

II. Functional dependencies

D(studentIdnr, studentName, login, branchName, programName, courseCode, courseName, credits, departmentName, capacity, classification, grade, position)

a)

- 1) studentIdnr -> studentName //the students name
- 2) studentIdnr -> login //the students login name
- 3) studentIdnr -> programName //students id should tell you which program that student belongs to
- 4) studentIdnr -> branchName //students id should tell you which branch that student belongs to
- 5) studentIdnr courseCode -> position //the students position in a waiting list for a course (if in a waiting list for that course)
- 6) studentIdnr courseCode -> grade //the students grade for a course (if they have taken that course)
- 7) courseCode -> courseName //each course has a name
- 8) courseCode -> credits //each course gives a certain number of credits
- 9) courseCode -> capacity //if the course has a limited capacity
- 10) courseCode -> departmentName //each course is given by 1 department

//read the Domain Description in task 0 to find the remaining FDs - fråga under handledning

- 11) login -> studentIdnr //each student login is also unique
- 12) courseCode position -> studentIdnr

b)

Schema of the FDs above using **BCNF normalization** by hand:

BCNF Normalisation algorithm

To normalize relation R:

This FD is referred to as a BCNF-violation

Find a non-trivial FD $X \rightarrow y$ such that $X^+ \neq R$ (X is not a superkey)

If there is no such FD you are done, R is already in BCNF

Otherwise decompose R into $R_1(X^+)$ and $R_2(X \cup (R - X^+))$ and normalize them

Note: R is replaced by R_1 and R_2 (so R is not present in the final schema)

R1 (studentIdnr, login, studentName, programName, branchName) ✓

R2 (studentIdnr, courseCode, courseName, credits, departmentName, capacity, classification, grade, position) ✗ **BCNF VIOLATION**

R2_1 (courseCode, courseName, credits, capacity, departmentName) ✓

R2_2 (studentIdnr, courseCode, classification, grade, position) ✗ **BCNF VIOLATION**

R2_2_1 (studentIdnr, courseCode, grade, position) ✓

R2_2_2 (studentIdnr, courseCode, classification) ✓

Result

R1 (studentIdnr, login, studentName, programName, branchName) ✓

R2_1 (courseCode, courseName, credits, capacity, departmentName) ✓

R2_2_1 (studentIdnr, courseCode, grade, position) ✓

R2_2_2 (studentIdnr, courseCode, classification) ✓

c)

MVD

Course classifications seems like a natural use of MVDs.

Relation **R2_2_2** can be further normalized using **4NF** normalization

R2_2_2 (studentIdnr, courseCode, classification) **✗ 4NF VIOLATION**

MVD: courseCode ->-> classification

Result

R2_2_2_1 (courseCode, classification)

R2_2_2_2 (courseCode, studentIdnr)