**REVIEW - 2.6.1**

Kunal - Blue

Sandeep - Red

Onkar - brown

Siddharth - Black

Jayadeep - Dark purple 3

Dev - Green

Pranav - Dark yellow 2

Varun - Magenta

Please add the text review and questions review at the appropriate place in each segment.

**Segment 1 - Module Introduction**

<https://learn.upgrad.com/v/course/78/session/11881/segment/58679>

**Text Review:**

~~Please remove the word “the” before Mapreduce in the first paragraph.~~

~~Add the words “a in front of NoSQL in this line~~

* ~~In this module, you will learn about the HBase - NoSQL database component of Hadoop Ecosystem.~~

~~This line needs to be rewritten. I am not sure what is being said here~~

* ~~Some of the limitations of Hadoop like only confined to batch processing of data and unable to perform random lookups on data and unable to update the existing data laid the foundation for HBase.~~

Have removed Gowtham’s name from HBase part and also changed the graded questions part.

### **Segment 2 - Session Introduction**

<https://learn.upgrad.com/v/course/78/session/11881/segment/58695>

**Text Review:**

Siddharth - Let’s set the expectations clearly here that we cannot cover industry applications of HBase as it is used along with a few other tools which you will learn in further courses.

### **Segment 3 - Introduction to NoSQL Database**

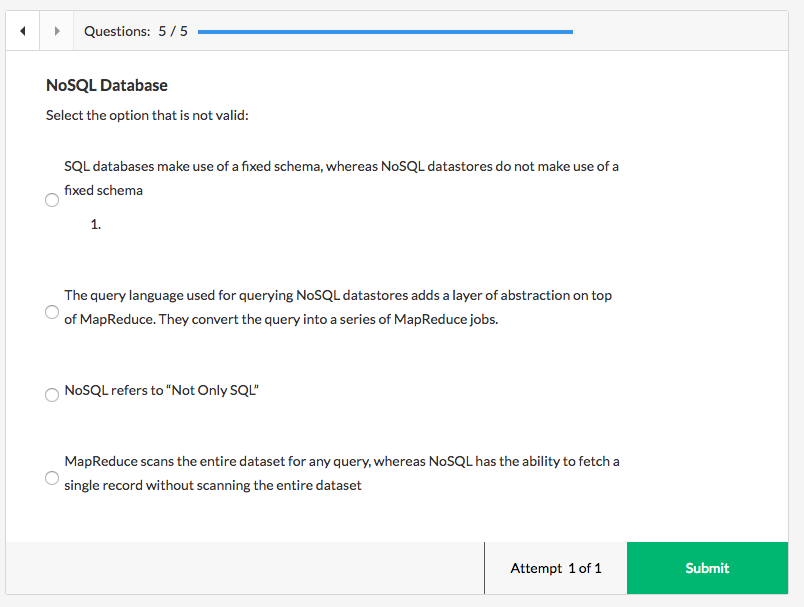
<https://learn.upgrad.com/v/course/78/session/11881/segment/58696>

**Text Review:**

**Questions Review:**

**Q1 - Check if the correct option has been marked**

~~The 5th question has an extra 1 in the first option. It seems out of place.~~



### **Segment 4 - CAP Theorem**

<https://learn.upgrad.com/v/course/78/session/11881/segment/58698>

**Text Review:**

Is the following copy-pasted from somewhere or is it an original piece of writing? If it is copy-pasted then we need to re-write this to avoid plagiarism

* The CAP theorem relates to the distributed systems that store data. This theorem was first published as the CAP principle in 1999 and later presented as a conjecture by Eric Brewer at the 2000 symposium on Principles of Distributed Computing (PODC). In 2002, Seth Gilbert and Nancy Lynch of MIT published a formal proof of Brewer’s conjecture. The CAP theorem states that it is impossible for a distributed data store to simultaneously provide more than two out of the three guarantees - Consistency, Availability and Partition tolerance. Some of the known systems and the guarantees provided by them are mentioned below:

