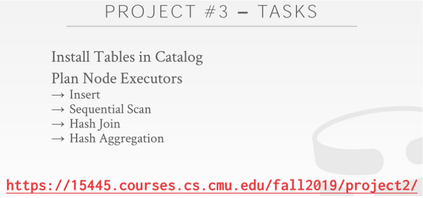
16–05

16–05

01:20:01 – 01:20:01
okay

1.20.01–1.20.05
so you guys are doing it could be single threaded execution，no exchange operators
So，你们可以使用单线程执行来做到这点，不需要使用exchange operator
01:20:06 – 01:20:07
and you don’t have to worry about transactions
你们不需要去关心事务方面的东西
1.20.07–1.20.10
you don’t have to worry about you know doing updates or deletes
你们也不用去担心更新或者删除之类的操作



PROJECT #3 – TASKS

Install Tables in Catalog
Plan Node Executors
→ Insert
→ Sequential Scan
→ Hash Join
→ Hash Aggregation

https://15445.courses.cs.cmu.edu/fall2019/project2/

01:20:11 – 01:20:14
so what do you have to build
So，你们得去构建什么呢？
01:20:14 – 01:20:16
so the first thing we’re asking you to do is build out the catalog
So，我们首先要求你做的事情就是构建出catalog
1.20.16–1.20.19
there’s allows you to install tables into the database
这允许你将表加载到数据库中
01:20:19 – 01:20:24
and then go back and get those tables back from the Catalog using the name or the internal identifier
通过使用name或内部标识符（我们常说的id）来从catalog中拿到这些表
01:20:24 – 01:20:27
And then you’re building executors for insert ,sequential scans, hash join and hash aggregation
然后，你要为插入，循序扫描，hash join以及hash聚合这些操作构建executor
01:20:28 – 01:20:30
for the hash join
对于hash join
1.20.30–1.20.36
you can implement it first using a you know in–memory hash table that will provide you

你可以先通过我们提供给你的内存型hash table来实现它

01:20:36 – 01:20:37

but the ultimate goal is that

但最终目标是

1.20.37-1.20.40

you want to use your linear probe hash table that you built from project two

你想去使用你Project 2所构建出的linear probe hash table

01:20:40 – 01:20:43

Because that allow you to do joins on tables that don’t fit in disk

~~因为这能让你对不放在磁盘中的表进行join~~

~~因为这允许你在不将表放入磁盘的情况下，对这些表进行join~~

因为这能让你在无须将数据放入磁盘的情况下对表进行join操作

01:20:43 – 01:20:48

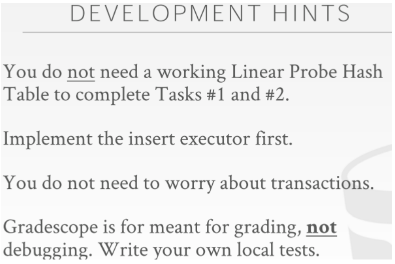
So you can do the first two tasks without having to do that working linear probe hash table

So，你可以在不使用linear probe hash table的情况下，先去完成前两个任务

1.20.48-1.20.50

the last one will require you to have that one working

最后一个任务则需要你去使用你自己Project 2中所构建的linear probe hash table



DEVELOPMENT HINTS

You do not need a working Linear Probe Hash Table to complete Tasks #1 and #2.

Implement the insert executor first.

You do not need to worry about transactions.

Gradescope is for meant for grading, **not** debugging. Write your own local tests.

01:20:51 – 01:20:55

so even up the catalog， implement insert executor first

So，除了catalog，你们还得先实现insert executor

1.20.55-1.20.58

because obviously you can’t do sequential scans unless you have data in your database

因为很明显，除非你的数据库中有数据，不然你没法进行循序扫描

01:20:59 – 01:21:01

you don’t need to worry about any transactions.

你也不需要担心任何事务方面的事情

1.21.01-1.21.04

and then when posted this on Piazza and I’ll just emphasize this again

当我把这个贴到Piazza上的时候，我会再强调一遍

1.21.04-1.21.07

GradeScope is not meant to be for debugging

GradeScope并不是帮你们用来对代码进行debug的

01:21:07 – 01:21:09

right if you submit it

如果你将你的代码提交到GradeScope

1.21.09-1.21.11

and it takes a half an hour or for starts running

它需要花半个小时来运行这段程序

1.21.11-1.21.12

we can’t fix that, right

我们没法修复这个

01:21:12 – 01:21:13

that’s up to GradeScope

这取决于GradeScope

1.21.13–1.21.19

there’s a queue of other students with100 students in the class, you not gonna run right away

