SCHWERPUNKTBEITRAG

Database (Lecture) Streams on the Cloud

Experience Report on Teaching an Undergrad Database Lecture During a Pandemic

Jens Dittrich[1] · Marcel Maltry[1]

© The Author(s) 2021

Abstract
This is an experience report on teaching the undergrad lecture *Big Data Engineering* at Saarland University in summer term 2020 online. We describe our teaching philosophy, the tools used, what worked and what did not work. As we received extremely positive feedback from the students, we will continue to use the same teaching model for other lectures in the future.

Keywords Online Teaching · Streaming Lectures · YouTube

1 Background

In February 2020, many German universities decided to move to virtual teaching due to the unfolding Covid-19 pandemic and the public lockdown. As the summer term approached, lecturers had to decide relatively quickly how to *give all lectures online*. This included our computer science department and our research group (Big Data Analytics Group) at Saarland University. We already had quite some experience with teaching online. In 2013, we started creating and using screencast videos for our lectures. From the beginning, we made our videos publicly available on YouTube. Our channel[1] currently has 11,000 subscribers and is one of the most successful database channels worldwide. We already reported on that endeavour and in particular our inverted classroom philosophy in a previous report [1] in this journal. However, since then, and given the experience with numerous inverted lectures, we decided to revisit all design decisions taken in the past.

In this experience report, we discuss how we evolved our lecture *Big Data Engineering* to an online class. The lecture is aimed at undergrad computer science students and students from computer science-related programs. We describe what we did and why we did it. For each decision, we briefly discuss the alternatives and their pros and cons. We do not claim that our model is the single right way for teaching online because there is no such thing neither for teaching online nor offline. There are a zillion different ways. The challenge is to find the teaching style that fits you best. In summer 2020, we found a teaching style that fits well for us – and as we learned from the students' evaluations – it also fits for our students. With this work we would like to inspire others seeking to find the right bits and pieces for their own teaching.

This paper is structured as follows. We explain our teaching philosophy in Sect. 2. Our teaching material is discussed in Sect. 3. The hardware and software used are presented in Sect. 4. In Sect. 5, we proceed to discuss tutorials, office hours, and assignments. We summarize advantages and issues with our teaching model in Sect. 6. Finally, we give an outlook to future plans in Sect. 7.

2 Teaching Philosophy

A major and common pitfall in online teaching is to *first* start discussing technical tools, and specific software, and *only then* consider the pedagogy and teaching philosophy behind it as a second thought. Like that, technology and software will define the pedagogy.

---

[1] https://youtube.com/jensdit.

✉ Jens Dittrich
   jens.dittrich@bigdata.uni-saarland.de

[1] Big Data Analytics Group, Universitat des Saarlandes, Saarbrücken, Germany

So let's first think about the teaching philosophy and pedagogy. There are two extremes:

1. **Flipped (aka blended or inverted) teaching.** In flipped teaching, students receive material for self-study ahead of time. The actual class is then used as a lab to work on exercises with the professor and Ph.D. students[2].
2. **Live Teaching.** The other option is to run the lecture live over the Internet. This may be done as an online version of the physical lecture where content is 'streamed' from the lecturer at the speaker's desk to the people in the lecture hall.

Of course, all kinds of hybrid solutions between these two extremes are possible. However, a major concern with all these approaches is the degree of interaction with and among the students. A considerable strength of flipped teaching is the lab which is heavily interactive and perceived to be very useful by many students. However, during the self-paced self-study phase, questions to the lecturer are not possible as the students are basically alone with the material. If they get stuck, issues might not be resolved quickly.

In contrast, a major strength of live teaching is the potential interactiveness. The lecturer can spontaneously ask questions, form short working groups, resolve urgent issues immediately, react to witty comments, and then depart in completely different directions, and so on. Like that, live teaching really becomes a live event that goes beyond a unidirectional stream of content from a lecturer to an audience.

So what is the *right* solution here? Again, there is no single right solution here. We debated back and forth and eventually decided to go with live teaching. Besides the potential of interactive elements, one reason for this decision was also the perceived high effort for creating video material for an entire lecture in a short period of time.

