Database project report

Rongxuan Hu

Yueran Zhang

1. Introduction:

In this database design project, our team is going to use music data from Apple Music. Since music data contains non-traditional type of data (the music content itself), it’s a great example for the database design project.

1. data

In this database design, our main data, the instances in the songs entity set, comes from Apple Music, which can be retrieved from <https://www.apple.com/music/> and the iTunes application in our laptop. Also, we are going to use user, lyricist (the person who writes the lyric), singer and composer as related entity sets to represent our database, these data could be retrieved from Wikipedia by searching the names of these musicians. Because of the limit of our database scale, we do need some assumptions of our data type and their relations, we will illustrate these assumptions later. And also, due to the limit, we will only select some attributes to describe these entities and relationships.

1. Index

In this database, we choose B+ tree for all of our tables. Also, our search key is the same as our primary key.

1. Applications.

We wrote 2 applications, administrator and user client, by Java in this project.

Here is a copy of the readme.txt which explains how to operates and what to expect from the project.

---------------------------------------------------------------------------------

|////////////////Apple Music Database System//////////////////////////////|

|/////////////created by Rongxuan Hu&Yueran Zhang/////////////////|

---------------------------------------------------------------------------------

To compile:

open the file directory then:

~$ javac AdminDriver.java

~$ java -cp /usr/share/java/mysql.jar:. AdminDriver

~$ javac ClientDriver.java

~$ java -cp /usr/share/java/mysql.jar:. ClientDriver

General:

Please follow the user prompt in the menu to operate, be careful with input format,

for certain key values (especially on insert&delete) you will need to put ' mark around

your string-type input(for example if I was trying to delete a person named mark from record,

I would type in: 'mark' instead of: mark )

AdminDriver.java:

In the AdminDriver.java, user can use 3 operations and 1 query:

update,delete,create and query purchase history of any specific user

Before each operation, the system will prompt the user to input the name of the table and then

display the attributes in that table

Please do notice that if a attribute's value does not tolarate null, when you create new entries,

that attribute must be in fields and has a fitted value.

-Update:

update will simply look for a tuple that has the attribute&attribute value we searched for, and

it will just the input value to replace the old one. Do notice that if you are operating on a

foreign key, this will not work. And if you are operating on an attribute thats parent of another

table's foreign key,it will cascade.

-delete:

Becareful to use the primary key, as this method will look for the value&attribute you entered and

delete whichever tuple owns them.

-insert:

Please do follow the format very carefully, or you will not be able to generate tuple.

ClientDriver.java:

simple-to-use, user friendly menu, just follow the prompt, no special spelling here

search by ID will only return one tuple(if found any)

whereas for the rest, it will return a list of tuples, depending on how many there could be.

Last update:2018/12/7