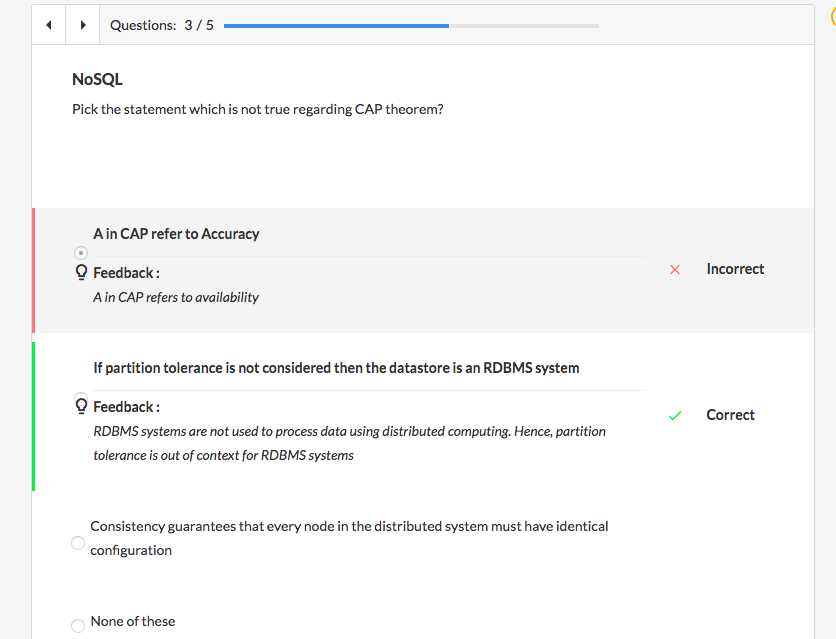
This is from the same link which we have given as additonal reading. Let’s remove the text and just present the additional reading.

Can we add something related to C, A and P to refresh the part which was covered in the video?

