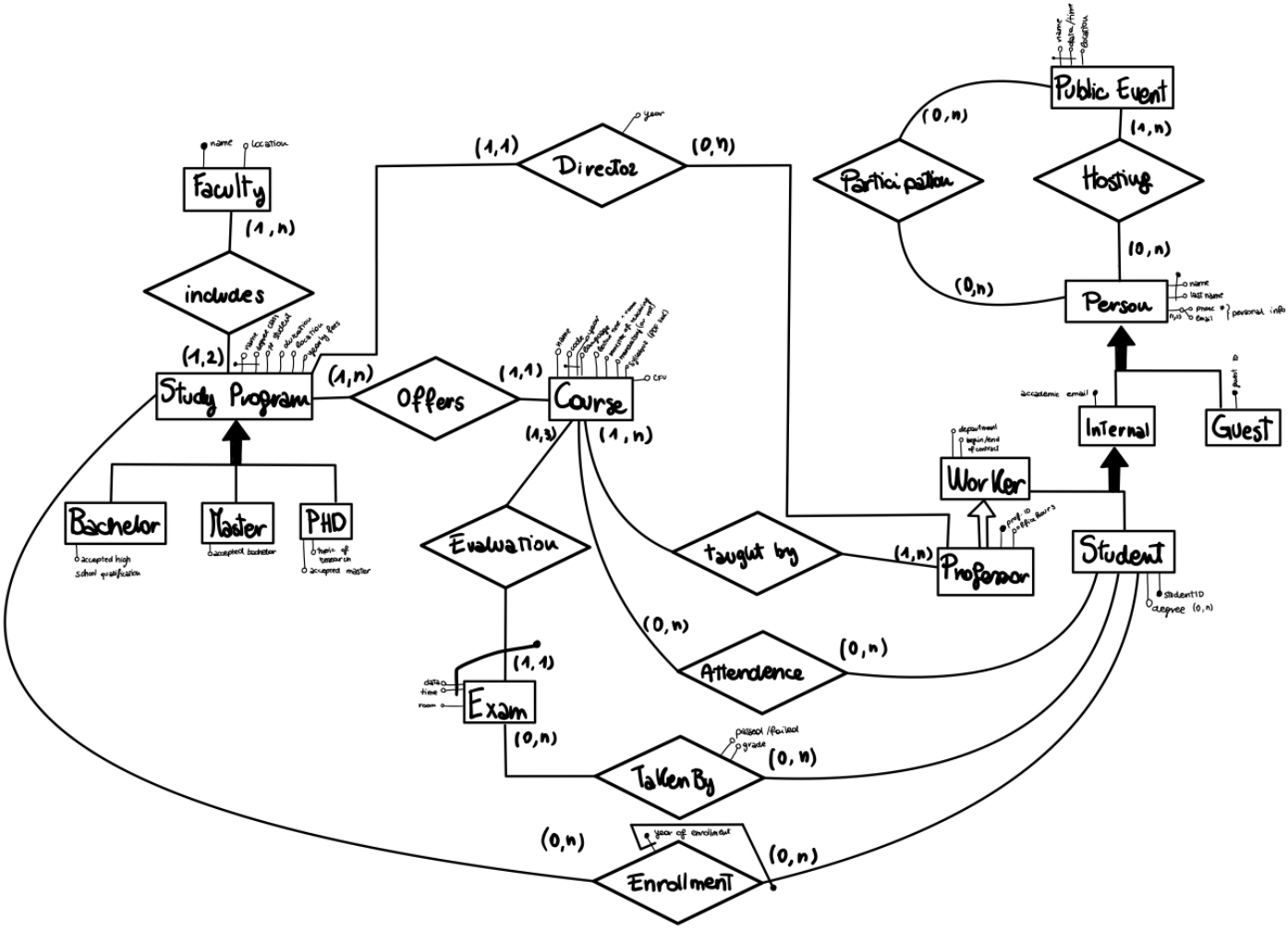
Introduction to Databases 2022-2023

Students: Irene Avezzù and Ivana Nworah Bortot

Project title: University database (inspired by Unibz)

Developed during the first semester of the academic year 2022-2023

Phase 1 – Conceptual Design



STRUCTURE AND ORGANIZED REQUIREMENTS:

Our Database regards the internal structure of a university. It represents the connections between the workers, students, departments, and course.

The University is made up of **Faculties**, which have a name, which is unique, and a location. Each faculty includes at least one study program, and each study program can be included in at least one and at most two faculties.

Each **study program** has a name and a degree class (classe di laurea), that are unique. It also has a max number of students, a duration, a location and the yearly fees. Each study program can be included in at most of 2 faculties. A Study Program is either a **Bachelor** (accepted High School Qualification), a **Master** (accepted bachelor's degrees) or a **PhD** (accepted Master, topic of research). Each Study Program is also directed by one Professor. The number of students who can enrol in Study Program is limited by the max number of student and can have none when the course gets first added into the system.

Each study program offers at least one **Course** that is identified by the code and the year it gets taught. It also has a name, the CFU, a language in which it is taught, the lecture time and room, the semester of teaching, if it is mandatory or not, and the syllabus (link to pdf file). Each Course refers to only one Study Program. Each Course is taught by at least one Professor.

Each course gets evaluated by an **Exam** which is uniquely identified by the course, the date, the time and the room in which it takes place. For each course there are at most three possible exams per academic year (which is an identifier for the course).

