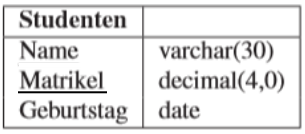
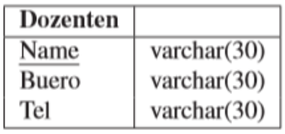
# Exercise 04: SQL DML: SFW blocks

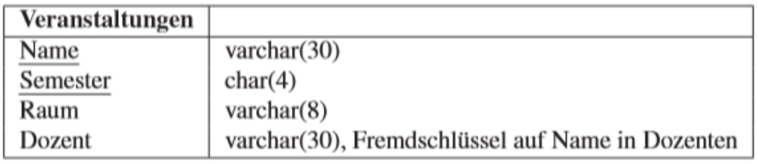
In the last exercise you created the Students, Lecturers, Events and Student_in_Event tables with the following attributes and keys:



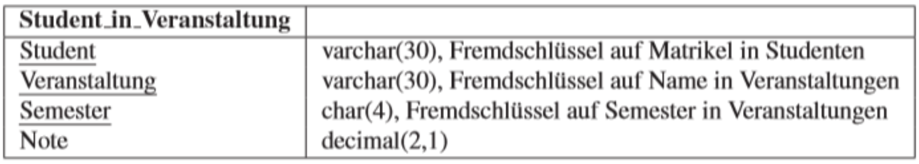
| Studenten | |
|---|---|
| Name | varchar(30) |
| Matrikel | decimal(4,0) |
| Geburtstag | date |



| Dozenten | |
|---|---|
| Name | varchar(30) |
| Buero | varchar(30) |
| Tel | varchar(30) |



| Veranstaltungen | |
|---|---|
| Name | varchar(30) |
| Semester | char(4) |
| Raum | varchar(8) |
| Dozent | varchar(30), Fremdschlüssel auf Name in Dozenten |



| Student_in_Veranstaltung | |
|---|---|
| Student | varchar(30), Fremdschlüssel auf Matrikel in Studenten |
| Veranstaltung | varchar(30), Fremdschlüssel auf Name in Veranstaltungen |
| Semester | char(4), Fremdschlüssel auf Semester in Veranstaltungen |
| Note | decimal(2,1) |

### Task 1: Simple selects

a) Create a list of all lecturers who have their office in wing D.
b) Create a list of matriculation numbers of students who have not yet received a grade in the summer semester 2018.
c) Create a list with the headings Student and Age, in which the names of the students and their ages are listed. Prevent the output of students who are younger than 20 and older than 40. The required T-SQL commands are `current timestamp` and `datediff`.

Check your solutions using the solution of the last exercise sheet or enter suitable test data yourself.

### Task 2: Join and cross product

a) Create a room plan for your fellow students for the summer semester 2018 that shows who is in which event in which room. In doing so, do not output any unnecessary columns and use meaningful identifiers for the output columns.
b) Create a table in which the name of a student is in the first column and the name of an older student is in the second column with the label *"..is older than"* . Generate all possible combinations of rows with this relationship and sort the output based on the first column, with the oldest student displayed first.

c) Create a table with only one column, which you call Text. In each row, there is a text based on the following pattern: *"Eva took part in the beach volleyball event in the summer semester 2018 and obtained the grade 4.0. Congratulations!*" Only output rows for the winter semester 2017, the summer semester 2017 and the summer semester 2018. Adjust the text for each line accordingly and also address the grade (for example *"failed"* or *"no grade received yet*").

**Task 3: Quantifiers and set operations**

a) Create a table which shows the best grade a lecturer has ever given for a specific event across all semesters.
b) Extend the above task to also indicate the worst grade given.
c) Additional task: how can you easily check whether `Student`, `Event` and `Semester` are keys in `Student_in_Event`? How can you output only the tuples that violate the key property?