3 Teaching Material

Until 2013, our lecture relied on material and examples from [2]. In 2013, we created our own material based around a hypothetical photo agency. In 2019, we again created new material from scratch based on the following idea. We did not want to primarily organize the lecture along different topics anymore (first entity-relationship model, then relational model, then relational algebra, etc.) as typically done, as we felt that this did not necessarily motivate students to fully appreciate the power of database technology.

Instead, we decided to motivate certain topics by different applications. Every two weeks, we would pick a differ-

---

[2] See our tutorial video https://youtu.be/RRqoQAaeGCc (in German) on how to organize a flipped classroom successfully.

Concept of this course: Learning by Application

Planned structure for every two weeks of lectures:
1. Concrete application: XY
2. What are the underlying the data management and analysis issues?
3. Basics to be able to solve these problems
   (a) slides
   (b) Jupyter/Python/SQL hands-on
4. Transfer of the basics to the concrete application

Planned structure of each exercise sheet:
1. 2 exercises related to basics: slides
2. 1 exercise related to basics: Jupyter/Python/SQL Hands-on
3. 1 exercise related to the transfer of the basics to the application

11/32

Fig. 1 Slide explaining the teaching philosophy of the course to students (translated)

ent application. This means, we would first show an application or problem and then showed how certain techniques from the database world would solve exactly that problem. Then, we would spent quite some time to make the transfer from the technology itself to how does that technology help in this application? This principle structure is summarized in Fig. 1. Table 1 shows the applications discussed and their mapping to topics.

As Saarland University decided to shorten summer term 2020 by starting only in May rather than in April, we also had to shorten the material. In addition, all courses had to be designed to allow all students to start in May only. If we provided material before May, we had to make sure to

Table 1 Course agenda and their learning objectives. In 2019, we also discussed data journalism to explain graph databases and security issues like SQL injection

| Topic	Learning objectives
Python (Part 1, videos and/or 5.5.)	Basics, functions, functional programming
IMDb (Part 1, 7.5.)	Data modeling, relational model
Python (Part 2, videos and/or 12.5.)	Object-orientation, unit tests and automatic testing
IMDb (Part 2, 14.5.)	Relationale algebra
NSA (Part 1, 28.5.)	Introduction to SQL
NSA (Part 2, 4.6.)	Analytical SQL, big data arithmetics, big data vs privacy, counter-measures
Query optimization (Part 1, 18.6.)	Automatic query optimization, physical operators, heuristic optimization
Query optimization (Part 2, 25.6.)	Cost-based optimization, join order, plan variants, pipelining, physical optimizations
Trade, banks, ticket system (Part 1, 2.7.)	Database management system (DBMS), transactions, serializability theory
Trade, banks, ticket system (Part 2, 9.7.)	Two-phase locking (2PL), isolation-levels
Summary (16.7.)	
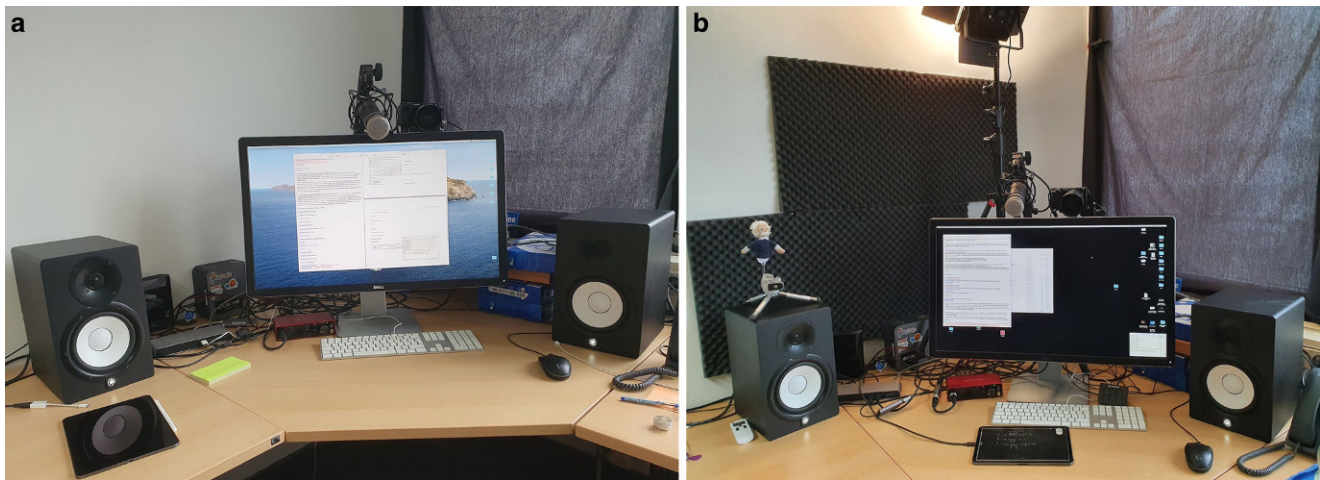
Fig. 2 The evolution of the hardware setup, **a** summer term 2020, **b** winter term 2020/21