A **Person** who is identified by a name and a last name and has also personal information (phone number, email). A Person can be an **Internal,** with an assigned academic email, or a **Guest**, identified by guestID. An internal can be a Worker or a Student.

Each **Worker** is assigned to a department and has a contract with begin and end date. A **Professor** is a type of Worked that is identified by the professorID and offers office hours. Each Professor teaches at least one course (multiple Professors can teach the same course).

A **Student** is identified by the studentID and can have previous degrees. A Student can enrol in one Study Program in an academic year (which uniquely identify the year of enrolment). Each Student attends as many/any courses as he/she wants. Each Course gets thought only if there are at least 10 students attending it. The limit of the number of students who can attend a Couse is limited by the students' number limit defined by each study program. Each Student can take as many/any exam as he/she wants. Once a student takes an exam, he/she can either pass it or fail it, but it always gets a grade.

The University can host **Public Event**s, which are identified by the name and date and time they are hosted and the location, and they can be hosted by one or multiple people. The Events are hosted only if at least 10 people wants to participate.

DATA DICTIONARY: ENTITIES

ENTITY	DESCRIPTION	ATTRIBUTES	IDENTIFIERS
Faculty	Faculty of a university	name, location	{name}
Study Program	Study programs of a faculty	name, degreeClass, numberOfStudents, location, durations, yearlyFees	{name, degreeClass}
Course	Courses of the study programs	code, year, name, language, lectureTime, room, semester, mandatory(orNot), syllabus, CFU	{code, year}
Guest	External people from the university	guestID	{guestID}
Internal	Staff/Students of the university	academicEmail	
Student	Student in the university	studentID, degree	{studentID}
Worker	Worker of the university	name, lastName, department, begin/end of contract	{name, lastName}

Professor	Professor teaching in the university	professorID, officeHours	{professorID}
Public Event	Events hosted in/by the university	name, date, hour, location	{name, date}
Exam	Exam of the course	data, time, room	{data, time, course}
Person	A person related to the university	name, lastName, personal information (phone, email)	{name, lastName}
Bachelor	Bachelor programs of the university	acceptedHighSchoolQualification	{}
Master	Mater programs of the university	acceptedBachelors	{}
PhD	PhD programs of the university	topic of research, acceptedMaster	{}

DATA DICTIONARY: RELATIONSHIPS

Relationship	Description	Components	Attributes
Includes	Inclusion of a study program in a faculty	Faculty, StudyProgram	
Offers	Inclusion of a course in a study program	StudyProgram, Course	
Evaluation	Which exam is used to evacuate the course	Course, Exam	
Enrolment	Enrolment of a student in a study Program	Student, Study Program,	EnrollmentYear, Student
TaughtBy	Professor Who teaches a course	Professor, Course	
Taken	Enrolment of a student in an exam	Student, Exam	passed/failed, grade
Director	Professor who is the director of the study program	Professor, StudyProgram	
Attendance	Attendance of a student in a course	Student, Course	
Participation	Participation of a person in a public event	Person, PublicEvent	
HostedBy	Person who hosts a public event	Person, PublicEvent	

DATA DICTIONARY: EXTERNAL CONSTRAINS

External Integrity Constrains	
1	The number of students who can enrol in Study Program is limited by the max number of student and can have none when the course gets first added into the system.
2	For each course there are at most three possible exams per academic year
3	Each Course gets thought only if there are at least 10 students attending it.

4	The limit of the number of students who can attend a Couse is limited by the students' number limit defined by each study program.
5	The Events are hosted only if at least 10 people wants to participate.

GLOSSARY OF TERMS

Term	Description	Synonyms	Connections
Faculty	Faculty of a university		Study Program
Study Program	Study programs of a faculty	Degree	Student, Course, Professor, Bachelor, Master, PhD
Course	Courses of the study programs		Exam, Student, Professor
Internal	External people from the university	Insider	Person
Student	Staff/Students of the university	Graduate, Undergraduate	Internal, Person, Course, Exam, Study Program
Worker	Student in the university	Employee	Internal, Person, Professor
Guest	Worker of the university	Visitor	Person
Professor	Professor teaching in the university	Lecturer	Course, Study Program, Worker
Public Event	Events hosted in/by the university	Festival	Person
Exam	Exam of the course	Assessment, Examination	Course, Student
Person	A person related to the university	Individual	Public Event, Internal, Worker, Professor, Student, Guest
Bachelor	Bachelor programs of the university		Study Program
Master	Mater programs of the university		Study Program
PhD	PhD programs of the university		Study Program

Includes	Inclusion of a study program in a faculty	Faculty, StudyProgram
Offers	Inclusion of a course in a study program	StudyProgram, Course
Evaluation	Which exam is used to evacuate the course	Course, Exam
Enrolment	Enrolment of a student in a study Program	Student, Study Program
TaughtBy	Professor Who teaches a course	Professor, Course
Taken	Enrolment of a student in an exam	Student, Exam
Director	Professor who is the director of the study program	Professor, StudyProgram
Attendance	Attendance in a course	Student, Course
Borrow	Borrowing of a book	Person, Book
Participation	Participation in a public event	Person, PublicEvent
HostedBy	Who hosted the public event	Person, PublicEvent

