To Use DB:

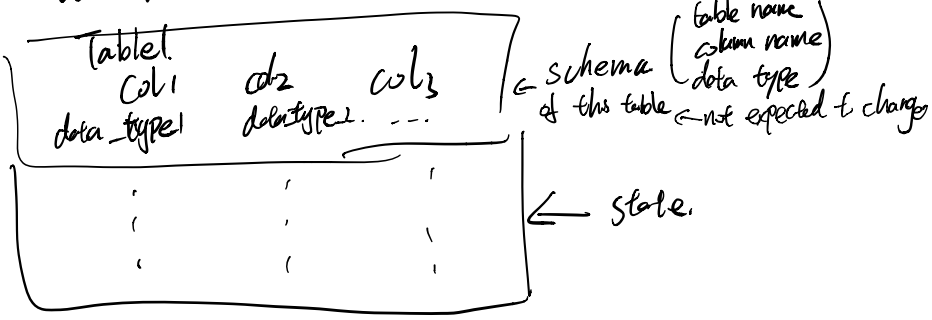
- Data intensive Apps
- Persistent storage
- Centralized control
- control of redundancy.
- Consistency & integrity
- Multiple user
- Shared data
- data documentation
- data independance
- Control of access & security
- backup & recovery

Not to Use:

- High cost.
- Generality
- real-time ✓.
- data & app are simple & stable.

Relational Model: (data is in tables)

SQL-like. : schema stable over time.



Constraint. constraint on state (individual data)
e.g. key: uniqueness constraint.

Operations: change, retrieve.

Desired Qualities:

Integrity: true data.
Consistency: no internal conflict.

Surrogate:

surrogate-based representation:

besides info cols, an extra column for unique identification

ANSI-SPARC:

3-Level DB Architecture (a database is divided to schema [describes intensions, types], and data [extension])

change data storage method w/o affecting application

storage of data (how data is organized physically)

(meaning of data)

use of data. applications

Database — internal schema —translate— conceptual schema — External schema 1 (apps)

translate — External schema 2 (how data is used) (apps)

External schema 3 (apps)

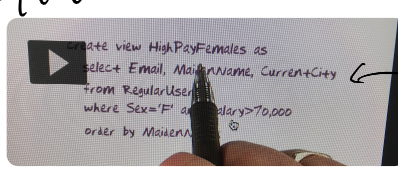
conceptual schema: conceptually relevant, general time-invariant structural aspect of reality.

Exclude: aspect of data representation and physical organization and access

e.g. in conceptual level, info available is column names & table name, but not sorting

External: Describe parts of the info in the conceptual schema in a form convenient to a particular user group's view

Derived from conceptual schema.

e.g. 

← create a view from conceptual

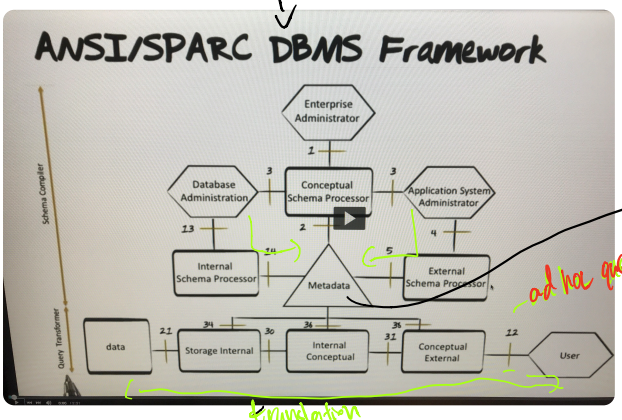
Physical data independence allow database admin to change internal schema w/o changing conceptual schema

Internal: Describe how info described in conceptual schema is physically represented to provide best performance

Physical Data independence: A measure of how much the internal schema can change w/o affecting the application

Logical Data independence A measure of how much the conceptual schema can change w/o affecting the application

Database management system



all schema information stored here.

ad hoc queries

translation

Meta data

Sys metadata (DBMS)

- where data come from
- how data were changed
- ~ stored
- ~ mapped
- who owns data
- who can access data
- usage history
- usage stats

Business Metadata (data warehouse):

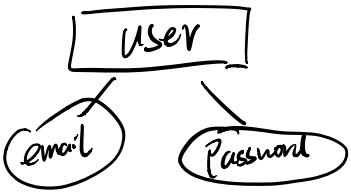
- what data are available.
- where data are located
- what ~ mean.
- how to access the data.
- Predefined reports
- Predefined queries
- How current the data are

Extended Entity Relational Model

Entity Type & entity surrogates

- entity type names must be unique

square

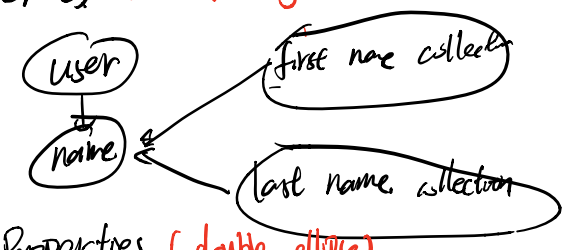


ellipse: single value property
single line ellipse means single value.

