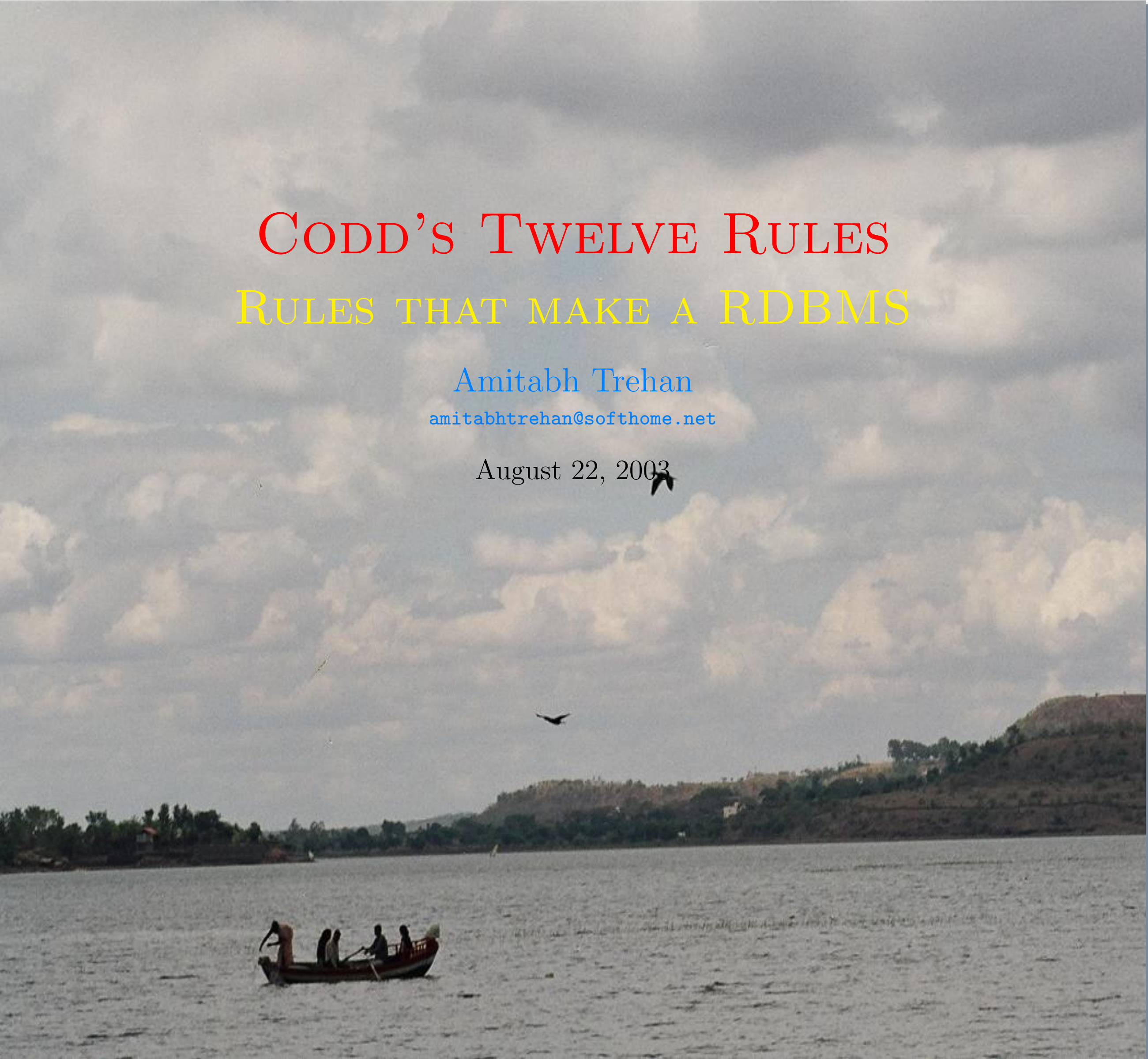
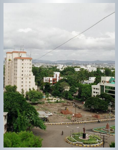


Codd's Twelve Rules

Rules that make a RDBMS

Amitabh Trehan
amitabhtrehan@softhome.net

August 22, 2003

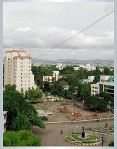


Home Page
Title Page
◀◀ ▶▶
◀ ▶

Go Back
Full Screen
Close
Quit

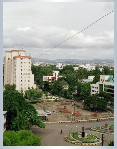
Codd's Rules

- 1985
- Proposed to test DBMSs for confirmation to concept of Codd's Relational model
- Hardly any commercial product follows all
- Oracle $= 8\frac{1}{2}$ out of 12.



