CS 338 course note

Chenxuan Wei

Sep 2022

Contents

1 Introduction to database

1. Terms
   Data redundancy: presence of duplicate data in multiple data files
   Data inconsistency: the same attribute may have different values

2. Database
   a collection of related information stored in a stuctured form

3. DBMS:
   a collection of programs that manipulate a database

4. Data Model

   - Relational Model
   - Object-oriented model
   - semi-structed data model
   - network model
   - Hierarchical model

5. Schema

   - Physical schema: database at physical level
   - logical schema: database at logical schema
   - External schema: database at external schema

2 Relational

1. Terms

    - attribute: each column with in a table
    - domain: all possible value of a attribute
    - Primiary key: a attribute in a row that must be unique in a table
    - Tuple: rows
    - Schema of a relation: definiton of a table
    - a instance: table content

2. Integrity Constaints
   is a condition that must be true for any instance of the database
   Domain constrain: must satisifeid domain
   Primary key constraints: each relation must have a primary key, and they must be unique
   Foreign key: set of filed in one relation used to refert to a tuple in another relation

3 Relational algebra and calculus

1. Relational Quesry language
   A major strengh of the relational model: supports simple, powerful querying of data

2. Relational algebra
   Result of a retrieval is new relation
   sqquence of relational algebra operations forms a relational algebra expression

3. Operations

   - selection (σ): select a subset of rows from relation
   - projection(π) deletes unwated columns from relation
   - cross-product(X) allows us combines 2 relation
   - Set-difference (-) tuples in relation1 but not 2
   - Union(Y) tuple in one of 1 or 2

   Format: (operation)$_{boolean}$ (relation)

4. Boolean
   used to show true value

5. Assignment operation
   $< -$ allowed to assign variable

6. Union compatible
   if 2 relation have the same degree and all attributes are defined on same domains

7. Foreign key
   Assume R1(ABC), R2(EFG) there is a FK: R1.A referrece R2.G
   the value of R1.A must be
   Null or unique in R2
   however, R2.G does not need to be PK

8. Rename operation (useless)
   format: $p_{(relation)}(relation)$ or $p_{(col,col)}(relation)$
   the first one rename relation, but the second one only rename column

9. Join operation
   symbol: $\bowtie$
   a combination of cross product and selection, notice must have different attributes name
   The following are the same:

   - $e < -R1XR2$
     result $< -\sigma_{bool}(e)$

- R1 (join)$_{bool}$(R2)

10. Natural join operation
    result $< -R1 * R2$
    Assume $R(ABC), S(AD)$, $R * S- > (ABCD)$
    will auto=same attributes, and combine attributes, also allowed same attribute name

11. Division Operation
    Assume $R1(r1_i)$, $R2(r2_i)$, $R1 \div R2 =$
    $(r1_i)$ such that $r1_i \notin R2$ and keep all tuple that all not included $r1_i$ appear in R2

12. Aggreation:
    $_{G_i}g_{f_i(A_i)}(E)$, allowed optional As to change the name of function F1
    function includes

    - avg
    - min
    - max
    - sum
    - count

4 SQL mannipulation

4.1 Data mainipulation

1. select basic format
   select (attribute) **from** (table) **where** (condition)
   if mutiple table selected, they will be cross prodected
   can use table**.**attribute to for duplicate column namess
   where, order by, group by,having must be in this order

2. rename
   can rename attribute name **AS**
   can give table temp name right after it's name

3. **distinct**
   a key word to eliminate duplicates in rows
   usage: **select distinct** (attributes)......

4. nested query
   when nest a table in from, must give the table a name
   when used in where, no need to give name

5. **join**
   usage: (table) **join** (table) **on** (condition (only equality))

6. **natural join**
   usage: (table) **natural join** (table)
   other join is the same by different name

7. **Like**
   compare text value in pattern
   % compare zero or more characters
   _ compare exactly one character

8. **IN** and **NOT IN**
   check if the attribute value is in the subsequence table

9. explicit sets
   like (1,2,3) for in and not in

10. **exsits**
    will return true if the table have atleast one row

11. **Unique/not unique**
    not supported in SQLite
    will check if there is any duplicate rows

12. **any** and **all**
    used with compare operation like (<)

13. **order by**
    sort result on one or more of attribute
    from small to big
    used desc to reverse

14. **group by**
    include grouping attributes
    if used, **select** (attribute) can only include aggregation function and groupting attributes

15. **having**
    is like use aggregation in where

16. **union** and **intersection**, minus
    (q1) union/intersect/except (q2),