

The DB Conceptual Design Document (CDD)

SHARAT CHANDRA REDDY TANGELLA

VENKAT KOUSHIK MUTHYAPU

March 2019

Indice

| | | |
|------|--|----|
| 1. | The Miniworld (or Universe of Discourse) | 3 |
| 2. | Conceptual Schema of the Database | 3 |
| 2.1. | Notation and definitions..... | 3 |
| 2.2. | Entities | 4 |
| 2.3. | Relationships..... | 7 |
| 2.4. | Explicit Integrity Constraints | 10 |
| 3. | Example Queries | 10 |
| 3.1. | Possible extensions and additional comments | 11 |
| 4. | Mapping the database to relational model | 11 |
| 4.1. | Additional Integrity Constraints for the relational schema | 13 |

1. The Miniworld (or Universe of Discourse)

The mini-world taken for the project is the information about students that will act as a good predictor for student success in its program of study. The database holds all the information about the students and the computer science department such as the student's program of study, courses taken by the student, advisor information, activities in which the student participated and so on. This information will help to improve the recruitment and retention of student in the program offered by the department. A student can be either a pre-major or a major of computer science. The database also gives information about the courses that need to be completed in order to be a pre-major or major in CS.

This database helps to keep track of the students who are pre-majors, transitioning from pre-major to major and students who majored in CS and make changes in order to increase recruitment and retention.

2. Conceptual Schema of the Database

The order of presentation of the conceptual schema is:

1. Entities: description and attributes.
2. Relationships: description and attributes (if they have them).
3. EER diagram
4. Explicit Integrity Constraints

An order must be chosen to present the entities and relationships. One option is to present them alphabetically; another is to start with the most important entity and continue with the entities related to it, the relationship

This document can also be considered a manual *data dictionary*. Each DBMS has mechanisms for defining and querying the data dictionary used by it to access the database and that would be the computerized data dictionary.

2.1. Notation and definitions

The notation used: all upper case for the entity names, lower case for the relationship names, and the first letter capitalized for attribute names.

The description of the entities starts with a sentence which explains their meaning. Then the attributes to describe the instances are included. The relationships are also described by a sentence and a list of attributes if it has them.

Each attribute has a four-letter code which describes the type of attribute according to the four classification criteria for attributes.

The format for this code is: (xyzw), where x tells that the attribute is

simple (S) or composite (C), y tells that the attribute has a single value

(S) or is multivalued (M),

z tells that the attribute is primitive (stored) (P) or derived (D), in case it is derived, an explanation of how to deduce it from other attributes or a formula must be specified, and w tells that the attribute is fixed (F) (i. e. it must have a value that is not null) or optional (O).

For example, an attribute that has the *SSPF* code is a simple attribute with a single value which is primitive and fixed. An example of this kind of attribute could be the Social Security Number (SSN). On the other hand, an attribute with the (*CSPO*) code is a composite attribute with a single value, primitive and optional. In this case, the date of birth could be an attribute with this code. If there is a single attribute that has the key constraint, it can be underlined. If the key constraint applies to more than one attribute or if there are several combinations of attributes with the key constraint property it is better to list them separately.

If there are attributes that are very common and are used more than once, they can be defined as general types to be used as the type of each attribute which uses the same format.

2.2. Entities

The entities defined for this database are:

- STUDENT
- EMPLOYEES
- UNDERGRAD
- MASTERS
- PHD
- ALL COURSES
- PROGRAM OF STUDY
- TUTOR
- RESEARCH
- SPECIAL ACTIVITIES
- ADVISOR
- PROFESSOR
- INTERNSHIPS
- SURVEYS
- CLUBS
- RESUMES
- SCHOLARSHIPS

- AWARDS

A detailed description of each entity follows.

STUDENT: The person who enrolls in the university at a certain department to pursue a field of study.

| | | |
|-------------|----------------------|--------|
| Attributes: | s_id | (SSPF) |
| | s_name | (CSPF) |
| | s_phn_num | (CMPO) |
| | s_univ_email | (SSPF) |
| | s_dob | (CSPF) |
| | s_start_date | (CSPF) |
| | s_end_date | (CSPO) |
| | s_pre-majors | (SMPO) |
| | s_majors | (SMPO) |
| | s_minors | (SMPO) |
| | s_currently_pursuing | (SSPF) |
| | s_credits_earned | (SSDF) |
| | s_current_gpa | (SSDF) |

EMPLOYEES: These are the students who work while actively pursuing their degree. They work for an organisation.

| | | |
|-------------|----------------------|--------|
| Attributes: | work_organization | (SSPF) |
| | no_of_hours_per_week | (SSPF) |
| | position | (SSPF) |

UNDERGRAD: These are the students who are pursuing their undergrad degree.

