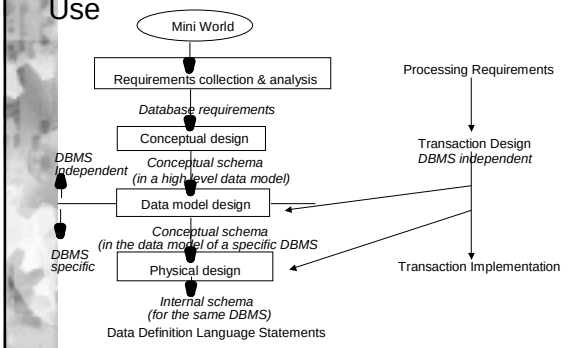
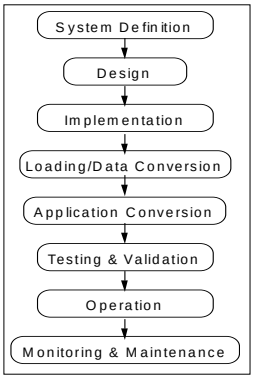
Design Considerations

CS2312

Conceptual Design includes Operational Use



Database Application Life Cycle



Conceptual design & Data model requirements

Conceptual design

- Complete understanding of database structure, semantics, constraints, relationships etc
- DBMS independent
- Stable description
- Database users and application users views; aids their understanding
- Communication with users

Data model requirements

- Expressive
- Simple
- Minimal
  - small number of basic concepts that are distinct and non-overlapping in meaning
- Diagrammatic
- Formal
  - accurate & unambiguous
- CONFLICTING REQUIREMENTS

Transaction Design

- Known transactions (applications) that will run on the database
- Database schema must include all information required by transactions
- Relative importance of transactions and expected rates of invocation important for performance tuning

Identify input/output & functional behaviour: 3 categories

- 1. Retrieval display/reports
  2. Update insert new data/modify old
  3. Mixed
- Transactions can be used to encapsulate integrity constraints

Transaction Design

- High level process specification technique
  data flow diagrams, process modelling etc
- Detailed design using programming techniques for loops, if statements etc
- Detailed design using set database operations
- Eight basic operations for updates on EER schema
  - insert entity, modify, delete entity
  - add, modify, remove relationship
  - add and remove from class
  - add and remove class

Transaction environment

- Pre-defined canned transactions
- A free-for-all using SQL directly

- Chiefly On-Line Transaction Processing (OLTP)
- Chiefly Management Information System (MIS)

- Multi-user or single-user
  - number of concurrent users—peaks, worst case, and average
  - potential conflicts—locking, timestamps
  - distributed transactions
- Integrity Checks
  - as updates made in transactions
  - batch run transaction

On-Line Transactions

ON-LINE	
DECISION SUPPORT	TRANSACTION PROCESSING
• Inquiry & Analysis	• Real time Transactions
• Retrieval Intensive	• Update Intensive
• Ad Hoc Queries	• Pre-Defined Transactions
• Unpredictable Query Complexity	• Transaction Throughput
	• Data Integrity
	• High Availability

Who is Using the Database?

- Users & Ease of Use
  - Who is the target end-user for queries and/or update transactions
- User Interfaces
  - graphical
  - forms-based
  - SQL
  - reports generated
  - menu-based
- Task analysis
- Work flows
- Views

Interfaces

- people
- software
- other databases
- hardware
- organisational processes

Housekeeping

- Backup & Archiving
  - on-line or off-line backups
  - size of backups
  - incremental vs dump
  - archiving strategy

- Security
  - passwords
  - permissions
  - views

Operational Considerations

Scope

- complete flexibility with 'bells and whistles'
- kernel activities

Model choice

- hierarchical / network / relational / object-oriented /object-relational

Software/Hardware

- Which database management system ?
- Configuration: e.g Unix server and PC front-ends?

Choice of DBMS

Costs

1. Software acquisition cost
2. Maintenance cost
3. Hardware acquisition cost
4. Database creation & conversion cost
5. Personnel cost
6. Training cost
7. Operating costs

- Data model depends on:
  - The structure and use of the data
  - Familiarity of the system
  - Available vendor services
    - communication software
    - data entry software
    - design and monitoring tools etc

Storage: Size and Volatility of data

- number of records (tuples)
- record (tuple) size
- growth potential
- volatility (growth/shrinkage)
- temporary space requirements

```
create table year
(yearno number(1) primary key,
yeartutorid number(4),
       yeartut_uk unique
       exceptions into bad_tutors
       using index
       not null
       constraint tut_fk
       foreign key (yeartutorid) references
                          staff(staffid))
       tablespace cags_course
       storage (initial 6144
                     next 6144
                     minextents 1
                     maxextents 5
                     pctincrease 5
                     pctfree 20);
```

Performance

- Query Profile
  - frequency of certain queries
  - hit rate on relations
  - certain relations used together
  - selection attributes
- Update Profile
  - dynamic or static
  - hit rate of certain updates
  - predictable—pre-fetch strategies

APPLICATION SPECIFIC
must know about queries, transactions & applications

- analysing DB queries and transactions
- analysing expected frequency of invocation of queries and transactions
- analysing time constraints of queries and transactions
- analysing expected frequency of update operations

Performance Measures

- Response time: how long will a query/update take ?
  - on average
  - at peak times: worst case
- Transaction throughput: how many transactions can be processed per second/millisecond
  - on average
  - at peak times: worst case
- How long will a report on the whole database take?
- Data take-on
- Analytical & experimental approaches

Benchmarks

1. Industry standard
   - external view of product;
   - samples performance on specific (simple) application;
   - meant for comparison across vendors
2. Vendor
   - identifying performance improvements
   - evolve with product
   - guide to development efforts & sales support
3. Customer-application
   - for important performance critical applications
   - vendors provided with benchmark by customer
   - high cost for customer
   - often rely on industry-standard measure

Industry Standard Benchmarks

"significant disk input/output, moderate system and application execution time, and transaction integrity"

The Transaction Processing Performance Council (TPC)

TPC-D:

- Debit/Credit Banking Application

Performance Metrics:

- Throughput transactions per second (tps)
- Response time of transaction (transaction elapse time)
- Cost metric $/tps
- OLTP multiple on-line terminal sessions—transaction arrival distribution. Wait time between requests is 'think time'

"a wide range of functions, provided over small to large databases"

- Not update-intensive
- Ad hoc queries
- Flexibility of query specification

Wisconsin

- Designed to produce predictable results

Performance Metrics:

- Response time of query (query elapse time)
- CPU & I/O utilisation
- Set Query
- average query throughput per minute & cost metric