TABLES OF VOLUMES ACCORDING TO APPLICATION LOAD

CONCEPT	CONSTRUCT	VOLUME	NOTES
Faculty	Entity	5	
Study Program	Entity	30	3 bachelor + 2 master + 1 PhD for each faculty
Course	Entity	645	25x3 bachelor + 18x2 master for each faculty
Guest	Entity	200	
Internal	Entity	5200	Students + workers
Student	Entity	4100	

Worker	Entity	1100	300 of stuff + professors
Professor	Entity	800	
Public Event	Entity	50	
Exam	Entity	1935	Courses x3 sessions
Person	Entity	5400	Internal + guest
Bachelor	Entity	15	3 bachelors in each faculty (on average)
Master	Entity	10	2 masters in each faculty (on average)
PhD	Entity	5	1 PhD in each faculty (on average)
Includes	Relationship	30	6 for each faculty
Offers	Relationship	645	3*25 + 2*18 for each faculty
Evaluation	Relationship	1935	3 for each course
Enrolment	Relationship	4100	1 for each student
TaughtBy	Relationship	1935	3 for each course (on average)
TakenBy	Relationship	32800	10 exams (on average) for 80% of student
Director	Relationship	30	1 for each study program
Attendance	Relationship	2500	60% of students
Participation	Relationship	5000	100 people for each event
HostedBy	Relationship	100	2 for each event

TABLE OF OPERATIONS

Operation:

1. Create a new user (professor, student, worker, guest)
2. Create a new faculty
3. Create a new study program
4. Create a new course
5. Create a new public event
6. Create a new exam
7. Assign a professor to be the director of a study program
8. Enroll a student in a study program
9. Enroll a student in a course
10. Enroll a student into an exam
11. Assign a professor to be the teacher of a course
12. Assign a person to be the host of an event (event x 2)
13. Assign a person to attend an event (event x 25)
14. Assign an exam to a course (course x 3)
15. Get top 10 students (batch)
16. Edit personal information of a person
17. Calculate the average grade of a student
18. Get timetable of a course (batch)
19. Get upcoming events (batch)
20. Get enrolled students in a study program
21. Get available courses for a student (those for which he/she still has not passed the exam)

OPERATION	TYPE	FREQUENCY	NOTES
1	Interactive	1470/year	total/3 (students) + 10 (professors) + 10 (worker) + 100 (guest)
2	Interactive	1 / 5 years	On average every 5 years a new faculty is added into the university
3	Interactive	1 / 3 years	On average every 3 years a new study program is added into the university

4	Interactive	645/year	Every year new courses are created because one of the identifiers is the year it gets taught
5	Interactive	50/year	
6	Interactive	1935/year	For each course 3 new exam are created (645x3)
7	Interactive	10/ year	On average every 3 year the director of a study program gets changed (30 Study Program / 3 years)
8	Interactive	1350/year	Every year about 1/3 of the students is a new student who need to enroll in a study program
9	Interactive	17700/year	Every year on average each course of the bachelor is attended by 40 students and each course of the master by 15. 375x40 (bachelor) + 180x15 (Master)
10	Interactive	41000/year	On average each student enroll in 10 exams (4100*10)
11	Interactive	1935/year	Each course has on average 3 professor (645x3)
12	Interactive	100/year	Each event has on average 2 host
13	Interactive	2500/year	Each event has on average 50 people attending
14	Batch	1935/year	Each course is assigned to 3 exams
15	Bach	1/year	When the "Dean letter" is published the top 10 students are calculated.
16	Interactive	1350/year	On average ¼ of the people change once one of their personal information ((guest + internal)/4)
17	Interactive	12300/year	After every exam session the average of each student is calculated (student x 3)
18	Bach	645/year	To publish the timetable on the website they need to be retrieve once a year for each course
19	Bach	12/year	Every month a list of upcoming events is published
20	Interactive	30/year	Every year a list of the enrolled student for each study program is published
21	Interactive	8200/year	Each semester each student gets information regarding the available courses (students x 2)