它会有一个队列，比如说，我们课上总共有100名学生，它会按个对你们的作业进行测试，你没法提交完就立马进行测试

01:21:19 – 01:21:23

so we provide some basic tests as a framework to figure out how to write more tests

So，我们提供了一些基本测试，将它们作为一个框架来帮你们弄清楚如何写更多的测试用例

01:21:23 – 01:21:28

but you should doing as much as all the development want to figure out what your problem is to be done locally

但你应当在本地尽可能多地进行测试，以此来弄清楚你的问题在哪

01:21:29 – 01:21:32

and then if you find this thing timing out, because it’s running too slow

如果你出现运行超时的情况，这是因为它的运行速度太慢所导致的

01:21:32 – 01:21:34

you should figure out why your system is running slow locally

你应该去弄清楚你的系统本地运行速度为什么这么慢

1.21.34–1.21.38

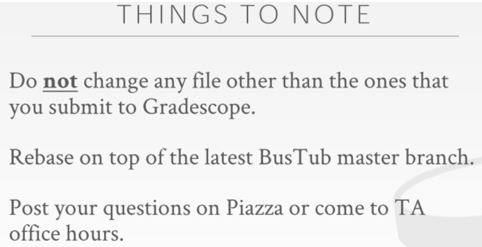
don’t make a bunch of you know minor changes, and keep submitting them to great scope ,and try to measure how long GradeScope takes

不要每改动一点点地方，你就把代码提交到GradeScope上，并试着测出GradeScope测试一次所要花的时间

01:21:38 – 01:21:40

you can figure these things out locally

你可以在本地对你的系统进行测试



THINGS TO NOTE

Do **not** change any file other than the ones that you submit to Gradescope.

Rebase on top of the latest BusTub master branch.

Post your questions on Piazza or come to TA office hours.

01:21:41 – 01:21:42

so we’ve already covered this

So，我们已经介绍过这个了

1.21.42–1.21.46

don’t change any file and then once we give you

除了你提交到Gradescope上的文件以外，不要修改任何文件

01:21:46 – 01:21:49

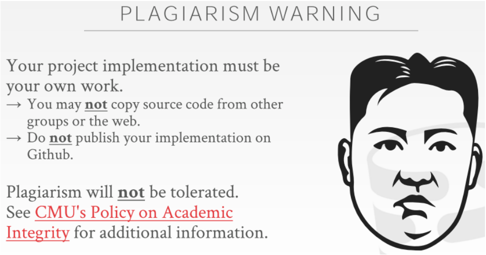
you want to rebase over the BusTub master there’s instruction Piazza how to do this

然后，你们要对最新的BusTub的master分支进行rebase操作，Piazza上有写该如何进行操作

01:21:49 – 01:21:51

And then come to office hours if you have questions

然后，如果你们有问题，那就在办公时间来找我



PLAGIARISM WARNING

Your project implementation must be your own work.

→ You may **not** copy source code from other groups or the web.

→ Do **not** publish your implementation on Github.

Plagiarism will **not** be tolerated. See CMU's Policy on Academic Integrity for additional information.

01:21:51 – 01:21:52

I always have to say this

我总是得反复提下这个

1.21.51–1.21.53

don't plagiarize

不要抄袭

1.21.53–1.21.55

we will destroy you right

不然我们会举报你抄袭

1.21.55–1.21.58

because what'll happen is if we do plagiarize new copy from other people

因为如果你抄了别人的作业

01:21:58 –01:21.59

University comes back to me say

学校就会来找我，并问我

1.21.59–1.22.01

hey did you tell your students not to plagiarize

你有没有告诉你的学生不要抄袭

1.22.01–1.22.02

and I show them the video

我就会向他们展示这段录像

1.22.02–1.22.05

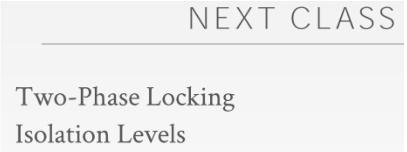
and I guess here's me telling you not to plagiarize ,don't plagiarize

我会在这段录像中记录我跟你们说不要抄袭的对话

01:22:05 – 01:22:07

and that's evidence against you're screwed ,okay

我会保留这些证据，防止翻车



NEXT CLASS

Two-Phase Locking

Isolation Levels

01:22:07 – 01:22:15

all right next class again I will be gone，yes no we go on,it's how we good

emmm，下节课我不在

01:22:15 – 01:22:20

all right we'll cover two-phase locking and that'll build upon what we talked about today okay

我们会介绍两阶段锁，这个是建立在我们今天谈论的内容之上的

01:22:21 – 01:22:23

Bye guys, enjoy your Wednesday

好好享受你们的周三吧