go through it again in May. Due to these additional constraints, we decided to offer material for students who did not know Python yet as we planned to use Jupyter notebooks to explain certain concepts in the actual lecture. That Python introductory material was then again repeated in two lectures in May.

Moreover, throughout the lecture we also recommended old videos from 2013/14 to students in case they wanted alternative explanations. For all material that we created, we paid attention that our notation was consistent with [2] in order to enable students to easily lookup yet another explanation in that book.

Also note that we left out considerable material that we felt like is not up to the reality of modern databases anymore, e.g. normal forms whose importance can be debated in the light of modern non-scalar SQL-types[3].

All videos, including the older ones from 2013/2014 are publicly available on our YouTube-channel[4]. The pdfs of our slides are available through our website[5]. If you are a lecturer and want to have access to the sources, send us an email. We also created Jupyter notebooks[6] which were presented during the lectures and used for the exercises.

4 Technology

In this section, we describe the hardware and software we used for our lecture.

4.1 Hardware

In terms of hardware, we experimented a lot until converging on the following setup. We used an existing 2016 Macbook Pro, a 32 inch monitor, a thunderbolt dock (Elgato Thunderbolt 3 Pro Dock), a dynamic microphone with internal pop-filter (Rode Procaster), a mike preamp (TritonAudio FetHead), a sound interface (Focusrite Scarlett Solo 3rd Gen), and two speakers (Yamaha HS7 Standard). That's it[7]! An existing iPad Pro was not used in the lecture. The office used for streaming was a standard office room without any extra soundproofing (one exception: see Sect. 4.3).

With this setup, see Fig. 2a, the sound quality of the audio is awesome and makes a huge difference over any setup we used before. In particular, previous issues with background noise are gone.

In winter term 20/21, we slightly improved this setup, see Fig. 2b. We now use the video camera for all lectures. In addition, we occasionally use an iPad Pro as a blackboard. We also added a stream deck mini to allow for easy switching between OBS-scenes. In addition, we acquired professional adjustable LED panel lights (Neewer) and use more soundproofing.

4.2 Internet Connection

An important and surprising issue we ran into early was how to connect to the Internet. Initially we had quite some issues when connecting the computer via Wi-Fi. After ex-

---

[3] See this talk by Markus Winand https://youtu.be/swR33jIhW8Q for a great explanation of how modern SQL has evolved from SQL-92 and why many people unfortunately only use very old standards of SQL.

[4] https://youtube.com/jensdit.

[5] https://datenbankenlernen.de.

[6] https://github.com/BigDataAnalyticsGroup/bigdataengineering.

[7] Occasionally, we would display a video of the lecture. We then used an existing APS-C camera (Sony alpha 6000 with 1.8/24 Zeiss lens) powered by a fake battery adapter (SONY AC-PW20), which was connected via an HDMI video card (Pengo HDMI USB-C 3.0 video capture-card).

perimenting with several setups, it became clear that Wi-Fi is simply not stable enough for live streaming purposes. This is not so much a problem of bandwidth but of channel conflicts and electromagnetic interferences from other devices, as well as latency issues. Though we carefully debugged the Wi-Fi connection, eventually we concluded that a wired ethernet connection should be preferred over Wi-Fi whenever possible. The same holds for audio and video conferences.