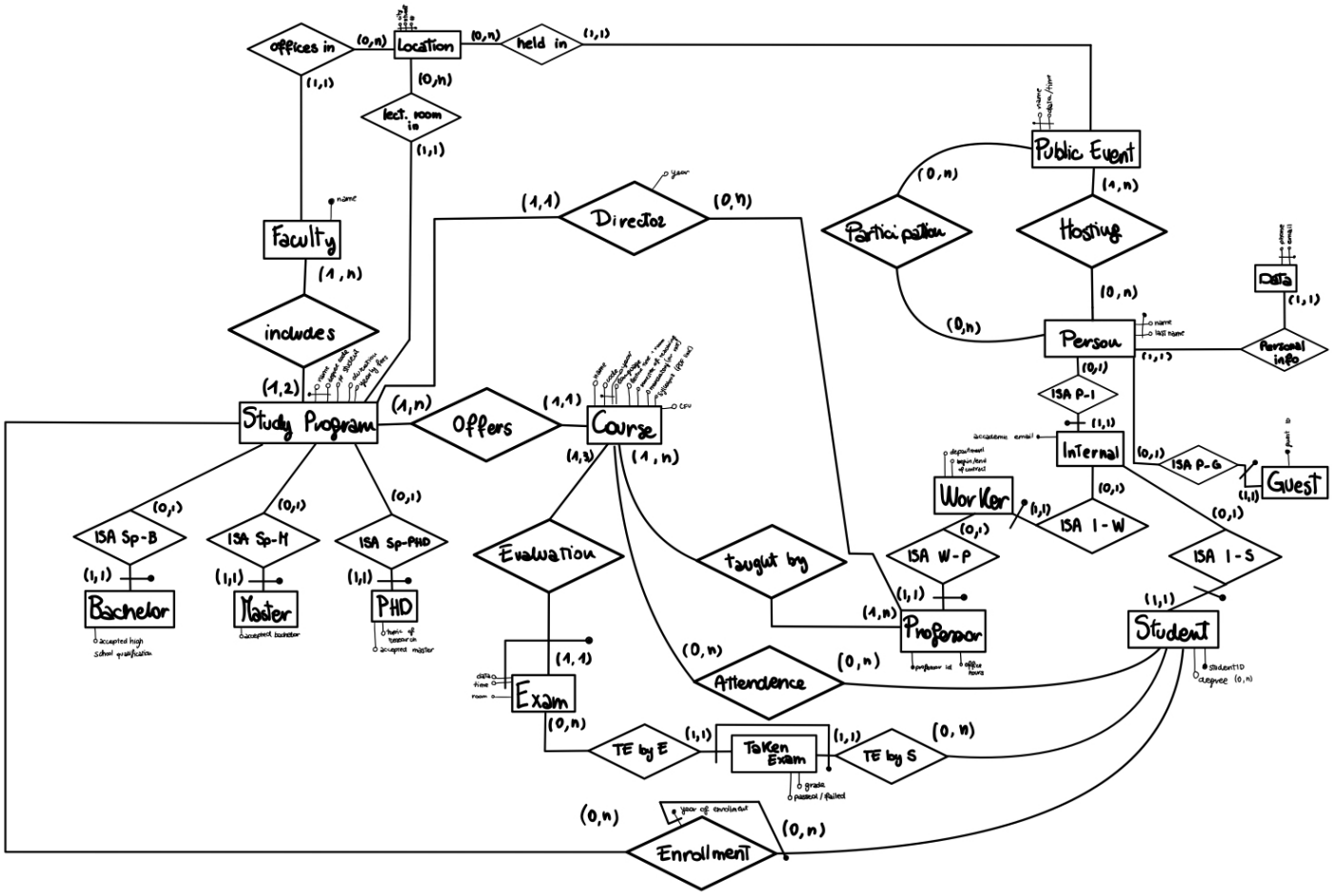
ACCESS TABLE

CONCEPT	CONSTRUCT	ACCESS	TYPE
		Operation 1	
Person	Entity	1	W
Guest/Internal	Entity	1	W
Worker/Student	Entity	1	W
Professor	Entity	0-1	W
		Operation 2	
Faculty	Entity	1	W
		Operation 3	
Study program	Entity	1	W
Faculty	Entity	1	R
Includes	Relationship	1	R
		Operation 4	
Course	Entity		W
Study program	Entity		R
Offers	Relationship		W
		Operation 5	
Public Event	Entity	1	W

Hosting	Relationship	2	W
Person	Relationship	2	R
Operation 6			
Exam	Entity	1	W
Course	Entity	1	R
Evaluation	Relationship	1	W
Operation 7			
Professor	Entity	1	R
Director	Relationship	1	W
Study program	Entity	1	R
Operation 8			
Student	Entity	1	R
Study program	Entity	1	R
Enrollment	Relationship	1	W
Operation 9			
Student	Entity	1	R
Course	Entity	1	R
Attendance	Relationship	1	W
Operation 10			
Student	Entity	1	R
Exam	Entity	1	R
Taken Exam	Entity	1	W
Operation 11			
Professor	Entity	1	R
Course	Entity	1	R
Taught by	Relationship	1	W
Operation 12			
Person	Entity	1	R
Public Event	Entity	1	R
hosting	Relationship	1	W
Operation 13			
Person	Entity	1	R
Public Event	Entity	1	R
Participation	Relationship	1	W
Operation 14			
Exam	Entity	1	R
Course	Entity	1	R
Evaluation	Relationship	1	W
Operation 15			
Student	Entity	4100 (one for each student)	R
Taken Exam	Entity	4100	R
TE by S	Entity	4100	R
Operation 16			
Person	Entity	1	W
Operation 17			
Student	Entity	1	R
Taken Exam	Entity	n (n = number of exams taken by the student, on average = 15)	R

TE by S	Entity	n	R
		Operation 18	
Course	Entity	1	R
		Operation 19	
Public Event	Entity	n (n = number of all PE existing, on average 50 per year)	R
		Operation 20	
Student	Entity	n (n = number of students enrolled, on average 6 for bachelor, 50 for master, 2 for PhD)	R
Study program	Entity	n	R
Enrollment	Relationship	n	W
		Operation 21	
Student	Entity	1	R
Evaluation	Relationship	x (x = number of exams taken by the student, on average = 15)	R
Exam	Entity	x	R
TE by E	Relationship	x	R
Taken Exam	Entity	x	R
TE by S	Relationship	x	R
Course	Entity	y (y = offered courses for Study Program)	R
Enrollment	Relationship	1	R
Study program	Entity	1	R
Offers	Relationship	y	R

