CSC343 Introduction To Databases

sean.ryan

January 2023

1 Introduction

what is data? Bits that represent values e.g. numbers, strings, images, etc

in fact, instead of using "flat" files that store data with a keyword then data, we need a better system for more organized, optimized data.

Databases and DBMSs

- DBMS (Database Management System): a powerful tool for storing, addition, deleting, and simply operating on data in a fast and organized fashion
- Database:

Data Models

- Every DBMS is based on some data model: a notation for describing data including the structure of the data, constraints on the contents of the data (endpoints, all possible values), and operations on the data (specific searches, parameters, etc.).
- we will be mainly working with relational data model

The relational Data Model

- main concept is a "relation" derived from the mathematical concept, where a relation is a set of couples
- think of it as tables of rows and columns

What DBMS provides

- ability to specify the logical structure of data explicitly and have it enforced.
- ability to query or modify the data
- good performance under heavy loads
- durability of data, keeping it safe and intact. data integrity

- concurrent access by multiple users/processes (suppose tables A and B are bank accounts, we have a couple of queries that remove 100 from A and then deposit that amount in B. what happens if another user checks A before the 100 is removed and B after the 100 is deposited?)

Architecture of a relational DBMS

- the DBMS sits between the data and the users or between the data and an application program
- within the DBMS are layers of software:
  - parsing queries,
  - more but he turned the slides too quickly

——————— note: end of lecture 1 ———————

Relational Model Cont. A relation is a subset of what's called a Cartesian product of two domains, A and B. The operands are sets. The Cartesian product of two sets is every possible ordered tuple.

This means a relation could be just a single tuple, or the whole Cartesian product, as long as it is a subset.

for example, let A = p, q, r, s, B = $1, 2, 3$ and C = $100, 200$

R = $< q, 2, 100 >, < s, 3, 200 >$ is a relation on A, B, C as it is a subset that would belong to their Cartesian product

We can use this logic on data tables as well, grouping together data into tuples and putting them into relations.

- Schema: definition of the structure of the relation. For example, Teams have 3 attributes: name, home field, and coach. no two teams can have the same name.
- Notation for expressing a relation's schema. Teams(Name, Home field, Coach)
- Instance: particular data in the relation. Instances change constantly

Terminology:

- Relation (table) Attribute (column)
- tuple (row0
- arity of a relation, number of attributes
- cardinality of a relation, number of tuples

Relations are sets of tuples, meaning there can be no duplicate tuples and the order doesn't matter.

Database schemas and instances

- Database schema: a set of relation schemas , Database instance: a set of relation instances

A superkey is when the attributes of a tuple can occur at most once. basically, a unique tuple, for which another duplicate tuple can never exist.

We are mostly interested in a minimal set of attributes with the superkey property

minimal in the sense that no attributes can be removed from the superkey without making it no longer a superkey

Keys

- KEY: a minimal superkey
- in the schema, by convention we often underline a key
- aside: the term superkey is related to term superset