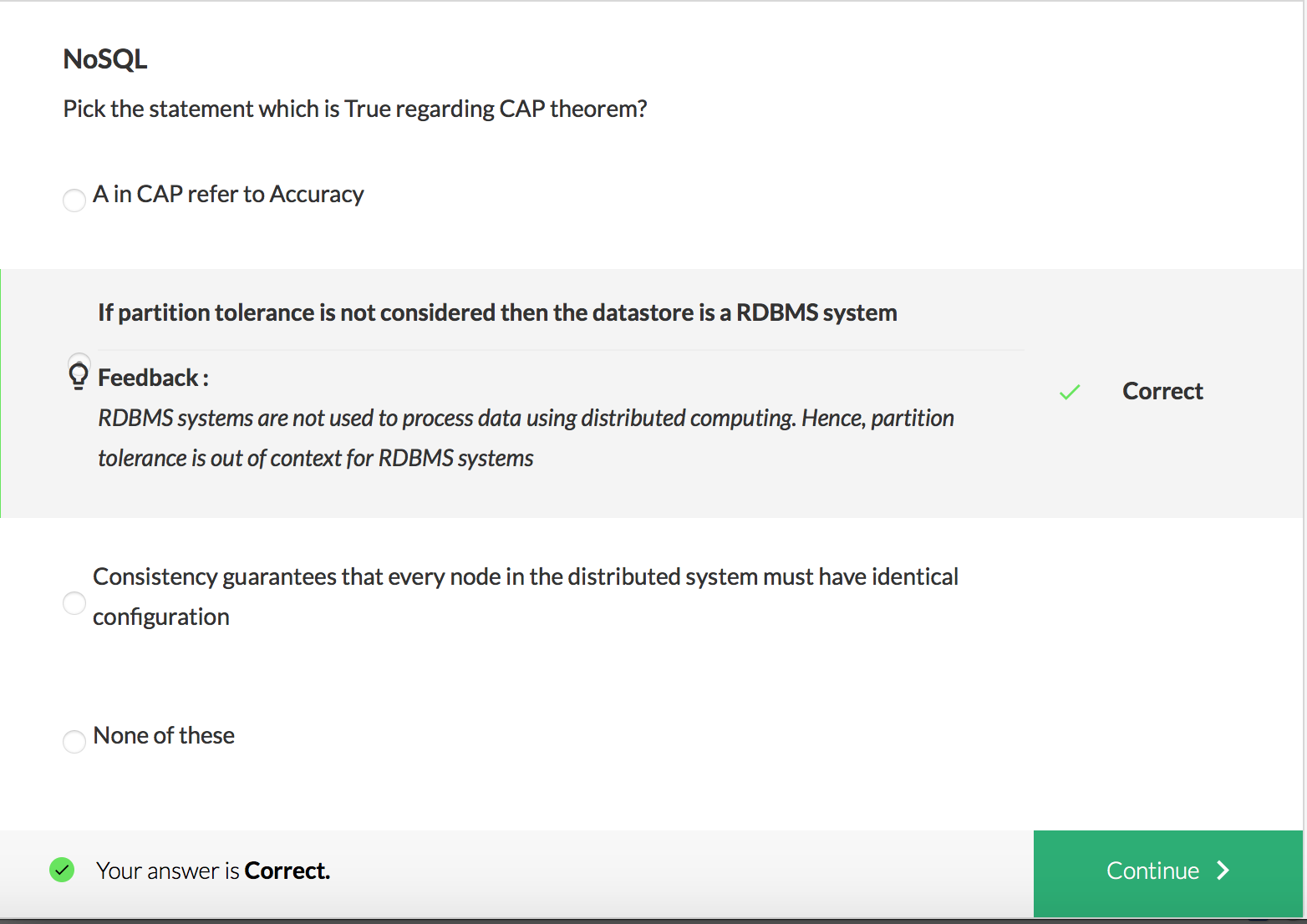
Besides the RDBMS, NoSQL systems also include CA, AP, CP system etc. so that students can easily relate to these later.

**Questions Review:**

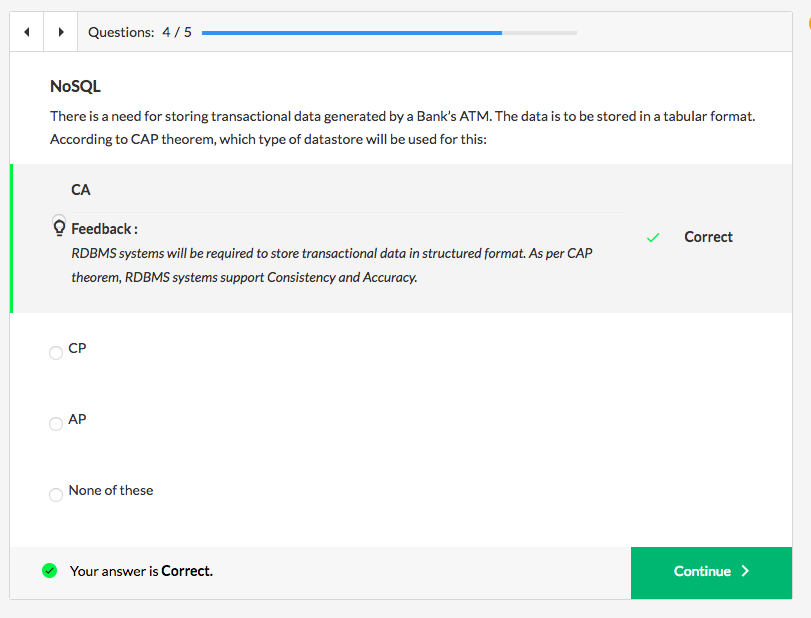
Question 3 seems very odd. It says which is NOT TRUE about CAP theorem. And then option 1 says A is for Accuracy. And this is NOT TRUE about CAP theorem. The feedback is correct that A is for Availability. Please look at this question again.