Phase 2 – Restructuring of the conceptual schema



DATA DICTIONARY: ENTITIES

ENTITY	DESCRIPTION	ATTRIBUTES	IDENTIFIERS
Faculty	Faculty of a university	name, location	{name}
Study Program	Study programs of a faculty	name, degree code, number of students, location, durations, yearly fee	{name, degree class}
Course	Courses of the study programs	code, year, name, language, lecture time, room, semester, mandatory(orNot), syllabus, CFU	{code, year}
Guest	External people from the university	guest ID	{guest ID}
Internal	Staff/Students of the university	academic email	
Student	Student in the university	student ID, degree	{student D}
Worker	Worker of the university	name, last name, department, begin/end of contract	{name, last name}
Professor	Professor teaching in the university	professor ID, office hours	{professor ID}
Public Event	Events hosted in/by the university	name, date, hour, location	{name, date}
Exam	Exam of the course	data, time, room	{data, time, Course}

Person	A person related to the university	name, lastname	{name, lastname}
Bachelor	Bachelor programs of the university	accepted high school qualification	{}
Master	Mater programs of the university	accepted bachelors	{}
PhD	PhD programs of the university	topic of research, accepted master	{}
Location	The different complexes of the university	city, street, number	{city, street, number}
Taken Exam	Status of an exam	grade, passed/fail	{Exam, Student}
Data	Personal information of a Person	phone, personal email	{phone, personal email}

DATA DICTIONARY: RELATIONSHIPS

Relationship	Description	Components	Attributes
Includes	Inclusion of a study program in a faculty	Faculty, Study Program	
Offers	Inclusion of a course in a study program	Study Program, Course	
Evaluation	Which exam is used to evacuate the course	Course, Exam	
Enrolment	Enrolment of a student in a study Program	Student, Study Program,	Enrolment Year, Student
TaughtBy	Professor Who teaches a course	Professor, Course	
TbE	Taken by Exam	Status, Exam	
TbS	Taken by Student	Status, Student	
Director	Professor who is the director of the study program	Professor, Study Program	
Attendance	Attendance of a student in a course	Student, Course	
Participation	Participation of a person in a public event	Person, Public Event	
HostedBy	Person who hosts a public event	Person, Public Event	
offices in	Location of a Faculty office	Faculty, Location	
lecture rooms in	Location of the lecture room of a Study Program	Study Program, Location	
held in	Location in which a Public Event is held	Public Event, Location	
Personal info	Information of a Person	Person, Data	
ISA Sp-B	ISA between Study Program and Bachelor	Study Program, Bachelor	
ISA Sp-M	ISA between Study Program and Master	Study Program, Master	
ISA Sp-PhD	ISA between Study Program and PhD	Study Program, PhD	
ISA W-P	ISA between Worker and Professor	Worker, Professor	
ISA I-W	ISA between Internal and Worker	Internal, Worker	
ISA I-S	ISA between Internal and Student	Internal, Student	
ISA P-I	ISA between Person and Internal	Person, Internal	
ISA P-G	ISA between Person and Guest	Person, Guest	
TE by E	Result of the restructure of the Exam-Student relation	Exam, Student	
TE by S	Result of the restructure of the Exam-Student relation	Exam, Student	

DATA DICTIONARY: EXTERNAL CONSTRAINS

External Integrity Constrains	
1	The number of students who can enrol in Study Program is limited by the max number of student and can have none when the course gets first added into the system.
2	For each course there are at most three possible exams per academic year
3	Each Course gets thought only if there are at least 10 students attending it.
4	The limit of the number of students who can attend a Couse is limited by the students' number limit defined by each study program.
5	The Events are hosted only if at least 10 people wants to participate.
6	Internal is always an instance of Worker or Student
7	Person is always an instance of Internal or Guest
8	Study Program is always an instance of Bachelor or Master or PhD

TABLES OF VOLUMES ACCORDING TO APPLICATION LOAD

CONCEPT	CONSTRUCT	VOLUME	NOTES
Faculty	Entity	5	
Study Program	Entity	30	3 bachelor + 2 master + 1 PhD for each faculty
Course	Entity	645	25x3 bachelor + 18x2 master for each faculty
Guest	Entity	200	
Internal	Entity	5200	Students + workers
Student	Entity	4100	
Worker	Entity	1100	300 of stuff + professors
Professor	Entity	800	
Public Event	Entity	50	
Exam	Entity	1935	Courses x3 sessions
Person	Entity	5400	Internal + guest
Bachelor	Entity	15	3 bachelors in each faculty (on average)
Master	Entity	10	2 masters in each faculty (on average)
PhD	Entity	5	1 PhD in each faculty (on average)
Location	Entity	3	Bozen, Brixen, Brunek
Status	Entity	86100	Students*25
Data	Entity	5400	The same as the number of people
Includes	Relationship	30	6 for each faculty
Offers	Relationship	645	3*25 + 2*18 for each faculty
Evaluation	Relationship	1935	3 for each course
Enrolment	Relationship	4100	1 for each student
TaughtBy	Relationship	1935	3 for each course (on average)
Director	Relationship	30	1 for each study program
Attendance	Relationship	2500	60% of students
Participation	Relationship	5000	100 people for each event
HostedBy	Relationship	100	2 for each event
TbE	Relationship	1935	For each exam
TbS	Relationship	102500	A student can try an exam multiple time
offices in	Relationship	5	1 for each faculty
lecture rooms in	Relationship	30	1 for each study program
held in	Relationship	50	1 for each Public Event

info	Relationship	5400	1 for each Person

TABLE OF OPERATIONS

Operation:

