# TEK/NAT Kursrapport

<table>
<thead>
<tr>
<th>Kurs</th>
<th>Kurskod</th>
<th>Poäng</th>
<th>År</th>
<th>Start v.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduktion till databashantering</td>
<td>5DV119</td>
<td>7.5</td>
<td>2015</td>
<td>04</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Institution</th>
<th>Antal registrerade (män/kvinnor)</th>
<th>Antal aktiva studenter (deltagit i minst en examinerande del)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Institutionen för datavetenskap</td>
<td>73 (66/7)</td>
<td>71</td>
</tr>
</tbody>
</table>

**Genomströmning (i %) och betygsutfall efter första tillfälle för examination (för varje betyg som satts på kursen ange antal som uppnått detta på formen °°?)**

Genomströmning: 53%  Betyg: U(35) 3(14) 4(20) 5(4)

---

### Hur mycket schemalagd lärar-/assistent-ledd tid har studenten tillgång till på kursen?

46.5 hours

### Hur är undervisningen upplagd?

See the syllabus:

---

### För vart och ett av lärmålen (FSR:en) i kursplanen, beskriv kortfattat hur det examineras.

- **beskriva relationsmodellen för data,**
  *written examination*

- **visa förmåga att deklarera relationella schema med SQL,**
  *written examination*

- **visa bred kompetens i att formulerå frågor och uppdateringar i SQL,**
  *written examination, obligatory exercises*

- **utveckla gränssnitt mot databashanteringssystem med stöd av ODBC med C eller Python som värdspråk,**
  *obligatory exercises*

- **beskriva säkerhetsproblem som SQL-injection och förklara hur de kan undvikas,**
  *written examination*

- **uttrycka frågor i relationsalgebra och relationskalkyl,**
  *written examination, obligatory exercises*

- **förrklara vad begreppsmässig modellering är och hur den används i utveckling av databasschema,**
  *written examination, obligatory exercises*

- **översätta ER-specifikationer till relationella schema,**
  *written examination, obligatory exercises*

- **visa gedigen förståelse för normalformer för relationella schema och beskriva algoritm för att realisera dem,**
  *written examination, obligatory exercises*

---

### Beskriv hur betygssättningen på kursen fungerar. ( Vilka betyg ges på kursen och hur sker bedömningen, dvs vilka delar betygssätts och hur vägs de samman? Finns det skrivtliga betygskriterier och/eller lärmål (FSR) för de olika betygen?)

See the syllabus:

---

### Samläses denna kurs med andra kurser??

- **Nej**

  *Om ja, hur många?*

  *Hur stor andel av kursen samläses?*

---

### Samläser flera program denna kurs?

- **Ja**

  *Om ja, hur många?*

  4

---

### Arbetar studenterna i projektform på kursen?

- **Nej**

  *Om ja, uppskattad omfattning i poäng på projektteken:*
Antal projekt som varje student deltog i:
Antal studenter i projektgrupp:
Förväntades studenterna använda en projektmetodik för dokumentation och styrning (tex LIPS)?
Nej
Hur skedde indelning av studenter i projektgrupper?
Har studenterna uppmanats föra projektdagbok?
Nej
Om ja, Har dagboken utgjort grund för examination?

Kursens samverkan med forskning
Annan samverkansform, nämligen:

Kursens samverkan med näringsliv eller offentlig verksamhet
Ingen samverkan med näringsliv/offentlig verksamhet förekommer på kursen

Annan samverkansform, nämligen

Genomförda förändringar till detta kurstillfälle
The only significant change was that the time for writing the examinations was increased from four hours to six hours.

Förändringsförslag från föregående kursrapport

Lärare

Information om inblandade lärare

Kursansvarig
Stephen Hegner
Antal övrig personal som ej föreläser
2
Antal övriga föreläsare
0

Hur stor del av den schemalagda tiden på kursen undervisas av forskande lärare (dvs lärare med mer än 25% forskning i sin tjänst)?
0%

Hur stor del av den schemalagda tiden på kursen undervisas av lärare verksamma i näringsliv/offentlig verksamhet (dvs lärare med mer än 25% av sin tjänst förlagd till näringsliv/offentlig verksamhet)?
0%

Kursvärd.