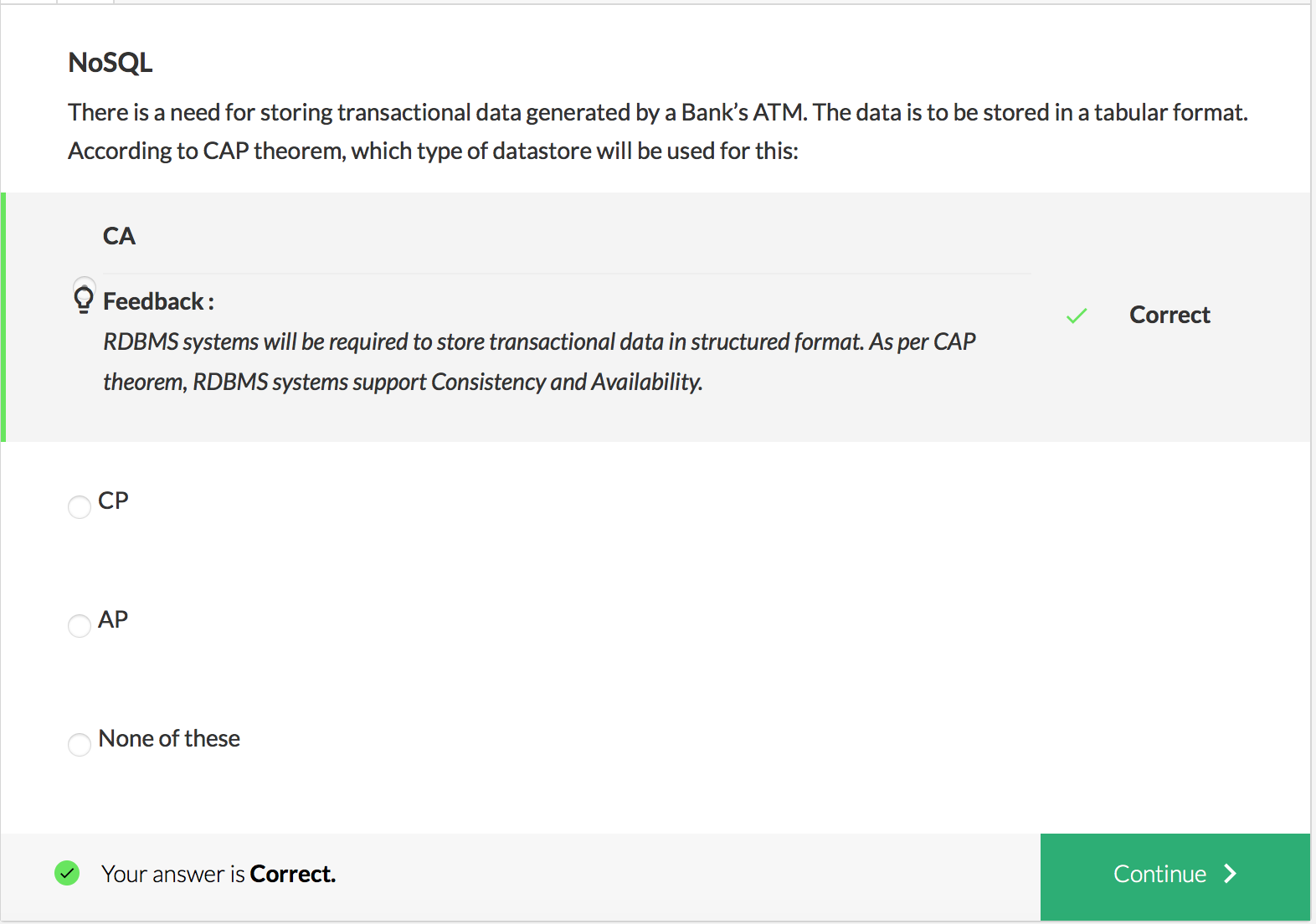
Modified the question. Please check:



Question 4 - The feedback in CA seems off. A stands for Availability not Accuracy.