As LAN was not available in some home offices, we used a LAN via power option (AVM FritzPowerline 510E Set). Though powerline may lower the available bandwidth, it is much more stable than Wi-Fi. In general, in terms of Internet bandwidth note that even a relatively weak but stable uplink of 2Mb/sec may be enough to stream a lecture that only shows slides in the video stream. If you want to stream camera input, you need a higher bandwidth though.

4.3 Zoom, Teams, or YouTube Streaming?

Initially we planned to use **Zoom** for the actual lecture and additionally stream the Zoom lecture to YouTube. This can directly be done from any Zoom meeting. However, shortly before the semester started we decided against using Zoom for several reasons: (1) ongoing discussions on privacy and data protection issues with Zoom, (2) relatively poor audio/video quality due to lossy compression, (3) the YouTube stream could not be made publicly available if at any time clear text names or webcam videos of students can be seen in the stream.

Similar concerns are true for **MS Teams**. After some brief investigation it turned out that the privacy and security issues with MS Teams were at least comparable to Zoom or even worse, e.g. SSO with cleartext university password on microsoft.com website. In addition, we experienced that the video and audio quality of MS Teams is simply not acceptable. We observed super-heavy compression artifacts and dichotomy of picture and sound. In addition, we found the UI of MS Teams to be extremely confusing.

For the lecture, we decided to use a solution that fixes all of the above issues: **YouTube**'s livestream feature. In order to send a stream from your computer you have to install a streaming software locally. There aren't many options so we decided to use **Open Broadcaster Software (OBS)**[8], see Fig. 3. Commercial options are also around, e.g. Wirecast. OBS worked well for our purposes. The only problem we ran into was that OBS is not really optimized for macOS. This results in a relatively high CPU load which in turn leads to notable fan noise. We had to fix this by positioning the Macbook as far away from the microphone as possible
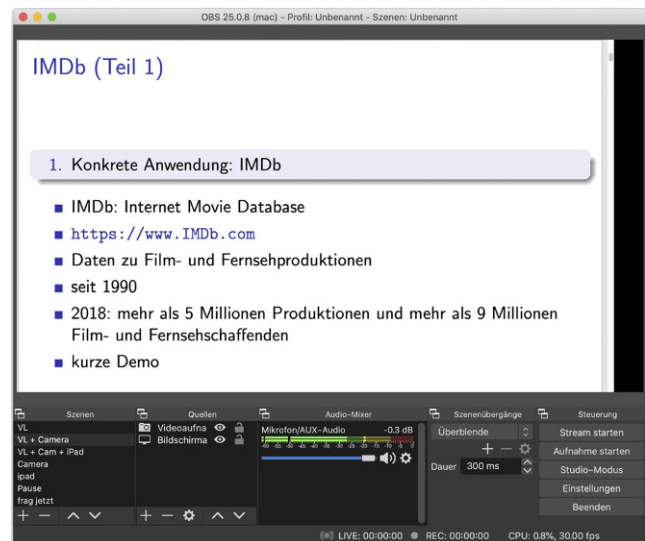

Fig. 3 OBS: Open Broadcaster Software was used to stream directly to YouTube

and improvising some soundproofing between the Macbook and the microphone (a chair covered by a thick blanket).

By default, the head of the YouTube stream is cached to be able to cope with network failures. We observed a **latency** of about 10 s. We configured this to a shorter value to allow for more interaction with students (see Sect. 4.4)[9]. We also disabled the **livestream chat** to avoid spam, **video comments** were allowed though.

This setup has numerous advantages. We achieved great audio and video quality (1080p) without notable compression artifacts. As the stream is automatically archived while being streamed, students can also pause the video, watch it time-shifted, or at any later point in time. In addition, all video format issues for different devices are automatically handled by YouTube. Moreover, we do not put any extra load on the university network (if a video is watched from outside the network) which would have to be paid by our university (if billed per volume which is true for Saarland University).

4.4 Audience Response System

To allow for interaction with the students, we needed an audience response system. We wanted something simple: a moderated chat where students could type in questions, a moderator (the tutor in chief) would select suitable questions, and forward them to the lecturer who would then drag a window with the question to the streamed area of the screen, and answer it live during the lecture.

---

[8] https://obsproject.com/.