MASTERS: These are the students who are pursuing their master's degree.

PHD: These are the students who are pursuing their Ph. D.

ALL COURSES: These are the subjects taken by the student in order to complete a field of study. A student can take any number of course.

| | | |
|-------------|---------------|--------|
| Attributes: | course_number | (SSPF) |
| | Course_name | (SSPF) |
| | Credit_hours | (SSPF) |
| | department | (SSPF) |
| | semester | (SMPF) |

PROGRAM OF STUDY: This is the program of study that a student opts to complete.

| | | |
|-------------|-------------------|--------|
| Attributes: | study_program | (SSPF) |
| | total_grade_req | (SSPF) |
| | total_credits_req | (SSPF) |

TUTOR: They are the people who help the students with the subjects they are having difficulty with. They are also students working as part time tutors.

| | | |
|-------------|--------------|--------|
| Attributes: | tutor_id | (SSPF) |
| | t_start_sem | (SSPF) |
| | t_start_year | (SSPF) |
| | t_rating | (SSDF) |

ADMINISTRATIVE ADVISOR: He is the person who help students with all the administrative procedures related to their life in a program of study.

| | | |
|-------------|----------------|--------|
| Attributes: | adv_id | (SSPF) |
| | adv_name | (CSPF) |
| | adv_univ_email | (SSPO) |

PROFESSOR: He is the person who teaches courses and acts as academic advisor to the students and meets with the student each semester to advise them on which courses to register for.

| | | |
|-------------|-----------|--------|
| Attributes: | pro_name | (CSPF) |
| | Pro_email | (SSPF) |

RESEARCH: This is the field of research conducted by the student under the supervision of a professor.

| | | |
|-------------|---------------------|--------|
| Attributes: | research_name | (SSPF) |
| | research_start_date | (CSPF) |
| | grants_alloted | (SSPO) |
| | grants_alloted_by | (SSPO) |

SPECIAL ACTIVITIES: These are the other activities that a student participates such as technical events which are organized by the department.

| | | |
|-------------|----------------------|--------|
| Attributes: | activity_name | (SSPF) |
| | Organization_holding | (SSPF) |
| | act_date | (CSPF) |

INTERNSHIPS: The internships that a student does during summer for full time.

| | | |
|-------------|-----------------|--------|
| Attributes: | intern_org | (SSPF) |
| | Intern_position | (SSPF) |
| | Intern_pay | (SSPF) |

SURVEYS: The surveys are taken from students at different phases that help evaluate the student.

| | | |
|-------------|-------------|--------|
| Attributes: | survey_type | (SSPF) |
|-------------|-------------|--------|

CLUBS: A student may be part of a club activities such as music, dance or any sports.

| | | |
|-------------|-----------------|--------|
| Attributes: | club_name | (SSPF) |
| | Club_start_date | (CSPF) |
| | Club_head | (SSPF) |

RESUMES: This is the resume of a student. It helps in the employment of the student.

| | | |
|-------------|---------------|--------|
| Attributes: | resume_id | (SSPF) |
| | resume_name | (SSPF) |
| | last_modified | (CSPF) |

AWARDS: These are the awards given to the students based upon reviewing the resume of the students.

| | | |
|-------------|------------|--------|
| Attributes: | award_name | (SSPF) |
|-------------|------------|--------|

SCHOLARSHIPS: These are the scholarships awarded to a student.

| | | |
|-------------|------------------------|--------|
| Attributes: | scholarship_name | (SSPF) |
| | Scholarship_recipients | (SMPF) |
| | Stipend | (SSPF) |

2.3. Relationships

The relationships in this schema are described below.

advices: This is the relation between student and professor. The professor advices the student regarding the courses to take. Each student has a single professor to advise.