Modified feedback:



### **Segment 5 - Introduction to HBase**

<https://learn.upgrad.com/v/course/78/session/11881/segment/58699>

At 1:16, why is the title of the plate “CAP theorem”?

**Text Review:**

~~Subject verb agreement. Use “distinguish”~~

* ~~Some of the prominent features of HBase that~~ **~~distinguishes~~** ~~it from Hadoop~~

I just saw on multiple sites that the key in HBase consists of the row key, column family, column name and timestamp, contrary to what Prof taught in the lecture. We should get this corrected in text. The following is from IBM’s website:

Data is accessed by rowkey, column family, column qualifier, and timestamp, which together make the key part of the key/value pair. All key/values for a given column family are stored together in the same set of files. Keys are compared on a binary level, byte by byte, from left to right. Row keys are always unique. Row keys can be any arbitrary array of bytes.

**Liked question 2 :)**

**Questions Review:**

### **Segment 6 - Data Model of HBase**

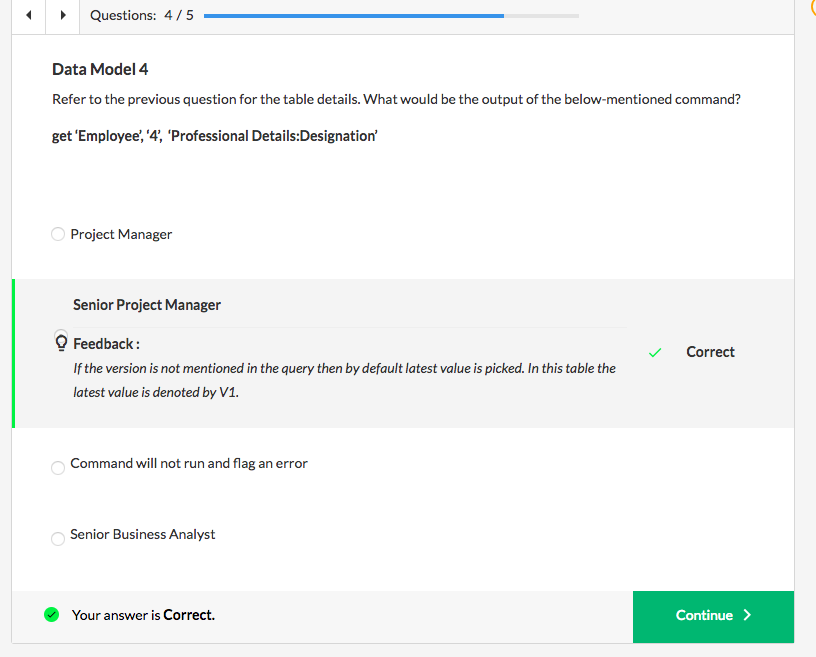
<https://learn.upgrad.com/v/course/78/session/11881/segment/58700>

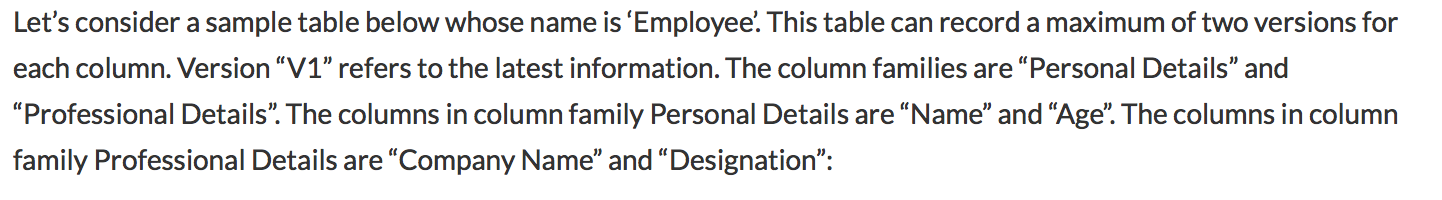
**Text Review:**

**Questions Review:**

~~In question 4, please add the screenshot of the Database schema. It is difficult to go back and forth between question 4 and 3.~~

Also, is V1 the latest or is V2 the latest? It seems to me that V2 should be more current than V1. But I am not the expert at this.





