**Lab 9: Indexes**

**Tailang Cao u1480633**

**Part 1 - Selecting Indexes**

A screenshot of a computer

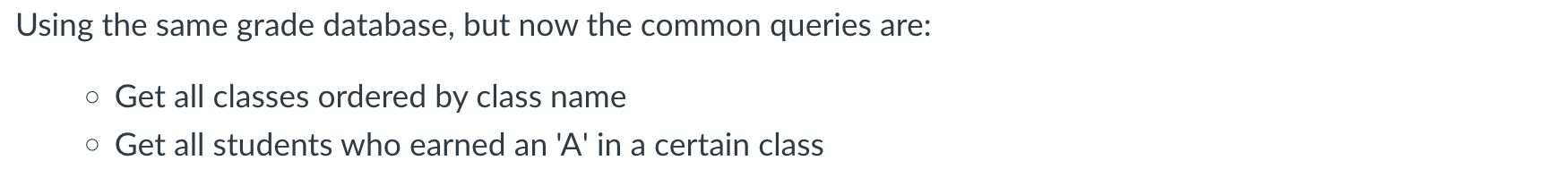
Description automatically generated

Adding composite indexes on ( ‘Start Date , ‘End Date’) would make the process more efficient. When searching for employees started after the certain date, the start date part of the composite index would be helpful. When doing the second query, both part of the composite would be used. Creating single composite index would be more efficient then creating two indexes on start date and end date separately.

A screenshot of a computer

Description automatically generated

Adding index on Grade.



Adding index on Grade and className.

A screenshot of a chat

Description automatically generated

1. Adding index on Player.Elo
2. Adding index on Games.WhitePlayer

A screenshot of a computer

Description automatically generated

No additional indexes are required.

A screenshot of a phone

Description automatically generated

1. Adding index on CheckOut.CardNum
2. No additional indexes are required.

A screenshot of a computer

Description automatically generated

Adding Index on Inventory.ISBN

**Part 2 - B+ Tree Index Structures**

A close-up of a text

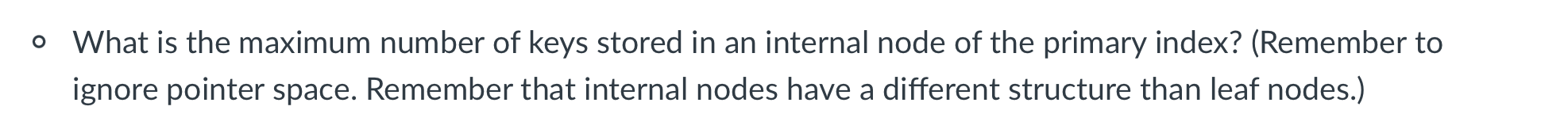
Description automatically generated

Student ID is an int which takes 4 bytes, className is a varchar(10) which takes 10 bytes, grade is a char(1) which takes 1 byte.

Therefore each row in the chart takes 15 bytes.

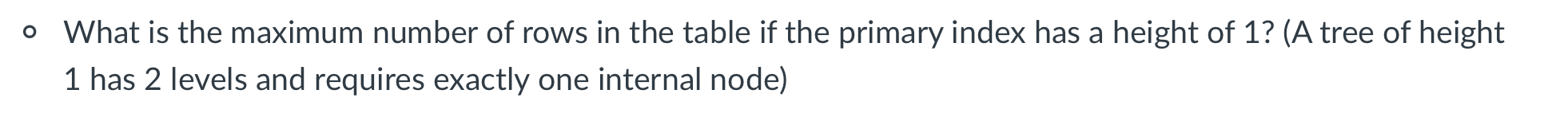
The leaf node of the primary index has 4096 bytes.

4096 / 15 = 273 rows.



The key is student ID and className (size: 4 bytes + 10 bytes = 14 bytes).

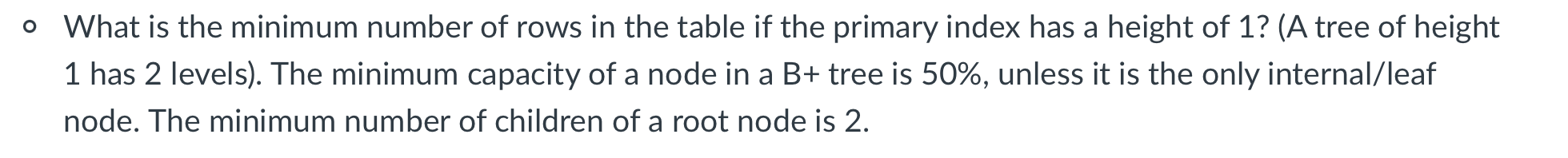
4096 / 14 = 292.



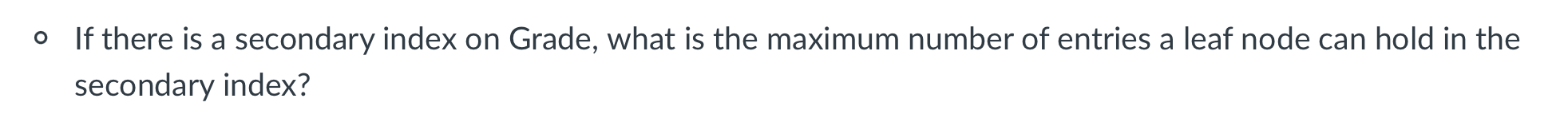
The B+ tree has an order of 292 + 1 = 293.

Each leaf node has 273 rows in it.

Therefore maximum number of rows is 293 \* 273 = 79,989 rows.



One node can have up to 273 rows, more than 273 rows will cause a split.  
Therefore the minimum number of rows for a height of 1 is 273 + 1 = 274 rows.



The grade is a char(1) which takes 1 byte. The primary key takes 14 bytes.

4096 / 15 = 273.

There for maximum number of entries is 273 rows.

A screenshot of a computer

Description automatically generated

1. 4096 / 128 = 32 rows.

As leaf nodes need to be half full, each node can hold minimum 32 / 2 = 16 rows.

48 / 16 = 3.

Therefore maximum number of leaf nodes is 3

1. In this case each node should hold 32 rows.

Ceil (48 / 32) = 2

Therefore minimum number of leaf node is 2.