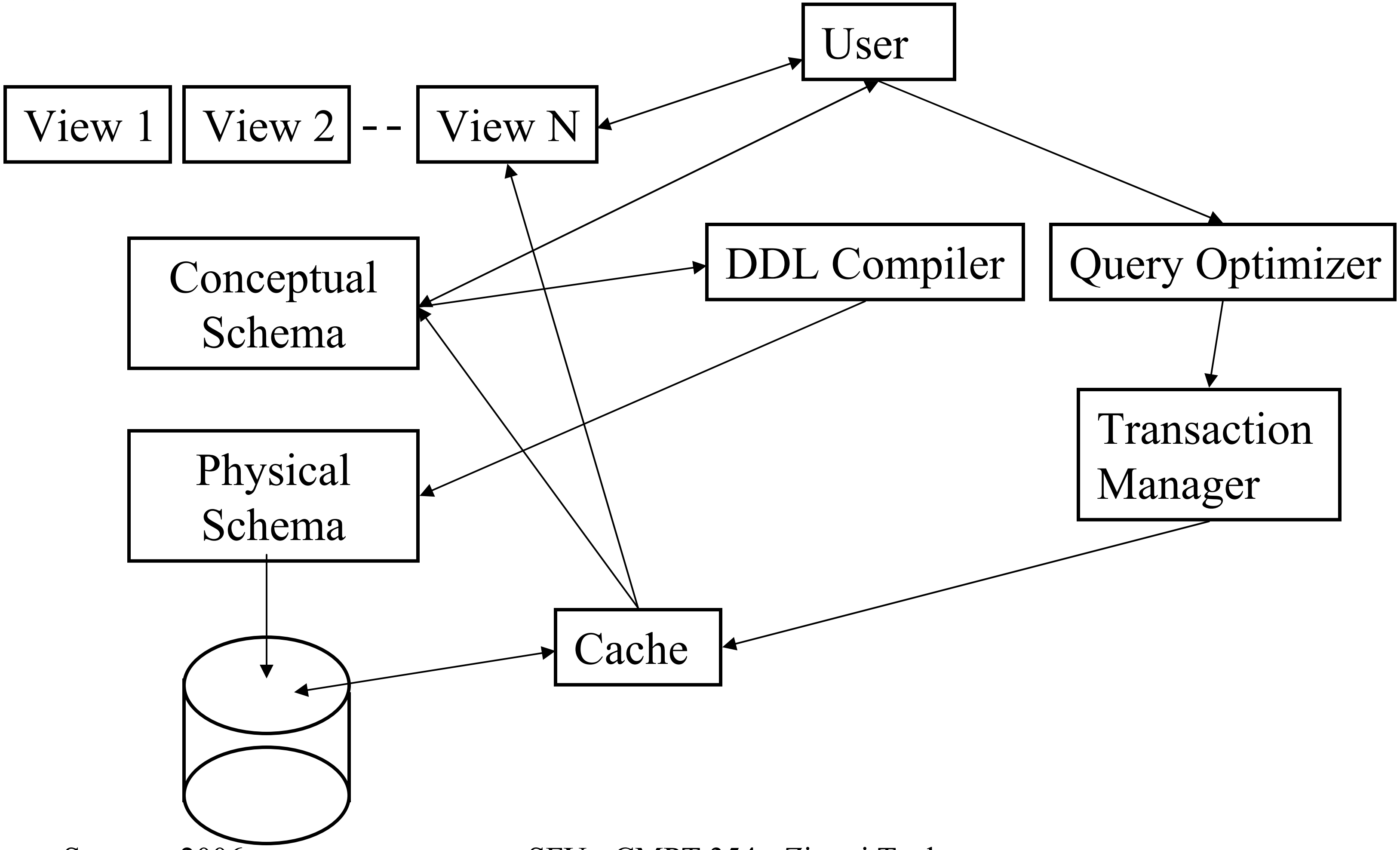
- How to handle multiple people trying to modify the database at once?
- How to make sure data in the system is consistent?
- How to answer queries efficiently?
- How to recover from system crashes?
- How to assure access only to authorized individuals?
- …….
  - DBMS are designed to handle all transactions issues automatically.
  - However, it is the responsibility of the DB designer and application programmer to program the DBMS properly.

Levels of Abstraction



- Views 1..N define some users' view of the data – External Schema.
- Conceptual Schema (or logical schema) – a logical design of the database, specified in DDL.
- Physical Schema – The actual data structures stored on disk, and indices for searching. User assists only.

DBMS System Overview



User

View 1

View 2

- -

View N

Conceptual Schema

DDL Compiler

Query Optimizer

Transaction Manager

Physical Schema

Cache

Database Design

Similarly to software engineering, a database design requires:

1. Requirements analysis.
2. High level conceptual design
   - Most commonly ER Diagram, but UML possible.
3. Logical relational schema design
   - Convert conceptual design to relational schema.
4. Physical level optimization and constraints specification
5. Assigning access permissions

Transactions Overview

- A transaction is a sequence of database operations performing *one logical operation*.
- Transactions must satisfy the ACID principles:
  - *Atomicity*: Either the entire transaction must be executed or non of it.
  - *Consistency*: After the transaction executed the database must remain in a **consistent state.**
  - *Isolation*: The transaction must perform as in a single user environment.
  - *Durability*: Completed transactions are never lost.

Consistency

- A Database is in consistent state (or simply consistent) if none of its data violates the system requirements, that is, the real-world allowable states.
- A transaction violating consistency is not allowed to *commit.*
- Main consistency checks - constraints:
  - Integrity (Uniqueness, Referential integrity)
  - Triggers and Assertions

Isolation

- Transaction isolation is a requirement that transaction results are not interfered by other concurrent transactions.
- Can accommodate multiple transactions using locks ⇒ transaction scheduling.
- Transaction isolation via scheduling leads to a serial order of transactions (but can still execute some in parallel).

Atomicity and Durability

- Durability – Must ensure that all committed transactions are never lost.
- Atomicity – A system crash during a transaction can have only partly updated database. Inconsistent and worse: wrong!
- A *transaction log* is used to record transactions
  - If transaction was not complete, undo all changes using the log – rollback.
  - If transaction was committed and lost, redo all changes using the log.