No attributes.

takes_part: The student may take part in any research that is being done by a professor.

| | | |
|-------------|-----------|--------|
| Attributes: | join_date | (CSPF) |
|-------------|-----------|--------|

does: This is the relation between professor and the research. A professor can have more than one research area. Students are included in the research.

Consults: This relation describes that a student consults a tutor if he has any doubts or difficulties in one or more than one subject.

| | | |
|-------------|-------------|--------|
| Attributes: | rates_tutor | (SSPF) |
|-------------|-------------|--------|

Helps_in: The tutor gives tutoring to students in the courses. One tutor can help with one or more than one course.

| | | |
|-------------|----------|--------|
| Attributes: | semester | (SSPF) |
| | Year | (SSPF) |

takes: This is the relation between students and courses. Students take many different courses to complete the field of study.

| | | |
|-------------|----------------|--------|
| Attributes: | grade_obtained | (SSPO) |
| | taken_sem | (SSPF) |
| | taken_year | (SSPF) |

pre-requisite: This is the relation between the courses and their pre-requisites. A course can have more than one pre-requisites.

parts_in: This is the relation between the program of study and the courses. For each program of study, there are certain courses to complete.

| | | |
|-------------|-----------|--------|
| Attributes: | min_grade | (SSPF) |
|-------------|-----------|--------|

opts_for: This is field of study that a student is currently opt to pursue. It can be pre-major, minor, major, masters or phd.

| | | |
|-------------|----------|--------|
| Attributes: | opt_sem | (SSPF) |
| | opt_year | (SSPF) |

teaches: The professors teach the courses. One professor can teach more than one course.

| | | |
|-------------|------------|--------|
| Attributes: | teach_sem | (SSPF) |
| | teach_year | (SSPF) |

works_in: The students works in the internship during the summer. One student can work in any number of assistantships.

| | | |
|-------------|-------------------|--------|
| Attributes: | intern_start_date | (CSPF) |
| | Intern_end_date | (CSPF) |

guides: The administrative advisor guides with all the administrative procedures related to their life in a program of study.

fills_out: The students fills out the surveys at each point of time. This helps to evaluate the student and get the views of the students to improvise the recruitment and retention of department.

| | | |
|-------------|-----------|--------|
| Attributes: | fill_date | (CSPF) |
|-------------|-----------|--------|

prepares: Each student has a resume which has the academic information about the student.

given_by: This is the relation between resume and rewards. Rewards are given by observing the resume of the student. Each reward is given to only one student for that particular semester.

