SOFTENG351 Notes 2017

Theodore Oswandi

June 25, 2017

1 Fundamentals of Database Systems

1.1 General Information

Database large integrated collection of data.
Contains [Entities, Relationships]

DBMS (Database Management System):
software package to store and manage databases

Database System : DBMS with database

DBMS and uses

- store large amounts of information
- code for queries
- protect from inconsistencies and crashes
- security
- concurrent access

1.2 Why Databases

Need to shift from computation to storage of large amounts of information

Accomodate for changes in:
Variety: types of data
Velocity: movement of data
Veracity: uncertainty of data
Volume: amount of data

Structures/Models Need to have a model to describe data, and a schema used to give an abstract description of the data model

1.3 Levels of Abstraction

Views: describe how data seen
Logical Schema: how data structures organised (variable types)
Physical Schema: how files structured
Data Definition Language: How to define database schema
Data Manipulation: how to update values in database
Query Language: used to access data

1.4 Data Independence

Logical Data Independence

- external handling separate from logical organisation
- mappings change, not external schema
- applications only see external schema

Physical Data Independence

- changes to physical schema doesn't affect logical layer
- abstract from DBMS storage organisation
- can perform optimisation/tuning

1.5 Concurrency Control

- many users have to be able to access information at the smae time and make updates without negatively affecting database
- don't want to access disk lots. It is slow and inefficient
- let multiple users access and keep data consistent
- let users feel like they're the only ones using system

2 Relational Model of Data

2.1 General Information

- is logical model of data
- distinguish between data syntax and semantics
- simple and powerful
- sql based off this

2.2 Simple approach

- use tuples to store data
- relations are sets of these tuples
- tables to represent sets of data
- properties (columns) are called attributes
- attributes associated with domains (variable types)

2.3 Relational Schemata

Use of attributes creates relation schema such as:
MOVIE(title: *string*, production_year: *number*)

Relation Schema provide abstract description of tuples in relation

Database Schema is set S of relational schemata. Is basically the set of all tables and their attributes

2.4 Keys

Are used to uniquely identify tuples over all data in a given table.
They are used to restrict number of database instances, to something more realistic and identify objects efficiently

Superkey over relation schema is a subset of attributes that satisfies this uniqueness property
Key is a minimal superkey, is key if no other superkeys exist for R

Foreign Key: is a key used to index values from other values. Used to make reference between relational schemata.

- ensures referential integrity
- no need to copy info from other tables
- need to ensure that [x,y] $\subseteq$ [x,y] and not [y,x] (Order matters)

Example
MOVIE(title: string, production_year: number, director_id: number)
with key [title, production_year]

DIRECTOR(id: number, name: string)
with key [id]

with foreign key: MOVIE[director_id] $\subseteq$ DIRECTOR[id]

2.5 Integrity Constraints

- Db schema should be meaningful, and satisfy all constraints
- should stay true to keys, and foreign keys
- constraints should interact with each other correctly
- should process queries and update efficiently
- should do this and make as few comprimises as possible

3 SQL as Data Definition and Manipulation Language