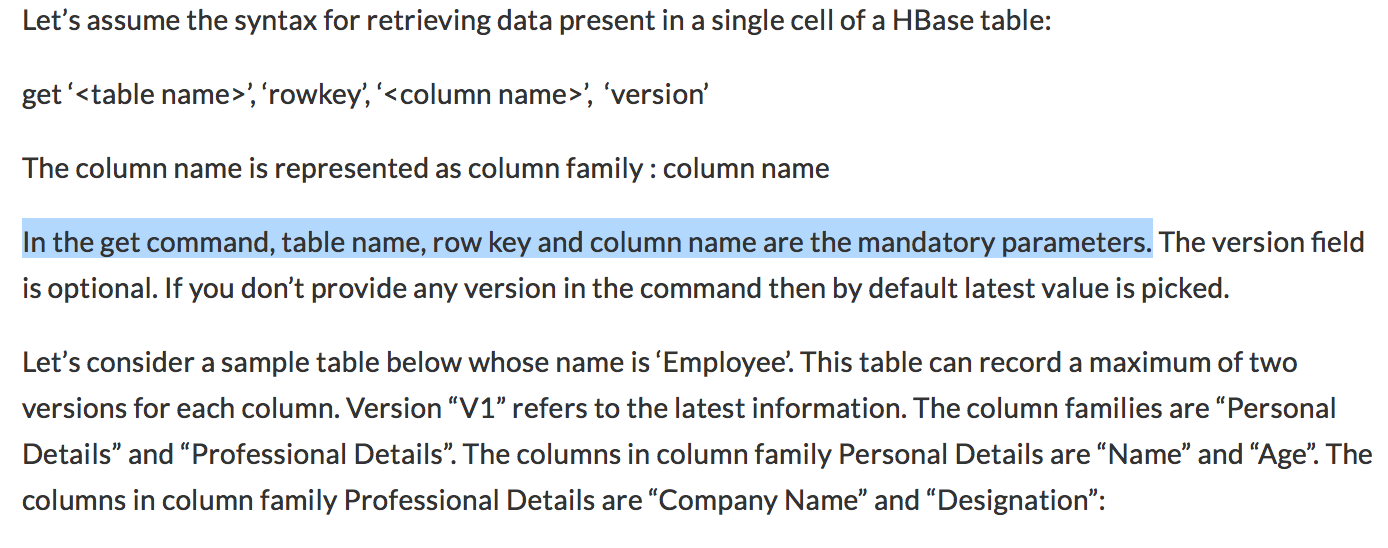
There are two problems with question 5. First please add the screenshot of the database schema so we don’t have to go to Question 3 again. Also, then update the text which says “please go back to question 3”

Secondly, the feedback is incorrectly written. Please DOUBLE CHECK it.

It says

* *This command will not run as the mandatory field i.e. the column family followed by the column family followed by the column name is missing.*