- Create a new user (professor, student, worker, guest)
- Create a new faculty
- Create a new study program
- Create a new course
- Create a new public event
- Create a new exam
- Assign a professor to be the director of a study program
- Enrol a student in a study program
- Enrol a student in a course
- Enrol a student into an exam
- Assign a professor to be the teacher of a course
- Assign a person to be the host of an event
- Assign a person to attend an event
- Assign an exam to a course
- Get top 10 students
- Edit personal information of a person
- Calculate the average grade of a student
- Get timetable of a course
- Get upcoming events
- Get enrolled students in a study program
- Get available courses for a student (those for which he/she still has not passed the exam)
- Assign a location for the office of each Faculty
- Assign a location for the lecture rooms of each Study Program
- Assign a location for each held public event

OPERATION	TYPE	FREQUENCY	NOTES
1	Interactive	1470/year	total/3 (students) + 10 (professors) + 10 (worker) + 100 (guest)
2	Interactive	1 / 5 years	On average every 5 years a new faculty is added into the university
3	Interactive	1 / 3 years	On average every 3 years a new study program is added into the university
4	Interactive	645/year	Every year new courses are created because one of the identifiers is the year it gets taught
5	Interactive	50/year	
6	Interactive	1935/year	For each course 3 new exam are created (645x3)
7	Interactive	10/ year	On average every 3 year the director of a study program gets changed (30 Study Program / 3 years)
8	Interactive	1350/year	Every year about 1/3 of the students is a new student who need to enrol in a study program
9	Interactive	17700/year	Every year on average each course of the bachelor is attended by 40 students and each course of the master by 15. 375x40 (bachelor) + 180x15 (Master)
10	Interactive	41000/year	On average each student enrol in 10 exams (4100*10)
11	Interactive	1935/year	Each course has on average 3 professor (645x3)
12	Interactive	100/year	Each event has on average 2 host
13	Interactive	2500/year	Each event has on average 50 people attending

14	Batch	1935/year	Each course is assigned to 3 exams
15	Bach	1/year	When the "Dean letter" is published the top 10 students are calculated.
16	Interactive	1350/year	On average ¼ of the people change once one of their personal information ((guest + internal)/4)
17	Interactive	12300/year	After every exam session the average of each student is calculated (student x 3)
18	Bach	645/year	To publish the timetable on the website they need to be retrieve once a year for each course
19	Bach	12/year	Every month a list of upcoming events is published
20	Interactive	30/year	Every year a list of the enrolled student for each study program is published
21	Interactive	8200/year	Each semester each student gets information regarding the available courses (students x 2)
22	Bach	5/year	Each Faculty has 1 office, and it is assigned to a location
23	Bach	30/year	Each Study Program lectures rooms are in one building
24	Interactive	50/year	Each Public Event is assigned to a location

ACCESS TABLES

CONCEPT	CONSTRUCT	ACCESS	TYPE
		Operation 1	
Person	Entity	1	W
Guest/Internal	Entity	1	W
Worker/Student	Entity	1	W
Professor	Entity	0-1	W
		Operation 2	
Faculty	Entity	1	W
		Operation 3	
Study program	Entity	1	W
Faculty	Entity	1	R
Includes	Relationship	1	R
		Operation 4	
Course	Entity		W
Study program	Entity		R
Offers	Relationship		W
		Operation 5	
Public Event	Entity	1	W
Hosting	Relationship	2	W
Person	Relationship	2	R
		Operation 6	
Exam	Entity	1	W
Course	Entity	1	R
Evaluation	Relationship	1	W
		Operation 7	
Professor	Entity	1	R
Director	Relationship	1	W
Study program	Entity	1	R
		Operation 8	

Student	Entity	1	R
Study program	Entity	1	R
Enrollment	Relationship	1	W
Operation 9			
Student	Entity	1	R
Course	Entity	1	R
Attendance	Relationship	1	W
Operation 10			
Student	Entity	1	R
Exam	Entity	1	R
Taken Exam	Entity	1	W
TE by E	Relationship	1	W
TE by S	Relationship	1	W
Operation 11			
Professor	Entity	1	R
Course	Entity	1	R
Taught by	Relationship	1	W
Operation 12			
Person	Entity	1	R
Public Event	Entity	1	R
hosting	Relationship	1	W
Operation 13			
Person	Entity	1	R
Public Event	Entity	1	R
Participation	Relationship	1	W
Operation 14			
Exam	Entity	1	R
Course	Entity	1	R
Evaluation	Relationship	1	W
Operation 15			
Student	Entity	4100 (one for each student)	R
Taken Exam	Entity	4100	R
TE by S	Entity	4100	R
Operation 16			
Person	Entity	1	R/W
Data	Entity	1	W
Personal info	Relationship	1	W
Operation 17			
Student	Entity	1	R
Taken Exam	Entity	n (n = number of exams taken by the student, on average = 15)	R
TE by S	Entity	n	R
Operation 18			
Course	Entity	1	R
Operation 19			
Public Event	Entity	n (n = number of all PE existing, on average 50 per year)	R
Operation 20			
Student	Entity	n (n = number of students enrolled, on average 6 for bachelor, 50 for master, 2 for PhD)	R

Study program	Entity	n	R
Enrollment	Relationship	n	W
Operation 21			
Student	Entity	1	R
Evaluation	Relationship	x (x = number of exams taken by the student, on average = 15)	R
Exam	Entity	x	R
TE by E	Relationship	x	R
Taken Exam	Entity	x	R
TE by S	Relationship	x	R
Course	Entity	y (y = offered courses for Study Program)	R
Enrollment	Relationship	1	R
Study program	Entity	1	R
Offers	Relationship	y	R
Operation 22			
Location	Entity	1	R
Faculty	Entity	1	R
offices in	Relationship	1	W
Operation 23			
Location	Entity	1	R
Study program	Entity	1	R
lecture room in	Relationship	1	W
Operation 24			
Location	Entity	1	R
Public event	Entity	1	R
held in	Relationship	1	W

From all our tables we have excluded the relationships generated by ISA dependencies in the original schema. They all behave similarly: for each instance of the child a relationship must be written.