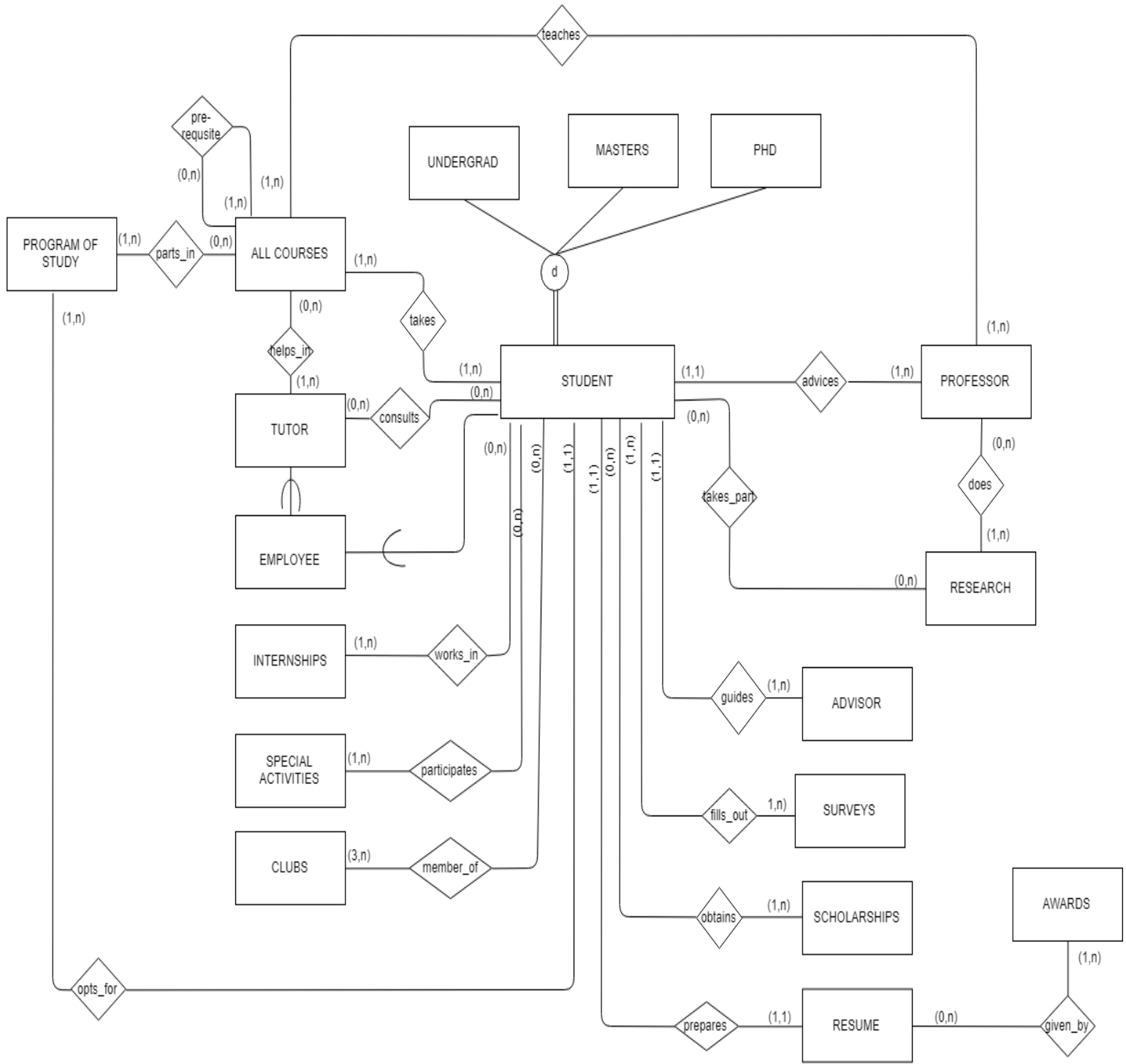
| | | |
|-------------|------------|--------|
| Attributes: | Given_sem | (SSPF) |
| | Given_year | (SSPF) |

participates: The student participates in the special activities. The student can participate in any number of activities.

obtains: The student can obtain scholarship given by the college or any other external organization. A student can get more than one scholarship or none.

Attributes: obtain_sem
 Obtain_year

member_of: This is the relation between student and the club he is the member of.



Formulas for derived attributes

Credits earned:

$$\sum_{k=0}^n \text{credits taken per subject}$$

Total GPA

$$(\sum_{k=0}^n \text{grade earned per subject}) / \text{number of courses taken}$$

Tutor rating

$$(\sum_{k=0}^n \text{rating given by each student}) / \text{total number of ratings given}$$

2.4. Explicit Integrity Constraints

Some examples of integrity constraints in our working example of a company database.

1. Start_date < End_date in internships.
2. Student date of birth should be less than graduation date.
3. A student cannot take a course more than 4 times.
4. A student employee must not work for more than 20 hours per week.

3. Example Queries

A list of the most important queries

1. List all the students who are pursuing their masters.
2. List the courses taken by a student in a particular semester.
3. Name all the courses that are tutored by a particular tutor.
4. List all the clubs that a student is member of.
5. List all the students that a professor is the advisor for.
6. List the scholarships that a student has.

3.1. Possible extensions and additional comments

There may be other factors such as locality of the university and the fee structure for a particular program of study that affect the choice of the student to choose a university in order to pursue a field of study.

4. Mapping the database to relational model

The conceptual schema described for the Database is mapped into the Relational Schema presented in this section. Primary keys are underlined.

All the attributes underlined in the same Relation belong to the primary key.