Identifying Properties: (underlying)

Each identifying property there is at most 1 identified entity
i.e. every entity must be uniquely referenceable.
e.g. an email address in G7Online web is an identifying property of a specific user (an entity)

Composite properties: a ellipse by 2 ellipse.



Multi-valued Properties: (double ellipse)

e.g. interest of user. (chess, math, reading...)

Relationship types (diamonds)

1-1 relationship type. (Partial function)

1-N relationship type. (total function)

~ Mandatory 1-N relationship type
elements → set. & ∀ element, must belong to a particular set

thick line

more to "many" side

N-M relationship type

N-ary relationship types (relationship with more entities)

(N>2)

. an incidence need several identifier.

e.g. club, student, team incidence.

Extended-Entity Relationship model

weak entity type: can not exist alone.

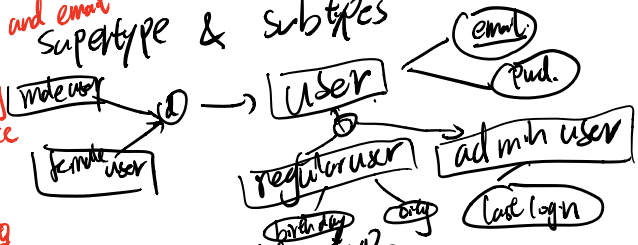
e.g. twitter, user & tweets time.

tweets time can not stand alone w/o user. However, we need user & tweet time to identify a particular tweet

recursive relationship

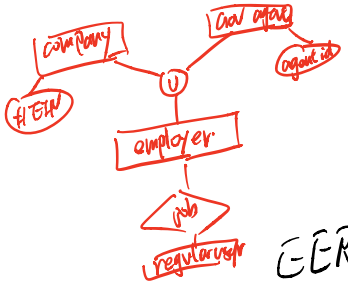
supertype & subtypes

note: user only has pid and email but regular user has pid, email, birthday sorry which is called inheritance



d: doesn't allow overlap

D: user can be both regular user/ admin user



Union entity type

e.g. employer can be company or Govt Agency

EIN Agency ID

EER: classification ✓ (define entity type supported)

aggregation not explicitly...

Generalization ✓ (sup-type super type)

no closed query language (a list of properties query result does not have type)

Relational Model

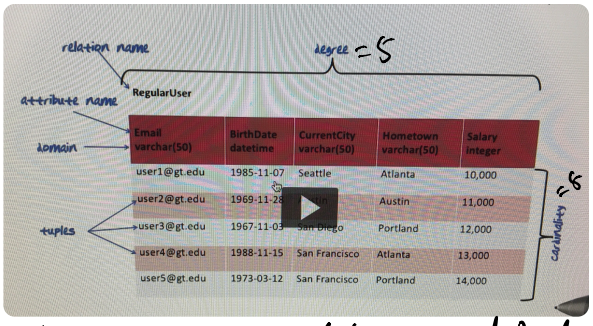
- Data structures
- Constraints
- Operation
  - Algebra
  - Calculus
    - tuple calculus sql
    - domain calculus (QBE)

Data structure:

a domain D: a set of atomic values (value has no meaning)

a relation R: a subset of set of ordered n-tuples
$R \subseteq (d_1, d_2 \ldots d_n \mid d_i \in D_i, i = 1..n)$

an attribute A: a unique name given to a domain in a relation. helping us interpret domain values



* The value of a relation is independent of attribute order & tuple order

Constraints:

Keys

Primary keys (Can't be null)

Entity integrity

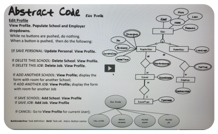
Referential integrity

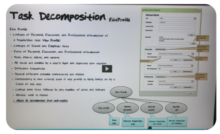
SPECIFICATION:

Task Decomposition:

Rule of Thumb:

- Lookup vs. insert, del & update.
- How many schema constructs are involved?
- Are enabling conditions consistent across tasks
- Are frequencies consistent across tasks
  (low frequency task vs. high frequency task)
- Is consistency essential?
- Is mother task control needed or not.


Abstract Code
Edit Profile
Edit Profile
View Profile. Populate School and Employer dropdowns.
While no buttons are pushed, do nothing.
When a button is pushed, then do the following:
(If SAVE PERSONAL: Update Personal. View Profile.
If DELETE THIS SCHOOL: Delete School. View Profile.
If DELETE THIS JOB: Delete Job. View Profile.
If ADD ANOTHER SCHOOL: View Profile; display the form with room for another School;
If ADD ANOTHER JOB: View Profile; display the form with room for another Job
If SAVE SCHOOL: Add School. View Profile
If SAVE JOB: Add Job. View Profile
If CANCEL: Go to View Profile for current User);
BoldUnderline: Task definition. Bold: Task call. Italics: Button names
View profile


Task Decomposition
EditProfile
Edit Profile:
- Lookups of Personal, Education, and Professional information of a RegularUser (task: View Profile)
- Lookups of School and Employer lists
- Edits of Personal, Education, and Professional information
- Read, insert, delete, and update
- All three are enabled by a user's login and separate edit request
- Different frequencies
- Several different schema constructs are needed
- Consistency is not critical, even if the profile is being looked at by a friend of the user
- Lookup done first followed by any number of edits and lookups
- Mother task is needed
- Must be decomposed into sub-tasks
Michael Bluth
View profile