Phase 3 – Direct translation

Faculty (name, location)
inclusion: Faculty[name] ⊆ IncludedStudyProgram[faculty_name]
inclusion: Faculty[name] ⊆ HasOfficeIn[faculty_name]

StudyProgram (name, degree_code, number_of_students, duration, location, yearly_fee)
inclusion: StudyProgram[name] ⊆ Bachelor[name]
inclusion: StudyProgram[name] ⊆ Master[name]
inclusion: StudyProgram[name] ⊆ PhD[name]
inclusion: StudyProgram [name, degree_code] ⊆ IncludedStudyProgram[study_program_name, study_program_degree_code]
inclusion: StudyProgram[name, degree_code] ⊆ HasLectureRoomIn [study_program_name, study_program_degree_code]

Bachelor (name, degree_code, accepted_high_school_qualification)
foreign key: Bachelor[name, degree_code] ⊆ StudyProgram[name, degree_code]

Master (name, degree_code, accepted_bachelor)
foreign key: Master[name, degree_code] ⊆ StudyProgram[name, degree_code]

PhD (name, degree_code, accepted_master)
foreign key: PhD[name, degree_code] ⊆ StudyProgram[name, degree_code]

Course (name, code, year, cfu, language, timetable, semester_of_teaching, mandatory_or_not, syllabus)
inclusion: Course[name, code, year] ⊆ TaughtCourses [course_name, course_code, course_year]
inclusion: Course[code, year] ⊆ EvaluatedCourses [course_code, course_year]

Exam (date_time, course_name, room)
foreign key: Exam[course_name] ⊆ Course[name]
inclusion: Exam[course_name] ⊆ TakenExam[course]
foreign key: Exam[date_time] ⊆ EvaluatedCourses [exam_data_time]
key: date_time

Person (name, lastname, phone, personal_email)
inclusion: Person[name, lastname] ⊆ Guest[name, lastname]
inclusion: Person[name, lastname] ⊆ Internal[name, lastname]

Data (phone, personal_email)
inclusion: Data[phone, personal_email] ⊆ PersonalInformation[data_phone, data_personal_email]

Guest (name, lastname, guest_id)
foreign key: Guest[name, lastname] ⊆ Person[name, lastname]
key: guest_id

Internal (name, lastname, phone, personal_email, academic_email)
foreign key: Internal[name, lastname] ⊆ Person[name, lastname]
key: academic_email

Student (student_id, name, lastname, phone, personal_email, academic_email, degree*)
foreign key: Student[name, lastname] ⊆ Person[name, lastname]
foreign key: Student[academic_email] ⊆ Internal[academic_email]
includes:
key: student_id, academic_email
constrain: degree is NULL

Worker (name, lastname, phone, personal_email, academic_email, department, begin_of_contract, end_of_contract)
foreign key: Worker[name, lastname] ⊆ Person[name, lastname]
foreign key: Worker[academic_email] ⊆ Internal[academic_email]

Professor (professor_id, name, lastname, phone, personal_email, academic_email, department, begin_of_contract, end_of_contract, office_hours)
foreign key: Professor[name, lastname] ⊆ Person[name, lastname]
foreign key: Professor[academic_email] ⊆ Internal[academic_email]
inclusion: Professor[professor_id, name, lastname] ⊆ TaughtBy[professor_id, professor_name, professor_lastname]
key: professor_id

Location (city, street, number)

PublicEvent (name, date_time, location)
inclusion: PublicEvent [name, date_time] ⊆ HostingEvent[event_name, event_date_time]
inclusion: Event[name, date_time] ⊆ HasOfficesIn [event_name, event_date_time]

PartecipationInEvent (person_name, person_lastname, event_name, event_date_time)
foreign key: PartecipationInEvent[person_name, person_lastname] ⊆ Person[name, lastname]
foreign key: PartecipationInEvent[event_name, event_date_time] ⊆ Event[name, date_time]

HostingEvent (person_name, person_lastname, event_name, event_date_time)
foreign key: PartecipationInEvent[person_name, person_lastname] ⊆ Person[name, lastname]
foreign key: PartecipationInEvent[event_name, event_date_time] ⊆ Event[name, date_time]

TaughtCourses (professor_id, professor_name, professor_lastname, course_name, course_code, course_year)
foreign key: TaughtCourses[professor_name, professor_lastname] ⊆ Professor[name, lastname]
foreign key: TaughtCourses [course_name, course_code, course_year] ⊆ Course[name, code, year]

TakenExams (student_id, student_name, student_lastname, grade, status, exam_data_time, course_name, course_code, course_year)
foreign key: TakenExam[student_id, student_name, student_lastname] ⊆ Student[student_id, name, lastname]
foreign key: TakenExam[exam_date_time,] ⊆ Exam[date_time]
foreign key: TakenExam[course_name, course_code, course_year] ⊆ Course[name, code, year]