STUDENT(S_id,S_fname,S_mname,S_lname,S_univ_email,S_dob,S_start_date,S_end_date,S_currently pursuing,S_credits earned,S_current_gpa,Adv_id,Pro_email)

Adv_id is foreign key, references ADVISOR

Pro_email is foreign key, references PROFESSOR

S_currently_pursuing is foreign key, references PROGRAM_OF_STUDY

SPhone(S_id,AreaCode,Phone num)

S_id is foreign key, references STUDENT

SPreMajors(S_id,PreMajor)

S_id is foreign key, references STUDENT

SMajor(S_id,Major)

S_id is foreign key, references STUDENT

SMinor(S_id,Minor)

S_id is foreign key, references STUDENT

ALL_COURSES(Course number,Course_name,Credit_hours,Department)

CourseSem(Course number,Semester)

Course_number is foreign key, references ALL_COURSES

PROGRAM_OF_STUDY(Study program,Total_grade_req,Total_credits_req)

EMPLOYEES(S_id,Work organization,No_of_hours_per_week,Position)

S_id is foreign key, references STUDENT

TUTOR(Tutor_id,T_start_sem,T_start_year,T_rating)

Course_id is foreign key, references ALL_COURSES

PROFESSOR(Pro_fname,Pro_mname,Pro_lname,Pro_email)

RESEARCH(Research_name,Research_start_date,Grants_alloted,Grants_alloted_by)

ADVSOR(Adv_id,Adv_fname,Adv_mname,Adv_lname,Adv_email)

CLUBS(Club_name,Club_start_start_date,Club_head_id)

Club_head_id is foreign key, references STUDENT

RESUME(S_id,Resume_id,Resume_name,Last_modified)

S_id is foreign key, references STUDENT

SURVEY(Survey_type)

SCHOLARSHIPS(Scholarship_name,Stipend)

Scholarship_Recipients(Scholarship_name,Recipients)

Scholarship_name is foreign key, references SCHOLARSHIPS

Works_in(S_id,Intern_org,Intern_position,Intern_pay,Intern_start_date,Intern_end_date)

S_id is foreign key, references STUDENT

Participates(S_id,Activity_name,Act_date,Organization_holding)

S_id is foreign key, references STUDENT

Takes(S_id,Course_number,Taken_sem,Taken_year,Grade_obtained)

S_id is foreign key, references STUDENT

Course_number is foreign key, references ALL_COURSES

Teaches(Course_number,Pro_email,Teach_sem,Teach_year)

Course_number is foreign key, references ALL_COURSES

Pro_email is foreign key, references PROFESSOR

Does(Research_name,Pro_email)

Research_name is foreign key, references RESEARCH

Pro_email is foreign key, references PROFESSOR

Takes_part(S_id,Research_name,Join_date)

S_id is foreign key, references STUDENT

Research_name is foreign key, references RESEARCH

Pre-requisites(Course_number,Pre-req_number)

Course_number is foreign key, references ALL_COURSES

Pre-req_number is foreign key, references ALL_COURSES

Consults(S_id,Tutor_id,Course_id,Rates_tutor)

S_id is foreign key, references STUDENT

Tutor_id is foreign key, references TUTOR

Course_number is foreign key, references ALL_COURSES

Helps_with(Tutor_id,Course_id,semester,year)

Parts_in(Study_program,Course_number,Min_grade)

Study_program is foreign key, references PROGRAM_OF_STUDY

Course_number is foreign key, references ALL_COURSES

Member_of(S_id,Club_name)

S_id is foreign key, references STUDENT

Club_name is foreign key, references CLUBS

Fills_out(S_id,Survey_type,Fill_date)

S_id is foreign key, references STUDENT

Survey_type is foreign key, references SURVEY

Obtains(S_id,Scholarship_name,Obtain_sem,Obtain_year)

S_id is foreign key, references STUDENT

Scholarship_name is foreign key, references SCHOLARSHIPS

Given_by(Resume_id,Award_name,Given_sem,Given_year)

Resume_id is foreign key, references RESUME

4.1. Additional Integrity Constraints for the relational schema

The integrity constraints that must hold for the database and that are not guaranteed by the relation schemas described above are listed in this subsection.

1. Not all students appear in the relation SPhone.
2. The attribute Adv_id cannot be null in STUDENTS.
3. Every value of Adv_id in PROFESSOR has to appear in atleast one tuple f relation STUDENTS in the Adv_id attribute.
4. EndDate > StartDate. This restriction holds for every pair of attributes with this semantics.
5. For each tuple in Pre-requisite the value of course_number must be different from the value of Pre-req_number.