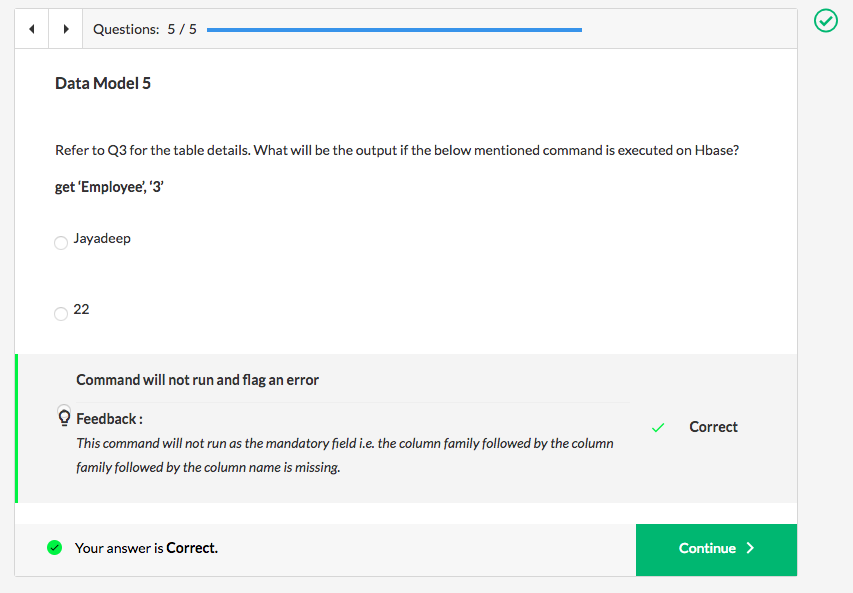
Till this session students are not introduced to the “get” command. They will be introduced to get command in next session. So just to explain how the data model in HBase works i introduced my version of get command with some modifications, whose description i have clearly mentioned in the problem statement:



So based on this description the answer to that question is correct. The will learn the correct command in next segment

Should it instead be

* Column family followed by column name is missing
* If so, in the feedback add what the correct command looks like



### 

### **Segment 7 - Perform Operations in HBase using Shell Commands**

<https://learn.upgrad.com/v/course/78/session/11881/segment/58701>

**Text Review:**

**IMPORTANT** - The two documents mentioned in the platform text are missing. We should not launch without them.

* basic CRUD operations in HBase using the shell commands on a Cloudera VM
* HBase basic operations using shell commands on AWS EC2

**IMPORTANT -** Because Prof. Hota does not actually write the commands, it is hard to follow the commands and their output. We need Sandeep to create a screengrab video where he shows how to type in these commands and show their output. This can be done after the course launches.

* To add to this at 2:49 Prof. Hota describes some syntax with r1, t1 and c1 which is not clear. When he says we make use of the put command, we should cut to the screengrab of Sandeep where he actually shows how this is done.

Is column\_value here supposed to be column\_family\_name?

* Syntax: **hbase> put ‘<table\_name>’, ‘<row\_key>’, ‘<column\_value>’, ‘<value>’**

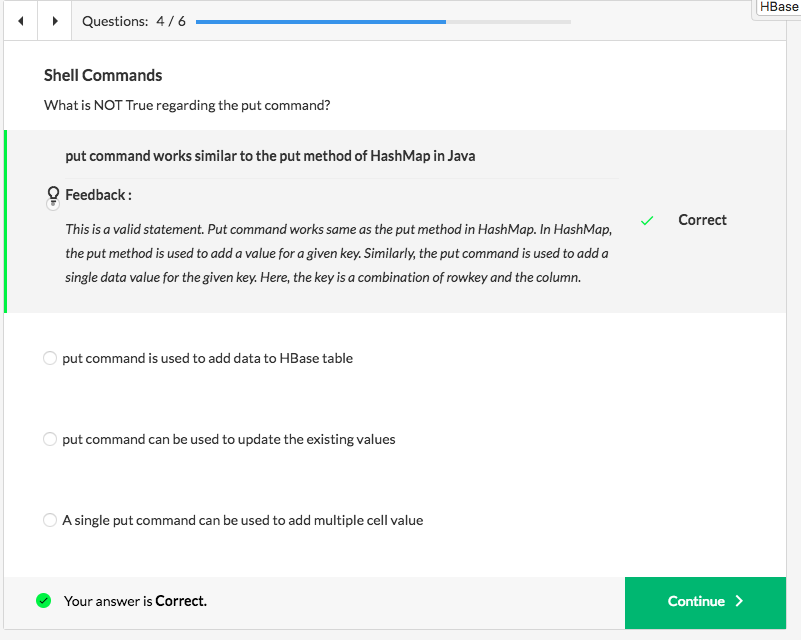
**Questions Review:**

Question 2 of 6 - The image here needs to be clickable so we can click on it and make it bigger.

