SimpleSql: Our RDBMS

Jiachi Zhang

Qi Xie

Zheyi Wang

What we implement:

1. SQL Parser

The SQL operations we can parse include “create”, “alter”, “drop”, “select”, “insert”, “update”, “delete”, “create index” and “drop index”.

We split select sentence by key words and we assume that key words are in an pre-defined order (“select”, “from”, “join”, “where”, “group by”, “having”). Therefore, using wrong key word order is unacceptable in our parser.

Our parser parses condition sentences (condition after key word “where”) by key words “and” and “or”. We convert the condition sentence to a tree of “or” and “and”. The leaf node stores information of single condition. For example, “attribute1 > 3”. Currently, our parse does not support parentheses to change logical operation priority.

1. Attribute

Attribute is the basic class in our RDBMS. Every attribute has its attribute name, data type (int, float, str and datetime), attribute type (primary key, not null and null) and attribute values which are stored in a list.

We also implement B+ Tree for each attribute. In the B+ Tree we build, every leaf node contains attribute value as key and corresponding index list as value. The leaf nodes are connected using doubly linked list. Therefore, our B+ Tree supports searching for value and searching for range.

1. Table

Every table contains its table name and several attributes. We implement methods “add tuple”, “update tuple”, “delete tuple”, “add attribute”, “drop attribute”, “select”, “group by” and so on.

1. Database

Every database contains its database name and several tables. The database also includes foreign key information so that once a foreign key’s reference key changes, the foreign key also changes.

We also implement inner join in database. Given 2 tables A and B, if |A|<lg(|B|) or |B|<lg(|A|), we use nested loop. Otherwise, we use merge scan.

We use pickle to save and load database.

1. SimpleSql

We also implement an interactive environment for users. Users can type the SQL statements they want and they can get feedback once the statement been executed.

The database will be saved automatically every time it is changed.

SimpleSql examples:

Create table: CREATE TABLE Students (ROLL\_NO int,NAME varchar NOT NULL,SUBJECT varchar,primary key(roll\_No));

Add attribute: ALTER TABLE students ADD gender varchar

Drop table: drop table students

Insert tuple: INSERT INTO students(roll\_no,name,subject) VALUES (1, 'Seiun', 'CS');

Update tuple: update students set name = 'riven' where name = 'seiun'

Delete tuple: delete from students where name = 'riven'

Create index: create index on students(roll\_no)

Drop index: drop index student(roll\_no)

Load database: load database demo

Select: select \* from t1 where t1.a11 inside (1, 10) or t1.a12 < 15

Join: select \* from t1, t2 join t1, t2 on t1.a11 = t2.a21 where t1.a11 inside (1, 10)

Aggregation:select count(t1.a11) from t1 where t1.a11 > 10

group by: select \* from t1 where t1.a11 < 10 group by a11 having avg(a12) > 7