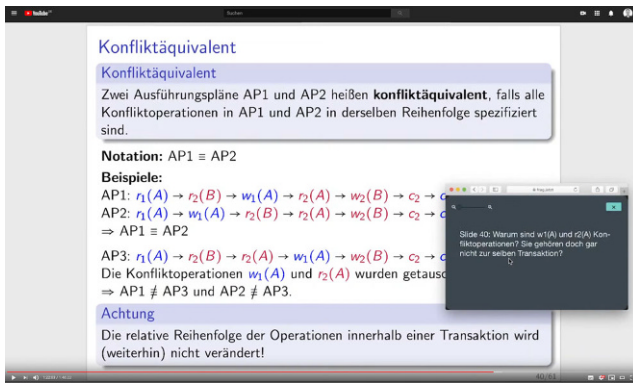[9] We observed a lower lower bound for the stream latency of 2.5 s.

Fig. 4 The student's view of the lecture stream during the reply to an audience question with the browser window of frag.jetzt dragged to the streamed area of the screen

Many different tools are available in this space and after some investigation we decided to use frag.jetzt[10] which is funded by the state of Hesse. It worked well for us. For this tool only the lecturer has to log in to make sure that for each lecture students can use the same URL. See Fig. 4 for the student's view while answering an audience question. The only problem we encountered with this tool is the missing possibility for the lecturer to ask back within the tool.

4.5 Content Management System (CMS)

We used a department-internal course management system for students to register to the course, retrieve lecture material, and hand in assignments. This system has been in use at the department for more than ten years for physical courses as well. It is in many dimensions superior to other systems like Moodle (which we used for three years). The biggest advantage is simplicity. In contrast, Moodle's UI is too complex and confusing. It is sometimes hard to find things and in the student's evaluations it was often criticized.

4.6 Communication Tools

Throughout the course, we used additional tools when needed. In the following, we summarize three tools and provide use cases for each of them.

We used **Discord** for office hours, tutorials, and mini office hours during the lecture break and right after each lecture. Discord is an audio conference tool widely used in the gaming community. The idea is to define virtual rooms which – depending on access rights – users are allowed to join anytime. Once you enter a room, anyone in that room can hear what you say and you can hear immediately what others in that room are saying. Though at first, this sounds

a bit like a standard video conference tool, in fact, it feels more like walking through different (physical) rooms. Discord also allows for screen-sharing and video transmission but at the time of writing both features are restricted to 50 and 25 users per channel, respectively. To join our server, students needed an invitation link that was only visible for students registered in our CMS.

We *boosted*[11] our Discord server to improve screen-sharing capabilities with a higher resolution. In addition, we installed a paid third-party Discord bot called VoiceMaster+. It allows students to create their own temporary voice channels. This was used by students to stream the lecture together and have a bit of a lecture hall feeling. The feature was used to create breakout sub-rooms during tutorial and for students' to form learning groups. Moreover, we also implemented multiple-choice questions for text channels (Pollbot) which were also used in the tutorials.

We also considered using Zoom and MS Teams for tutorials (cf. Sect. 4.3). We announced initially that we would only consider these tools as a backup. In the end there was no need for this and the tutors and most students were happy with Discord. Sometimes we observed stability issues with Discord. In addition, screen sharing and video is still a bit buggy and behind Zoom's abilities. Still, Discord is a great and very helpful tool that will surely further improve in future versions.

We used our CMS allowing students to give **anonymous feedback**. This was used about ten times in total over the entire semester.

Further, we provided Discourse[12], a **textual forum**. We witnessed several interesting discussions and witty student's comments. We made sure to answer questions quickly and the lecturer involved himself into answering the hard questions. Due to Covid-19, the forum was more frequently used than in previous years.

In the tutorials, our students also used a tool for **shared scribbling**: A Web Whiteboard (AWW)[13].

Obviously, this sounds a bit like we used **a zoo of tools** – which is actually true. Even though this is only a very small zoo, it still is a zoo. To fix that problem, we tried to integrate the tools wherever possible and make everything easily accessible. In the end, all the student-facing tools were web-based anyways, so the divide among the different tools was not as severe as it would have been for different desktop applications.

---

[10] https://frag.jetzt.