Question 3 of 6 - Add the image here also.

Why does the command run correctly?

Question 4 does not make sense. We are asking students to pick something that is NOT true with respect to the put command. And then when you select the first option, the feedback says that the following statement is valid - put command works the same ways as put in HashMap. How can this be VALID and NOT TRUE at the same time?



### **Segment 8 - Perform Operations in HBase using Java API**

<https://learn.upgrad.com/v/course/78/session/11881/segment/58702>

Apart from what Kunal suggested in the previous segment, I feel the student will find this portion difficult since Prof. Hota does not explain the Java code at all. It would be nice if we could squeeze a video here explaining the basic program.

**Text Review:**

**IMPORTANT** - The two documents mentioned in the platform text are missing. We should not launch without them.

**IMPORTANT** - The video linked on this page is INCORRECT. It is a repeat of the video from Segment 9 and DOES NOT teach how to perform CRUD operations using JAVA API.

**IMPORTANT** - We need to add platform text below the correct video that explains how CRUD operations are done using Java API

**Questions Review:**

**IMPORTANT** - We need to add some questions at the end of this segment

Q1) What is the functionality of objects of HTable class?

1. Used to insert data
2. **Used to communicate with a single table**
3. Reads the configurations defined in the two configuration files hbase-site.xml and hbase-default.xml
4. None of these

* Option a is incorrect. Objects of Put class are used to insert data into an HBase table. Please go through the session again to find the functionality of HTable class.
* **Option b is correct. Objects of HTable class are used to communicate with a single table. All get and put operations are done using the objects of HTable class.**
* Option c is incorrect. Objects of **HBaseConfiguration** class reads the configurations defined in the two configurations files hbase-site.xml and hbase-default.xml. Please go through the session again to find the functionality of HTable class.
* Option d is incorrect. One of the options is correct. Please go through the session again to find out the correct answer.

Q2) Before executing a Put operation why the data elements are converted to bytes array using the Bytes.toBytes() method

1. Converting to bytes helps in faster insertion of data into the table
2. Helps in serialisation
3. **There are no fancy data types such as String , INT , or Long in HBase; it's all byte array**
4. None of these

* Option a is incorrect. This is not the valid reasoning for converting the data elements to bytes before inserting the data into the table
* Option b is incorrect. This method does not serialize the data. Hence, this option is not correct
* **Option c is correct. HBase is a kind of byte-in and byte-out database. Hence, all data elements before insertion are converted into a byte array**
* Option d is incorrect. One of the options is correct. One of the remaining option is correct. Please check again.

Q3) Which class is used to collect and display the results generated after a scan operation?

1. **ResultScanner**
2. Result
3. Results
4. ScanResults

* **Option a is correct. In Java, objects of ResultScanner class hold the output generated after the Scan operation.**
* Option b is incorrect. In Java, objects of Result class hold the output generated after the Get operation
* Option c is incorrect. In HBase context, Results class is an invalid class
* Option d is incorrect. In HBase context, ScanResults class is an invalid class

Q4) Judge if the given statement is True or False

The output of a get operation is of type byte array.

Ans) True.  **HBase is a kind of byte-in and byte-out database. So all data are stored as byte arrays in HBase. In java, the data fetched from HBase is of type byte[] by default. Data can be explicitly type casted to String using Bytes.toString() method**

### **Segment 9 - Additional Shell Commands in HBase**

<https://learn.upgrad.com/v/course/78/session/11881/segment/58703>

**Text Review:**

**Questions Review:**

### **Segment 10 - Summary**

<https://learn.upgrad.com/v/course/78/session/11881/segment/58704>

**Text Review:**

**IMPORTANT**

**Where is the Optional Student Project? It is missing.**