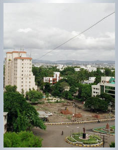
Rule Zero

- For a system to qualify as an RDBMS it must be able to manage its databases entirely through its Relational capabilities
- The other 12 rules derive from this rule



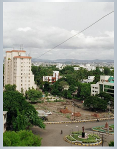
Rule 1: Information Rule

- All Information (inlcuding metadata) is to be represented as data stored in cells of tables.
- The rows and columns have to be strictly unordered.



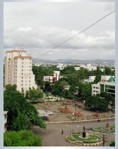
Rule 2: Guaranteed Access

- Each unique piece of data (atomic value) should be accesible by : TableName + Primary Key (Row) + Attribute (Column)
- *Violation:* Ability to directly access via pointers



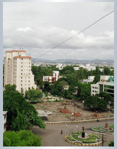
Rule3: Systematic treatment of NULL

- NULLs may mean: Missing data, Not applicable, No value
- Should be handled consistently - Not Zero or Blank
- Primary keys — Not NULL
- expressions on NULL should give NULL



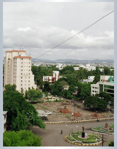
Rule4: Active On-Line Catalog

- Database dictionary (Catalog) to have description of the Database
- Catalog to be governed by same rules as rest of the database
- The same query language to be used on catalog as on the application database



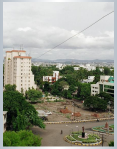
Rule5: Powerful language

- One well defined language to provide all manners of access to data
- Example: SQL
- If file supporting table can be accessed by any manner except a SQL Interface, then a violation



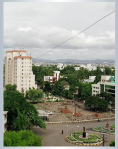
Rule6: View Updation Rule

- All views that are theoretically updatable should be updatable
- View = "Virtual table", temporarily derived from base tables
- Example: If a view is formed as join of 3 tables, changes to view should be reflected in base tables
- *Not updatable:* View does not have NOT-NULL attribute of base table
- Problems with computed fields in view e.g. Total Income = White income + Black income



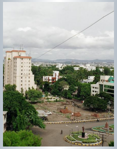
Rule7: Relational level operations

- There must be insert, update, delete operations at the level of Relations
- Set operations like Union,Intersection and Minus should be supported



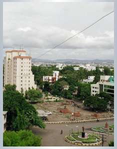
Rule8: Physical Data Independence

- The physical storage of data should not matter to the system
- If say, some file supporting table was renamed or moved from one disk to another, it should not effect the applications.



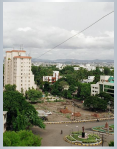
Rule9: Logical Data Independence

- If there is change in the logical structure (table structures) of the database the user view of the data should not change
- implemented through views. Say, if a table is split into two tables, a new view should give result as the join of the two tables
- Difficult rule to satisfy



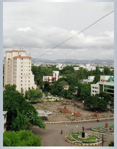
Rule10: Integrity Independence

- The database should be able to enforce its own integrity rather than using other programs
- Integrity rules = Filter to allow correct data, should be stored in Data Dictionary
- Key and check constraints, triggers etc should be stored in Data Dictionary
- This also makes RDBMS independent of front end



Rule11: Distribution Independence

- A database should work properly regardless of its distribution across a network
- This lays foundation of Distributed databases
- Similiar to Rule8 only that applies to distribution on a local Disk



Rule12: Nonsubversion Rule

- If low level access is allowed to a system it should not be able to subvert or bypass integrity rules to change data
- This may be achieved by some sort of locking or encryption
- Some low level access tools are provided by vendors that violate these rules for extra speed