Totalt antal svarande
16
Sammanställningsdatum
2015-04-24
När genomfördes kursvärderingen?
Efter genomfört första examinationstillfälle

För varje lärmål på kursen ange hur stor del av de studerande som uppger att det har behandlats på kursen - ange svaret i procent på formen
har behandlats/har inte behandlats/vet ej

beskriva relationsmodellen för data,
15/0/1 (yes/no/unsure)
visa förmåga att deklarera relationella schema med SQL,
15/0/1
visa bred kompetens i att formulera frågor och uppdateringar i SQL,
15/0/1
utveckla gränssnitt mot databashanteringssystem med stöd av ODBC med C eller Python som värdspråk,
14/1/1
beskriva säkerhetsproblem som SQL-injection och förklara hur de kan undvikas.
15/0/1
uttrycka frågor i relationsalgebra och relationskalkyl,
15/0/1
förklara vad begreppsmässig modellering är och hur den används i utveckling av databasschema
13/0/2
översätta ER-specifikationer till relationella schema.
15/0/1
visa gedigen förståelse för normalformer för relationella schema och beskriva algoritmer för att realisera dem.
13/0/3

Sammanf.

Sammanfattning av åsikterna i kursvärderingen - positivt och negativt kring föreläsningar, seminarier, grupparbeten, laborationer, examination etc
All comments from the evaluations are presented here.

Fråga 1. What was positive about the course and should be retained?

The laborations
- the overall structure
- the practical assignment (C/Python)
Great lectures and a simple and clear sectioning of the course items. The exercises clearly correlates with the goals for the course.

The laborations.
The slides, the obligatory exercises, the overall setup.
*The slides were very good. No book was needed as everything was covered in the slides.

6 hours for the exam was a great change because queries are a bit challenging and they can take some time to be figured out. Unlike some other type of questions where you are either prepared or not.*
Good obligatory exercises!
The lecture is good the way there are.
Most of it was good.
Possibility to solve the labs in groups.

Fråga 2. What can be improved? If possible, please explain how.

Same thing is being examed both in the obligatory assignments as on the written exam. Makes it feels like the obligatory assignments shouldn’t be obligatory, or the exam should focus on other things.
The exam. More laborations, less focus on exam
*The definitions and algorithms presented on the slides “Functional Dependencies and Normalization” were very difficult to understand. A possible reason: too much mathematics. Youtube proves the same things can be explained in much more intuitive ways.

6 hours provided for the exam is still not enough to cover all the questions, unless a student is approaching the SQL questions by memorizing the answers to similar questions from the previous years. Approaching the SQL problems by thinking from zero required more time.*
2.5 days is not enough time for some of the obligatory exercises.

Vetter explanation of normalization.
The final examination. The questions were more difficult than usual.
The instructors had very little knowledge for answering questions about the assignments and normalization.
*Normalization could be split up and studied more thoroughly. More real life examples and common industry practices.
More exercises about normalization, preferably in small groups rather in one big lecture.
Rearrange the lab deadlines so that they don’t happen to be early morning after the weekend.

Could be good with practical teacher held exercises and lectures

Fråga 3. Did you find something missing from the course which should have been included?

More examples how to solve a bit more complex queries in relational algebra, tuple calculus and sql exams with solutions. Better explanation of the BCNF algorithm.

No
Insight into how large databases are typically managed and administered in real life.
Not that I can think of.

Lärarnas synpunkter på kursens innehåll och genomförande
First, it is important to repeat the major observation from previous reports. The course has a very diverse audience. At one extreme are the students in the technical computer science program, who typically have taken at least eleven courses in CS. At the other extreme are students from other non-CS programs, who often have taken only two CS courses. It is virtually impossible to give a course which meets the needs of these two distinct groups simultaneously. The CS students have much more experience in programming and in using CS-related ideas, work with computers all day, every day, can assimilate new material much more rapidly, and expect a more in-depth presentation. Combining a mid-level course which is required for CS majors with a service course does not work well. This argument for giving just one course is financial; it is less expensive to do so, and the resources to give two separate courses are not available. This does not, however, make designing and running such a course any more feasible.

A further issue regarding this “one-size-fits-all” course philosophy impacts the course evaluation directly. It is not indicated from which group of students a given comment in this evaluation comes, which makes it even more difficult to address the associated issues.