[11] This is a fancy marketing term used by Discord to describe that we pay a monthly fee to get better services.

[12] https://www.discourse.org/.

[13] https://awwapp.com/.

5 Tutorials, Assignments, and Office Hours

In addition to the lecture, we wanted to give the students the opportunity to deepen their understanding of the course material and ask questions. Therefore, we offered tutorials, office hours, and assignments. In the following, we first discuss our requirements for each of the offers and then explain how we implemented these requirements with the tools discussed in Sect. 4.

5.1 Tutorials

In the past, we offered tutorials for up to 30 students in seminar rooms at our university. Students could vote for their favourite time slots in our CMS and were then assigned accordingly. This year, tutorials were held on our Discord server with a similar number of participating students. For some time slots two tutorials were offered simultaneously. Therefore, we created two categories, one for each virtual seminar room. Each category was structured as follows:

Plenum voice/video channel: Participants and tutor meet here at the beginning of the tutorial. The tutor shares screen, gives instructions, and moderates the discussion.

Temporary voice/video channels: Students form groups of up to five to discuss exercises.

Plenum text channel: Tutors can send files and give instructions in case students are not in the plenum voice channel. Students can ask the tutor to join their temporary channel for questions.

In previous iterations of this lecture, tutors presented the solutions to the assignment handed in most recently. However, we observed that students tended to be very passive and simply wrote down the solutions in these types of tutorials. In fact, we could have simply provided a prerecorded video explaining the solutions but we wanted the students to participate in the tutorial actively.

Therefore, we decided against presenting solutions and instead prepared additional smaller exercises that could be solved during the tutorials. These exercises were explained by the tutors in a presentation-style screencast. Depending on the type of exercise, they were either discussed in the plenum or students were given 10 minutes to form small breakout groups and solve the exercises. We prepared exercises of the following three types:

1. **Multiple-choice exercises**: These exercises were usually discussed in the plenum. Tutors would use our Pollbot to post a poll on the text channel and students would vote by reacting to the post. These exercises were especially helpful to gain insight on which topics were already well understood.
2. **Written exercises**: These exercises were solved in smaller groups using temporary channels. Students had to apply concepts from the lecture to small problems. They often used the online whiteboard app AWW to work together on a solution.
3. **Discussion exercises**: We often asked open questions that engaged the students to participate in a discussion, either in the plenum or in small groups.

We learned that providing exercises to be solved in smaller groups significantly helps students to actively participate in the tutorials. We plan to also keep this approach once tutorials are held physically at the university again.

5.2 Assignments

From previous iterations of the lecture, we learned that encouraging the students to continuously recap and apply the concepts presented in the lecture leads to successful participation in the course. Therefore, we made correctly solving the majority of weekly assignments a prerequisite for admission to the exam.

Assignments should not only verify that students understood the concepts but also challenge academically strong students to think further. Thus, assignments were divided into three parts:

1. **Application tasks**: Students had to directly apply concepts from the lecture to familiar problems.
2. **Transfer task**: Students had to apply concepts to new problems and argue about implications.
3. **Programming task**: Students had to implement algorithms in empty cells in the Jupyter Notebook from the lectures. We provided basic unit tests for them to check their implementations.

Assignments were published on, handed in, and graded using our internal CMS. Since we chose a different approach for our tutorials this year, all solutions to the assignments were also published on our CMS.

As the lecture is aimed at undergrad students, we wanted to support them in practicing teamwork. We decided that assignments could be handed in by groups of up to three members. Since meeting physically to work on the assignments was not an option, we created a workspace category on our Discord server where students could create temporary channels for video calls, sharing their screen, and discussing the assignments.

5.3 Office Hours

Two years ago, we introduced office hours. In contrast to the tutorials that have a fixed structure, we wanted to provide the students with a more open offer to stop by, discuss individual problems, and get help with technical issues. We held office hours once a week for two hours in a seminar room. However, most of the time, participating students just wanted to solve the assignment on-site and have a contact person in case of difficulties.