StudentsEnrolled (student_id, student_name, student_lastname, year_of_enrollment, study_program_name, study_program_degree_code)
foreign key: StudentsEnrolled [student_id, student_name, student_lastname] ⊆ Student[student_id, name, lastname]
foreign key: StudentsEnrolled [study_program_name, study_program_degree_code] ⊆ StudyProgram[name, degree_code]

OfferedCourses (course_name, course_code, course_year, study_program_name, study_program_degree_code,)
foreign key: OfferedCourses[course_name, course_code, course_year] ⊆ Course[name, code, year]
foreign key: OfferedCourses[study_program_name, study_program_degree_code] ⊆ StudyProgram[name, degree_code]

EvaluatedCourses (course_name, course_code, course_year, exam_data_time)
foreign key: EvaluatedCourses [course_code, course_year] ⊆ Course[code, year]
foreign key: EvaluatedCourses [exam_data_time] ⊆ Exam[date_time]
key: name, room

IncludedStudyProgram (faculty_name, study_program_name, study_program_degree_code)
foreign key: IncludedStudyProgram [faculty_name] ⊆ Faculty[name]
foreign key: IncludedStudyProgram [study_program_name, study_program_degree_code] ⊆ StudyProgram[name, degree_code]

Directors (study_program_name, study_program_degree_code, professor_id, begin_year_of_direction)
foreign key: Director[study_program_name, study_program_degree_code] ⊆ StudyProgram[name, degree_code]
foreign key: Director[professor_id] ⊆ Professor[professor_id]

PersonalInformation (person_name, person_lastname, data_phone, data_personal_email)
foreign key: PersonalInformation [name, lastname] ⊆ Person[name, lastname]
foreign key: PersonalInformation [data_phone, data_personal_email] ⊆ Data[phone, personal_email]

StudentAttendence (student_id, course_code, course_year)
foreign key: StudentAttendence [student_id] ⊆ Student[student_id]
foreign key: StudentAttendence [course_code, course_year] ⊆ Course[code, year]

HasOfficesIn (faculty_name, location_city, location_street, location_number)
foreign key: HasOfficesIn [faculty_name] ⊆ Faculty[name]
foreign key: HasOfficesIn [location_city, location_street, location_number] ⊆ Location[name, city, number]

HasLectureRoomIn (study_program_name, study_program_degree_code, location_city, location_street, location_number)
foreign key: HasLectureRoomIn [study_program_name, study_program_degree_code] ⊆ StudyProgram[name, degree_code]
foreign key: HasLectureRoomIn [location_city, location_street, location_number] ⊆ Location[name, city, number]

HeldIn (event_name, event_date_time, location_street, location_number)
foreign key: HasOfficesIn [event_name, event_date_time] ⊆ Event[name, date_time]
foreign key: HasOfficesIn [location_city, location_street, location_number] ⊆ Location[name, city, number]

All relationship deriving from ISA dependencies behave similarly:
ISARelation (parent_identifier, child_identifier)
foreign key: Parent[parent_attribute] ⊆ Child[child_attribute]
key: childEntity

External Constraints:

- Constraints that relates Student[student_id] to the instances of StudyProgram
- Constraints that relates Student[student_id] to the instances of Exam
- Constraints that relates Exam[course_name] to the instances of Course
- Constrains that relates PartecipationInEvent[name,lastname] to the instances of Person
- Constrains that relates Internal[academic_email] to the instances of Worker and Student
- Constrains that relates Person[name, lastname] to the instances of Internal and Guest
- Constrains that relates StrudyProgram[name, degree_code] to the instances of Bachelor, Master, PhD

Phase 4 – Restructuring of the relational schema

Database schema R: {

Faculty(name, location),
StudyProgram(name, degree_code, number_of_students, duration, location, yearly_fee),
Bachelor(name, degree_code, accepted_high_school_qualification),
Master(name, degree_code, accepted_bachelor),
PhD(name, degree_code, accepted_master),
Course(name, code, year, cfu, language, timetable, semester_of_teaching, mandatory_or_not, syllabus),
Exam(date_time, course_name, room),
Person(name, lastname, phone, personal_email),
Data (phone, personal_email),
Guest (name, lastname, guest_id),
Internal (name, lastname, phone, personal_email, academic_email),
Student (student_id, name, lastname, phone, personal_email, academic_email, degree),
Worker (name, lastname, phone, personal_email, academic_email, department, begin_of_contract, end_of_contract),
Professor (professor_id, name, lastname, phone, personal_email, academic_email, department, begin_of_contract, end_of_contract, office_hours),
Location (city, street, number),
PublicEvent(name, date_time, location)

}

We transformed the Study Program generalization with Bachelor, Master and PhD into ISA-relationship to reduce the number of accesses. We similarly did with Internal with Worker and Student, and with Person with Internal and Guest. By transforming the ISA dependencies into relationship, we had to include some external constraints such that the father entity is always an instance of one of the child entities.

We also transformed the TakenBy relationship between Exam and Student into an Entity in relation with the other entity. This was also made to reduce the number of access because by having a TakenByStudent relationship it is sufficient to access that to know the status of a student's exam.

We transformed the attribute location into an entity in relation with all those entities which had it as attribute in order to simplify the schema. Another attribute that we simplified is the personal information attribute, since it was a multi-value attribute, we transformed it into an entity with attributes.

Phase 5 – Specification of the database in SQL

See tables specification in the file createTables.sql and populateTables.sql .