The individual comments will next be addressed. It should be remembered, however, that fewer than 25% of the registered students completed the evaluation, and no point was noted by more than three students. Thus, major changes in the course organization which are based upon the student comments could easily create more problems than they solve.

The overlap between laboratory exercises and examination questions is deliberate. Some students like this and others do not. The reason why topics (such as SQL programming) are on the examinations, even though they are covered thoroughly in the exercises, is that the exercise scores provide a poor measure of individual competence. It is not uncommon to see a student with almost perfect scores on the SQL exercises be unable to solve even the simplest problem on the examination. Of course there can be several reasons for this, but the most common is clearly that the student relied upon lab partners to solve the problems. The instructor has a responsibility to evaluate the competence of the students in a reasonable fashion, and in many cases the exercises alone do not provide an adequate way to do this.

Initially, the plan was to have the laboratory exercises due on Friday afternoon. It was subsequently decided to give the students until Monday morning. It is not clear whether the students who complain about a deadline on Monday morning would really prefer to have the deadline moved forward to Friday afternoon, but from an administrative point of view this would not pose any difficulty.

Regarding the length and difficulty of the examination, 86% of the students who wrote the one given in March received a passing grade, with over 50% achieving at least a 4. Such results are not consistent with a general claim that the examinations are too long or too difficult. There are a few students every year who complain about the examinations, but the success rate is very high overall. There may nevertheless be a significant issue that non-CS students have much more difficulty in this respect than CS students, but the anonymity of responses hinders any effort to determine whether this is the case.

Regarding examination solutions for study, I suggest that student groups assemble collections of old examination papers which received good grades, as is done at some universities in Germany.

Regarding normalization, I agree that this is a difficult topic. I also agree that there are other presentations which are simpler and easier to understand. The problem is that they are mathematical exercises which ignore key concepts which are central to a meaningful solution. Sorry, YouTube does not explain the same thing. All of these “simplified” presentations ignore independence, and some ignore other important issues as well. Normalization to a non-independent representation creates more problems than it solves. Students need to appreciate this, even if they do not understand all of the details. If normalization is really too difficult for this course, then it should be moved to a more advanced course. Presenting dumbed-down normalization to satisfy some FSR superficially is counterproductive to a good education.

I fully agree that group exercises are worth considering. They were once common in the CS department, but have been largely dropped in all but introductory courses due to budgetary issues. It is my understanding that the financial resources for them are not available for courses at the level of 5DV119, but if they are, I would suggest strongly that they become an optional part of the course. On the other hand, when we had them during the late 1990s for the introductory DBMS course which was then given, very few students participated.

Finally, I agree that staffing for this course is a difficult problem. There are very few staff members in the Department with experience in DBMS, and one of them was unfortunately unavailable this year, due to other commitments. The Department clearly needs to address this issue. On a more positive note, I suggest that the main instructor (kursansvarig) be given hours for consultation, and perhaps even limited grading of exercises. In the current system, course duties are very precisely identified by category, and the main lecturer almost never is assigned consultation hours. This means that the assistant instructors have the responsibility for answering all questions, even those about the conceptual ideas presented in the lectures, even though the main lecturer may have a much more comprehensive grasp of the subject matter. Since instructors are very heavily loaded with duties assigned by this system, it is essential that explicit hours for consultation be allocated.

It might be preferable to move the deadlines for exercises from Monday morning to the preceding Friday afternoon. Although the students would actually have less time to complete them, a deadline of Friday afternoon my be more acceptable psychologically.

It is recommended that the main lecturer (kursansvarig) be allocated hours for consultation, and possibly even grading assistance for certain exercises.

It is also suggested that resources for group exercises be allocated to the course.

Note that the last two changes would require administrative support. They cannot be implemented by the instructor(s) alone.
<table>
<thead>
<tr>
<th>Rolle</th>
<th>CAS-identitet</th>
<th>Namn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Granskare lärare</td>
<td>sthe0001 [Hegner, Stephen]</td>
<td></td>
</tr>
<tr>
<td>Granskare student</td>
<td>suka0016 [Kadelbach, Susanne]</td>
<td></td>
</tr>
<tr>
<td>Granskare studieadministratör</td>
<td>leka0001 [Kallin Westin, Lena]</td>
<td></td>
</tr>
</tbody>
</table>

Eventuella kommentarer på granskningsprocessen