Therefore, we decided to offer shorter and more frequent office hours of one hour each. Office hours were held by two tutors on our Discord server. Participating students would enter a waiting area (i.e. a public voice channel) and tutors would then accompany them to a private table (i.e. a temporary voice channel). The temporary channels were configured such that other students could not join them to avoid distractions and keep conversations private. If a student wanted to join the conversation, the tutor had to move the student actively to the channel. Students mainly used office hours to receive help with technical difficulties, mostly setting up our environment for the Jupyter Notebooks. They shared their screen in the temporary channel where tutors then guided them towards a solution. We wanted the students to familiarize themselves with Discord and tackle any technical issues as early as possible. Thus, we published the setup instructions and offered office hours two weeks ahead of the start of lectures. Having shorter virtual office hours also resulted in fewer students participating. However, those who decided to stop by usually had a concrete question or problem.

6 Advantages and Issues of the Model

In summary, we are very happy with our teaching model. We also received tremendous positive feedback from our students in two different course evaluations. Students mainly gave two reasons for liking the lecture. First, students enjoyed that the lecture was streamed live with the opportunity to interact and made available as an archived stream. This helped them to better structure their weekly working routine. Second, students liked the practical examples and the application-oriented approach to presenting the topics.

6.1 Advantages

1. We avoided many of the privacy issues with Zoom and MS Teams.

2. The streams and their recordings are easily accessible, support time-shifted viewing, and are automatically archived.
3. There is no need to organize video servers, no load on the university's network (if watched from outside uni), all audio/video-format issues are automatically handled by YouTube, e.g. to watch videos on different devices.
4. As we added video agendas in the video's comment sections, students could easily re-find things in the archived streams.
5. The tutorials with Discord were sometimes perceived to be better(!) than physical tutorials. The main reason was that when working in small teams, students were not distracted by noise from other groups.
6. The public videos on YouTube are a great advertisement for our department and university (and let's hope for relational database technology in general).
7. Videos are replacing textbooks. Our YouTube videos are used by students from other universities. Rather than learning material from classical textbooks, they now use videos to prepare for exams.

6.2 Issues

1. For the lecturer, teaching without seeing the students was extremely exhausting. The main reason is probably that important communication channels for feedback were missing, e.g. the expressions on student's faces, grumbling, laughter. See also "When everyone laughs for themselves" [3] for a discussion on that matter. Teaching often felt like talking to a wall. Sometimes while teaching you forget that you are speaking to two-hundred people.
2. For students, teaming up was sometimes difficult if they were not physically present on campus. This is not so much a problem in the 4th semester (as in this lecture) if they already found peers to team up with. Yet, this seems to be a severe problem for 1st semester students who may not know many peer students yet.

7 Future Plans

In future, though we are not planning to change much, we aim to further improve the concept and fix the remaining issues. Here are some thoughts.

1. How to fix the missing visual backchannel problem? It would help the lecturer a lot to be able to see at least the

faces of a subset of the students [14]. One solution could be to offer some students to additionally join a video conference call (Zoom or Discord) where they *must* switch their camera on. However, it is unclear what the incentive for students could be. In particular, that incentive should be something that does not penalize students not willing to join that video call. A discussion of the psychological motivation and hurdles of students is beyond the scope of this paper and our expertise.

2. In online teaching, there is an epic debate on whether the lecturer should switch the video camera on. What is the added value of this? In this lecture, we did not use the camera. However, we believe that, in particular for new students, it may be more helpful than for older students. Therefore we are considering this for an upcoming first semester lecture.

3. Due to many requests on YouTube, in the long run, we want to make all material available in English.

References

1. Dittrich J (2014) Die Umgedrehte Vorlesung – Chancen für die Informatiklehre. Datenbank Spektrum 14(1):69–76. https://doi.org/10.1007/s13222-013-0143-9

2. Kemper A, Eickler A (2015) Datenbanksysteme: Eine Einführung, 10th edn. De Gruyter Studium. De Gruyter Oldenbourg, Munich

3. Kühl S (2020) Wenn jeder für sich allein lacht. Forschung und Lehre (5). https://www.forschung-und-lehre.de/lehre/wenn-jeder-fuer-sich-allein-lacht-2778/. Accessed 21 Jan 2021

---

[14] Did you ever see students in a physical lecture hall who completely covered their faces? This would not be acceptable in Western societies, yet this is commonly accepted in the virtual world.