**University Database**

Group:

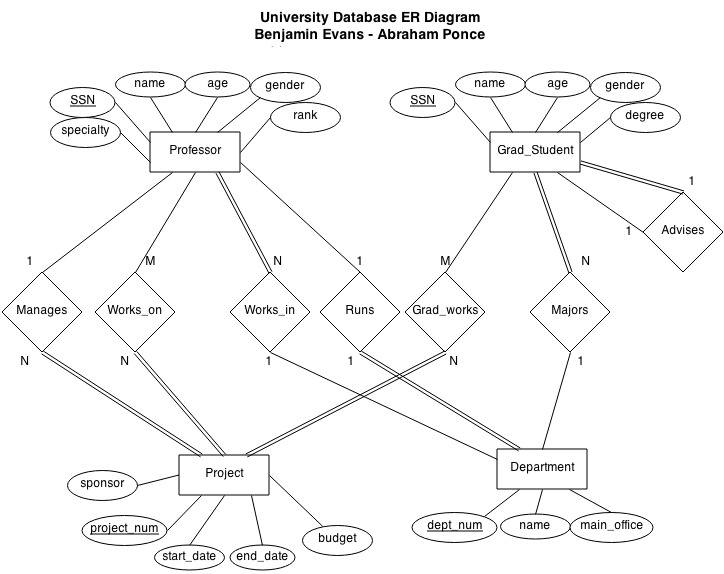
Benjamin Evans, Abraham Ponce

**Project Description**

Design and implement a university database from a set of requirements.

**Design**

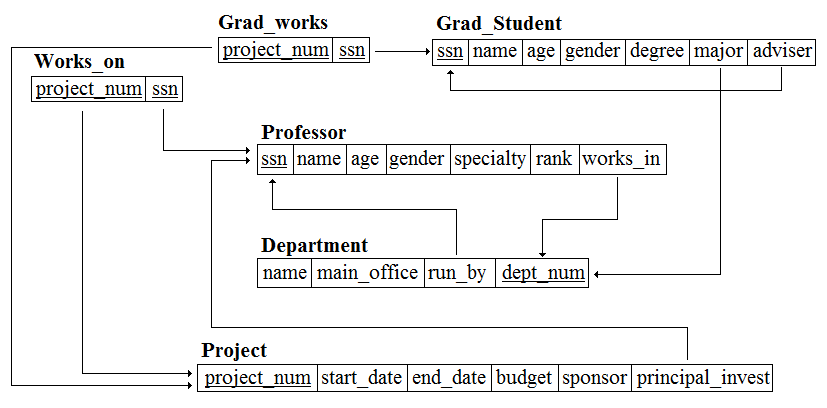
First we created an ER Diagram.



(In the diagram we assume that a professor can run more than one department and that student advisers only advise one graduate student since neither was specified in the requirements)

**Tables**

We then translated the diagram into tables and created SQL statements for them.



drop table if exists Grad\_Student Cascade;

drop table if exists Project Cascade;

drop table if exists Professor Cascade;

drop table if exists Department Cascade;

drop table if exists Grad\_works Cascade;

drop table if exists Works\_on Cascade;

CREATE TABLE Grad\_Student(

ssn Integer primary key,

name varchar(32) Not Null,

age Integer Not Null,

gender char(1) Not Null,

degree varchar(32) Not Null,

adviser Integer Null Unique,

major Integer Not Null

);

INSERT INTO Grad\_Student values(111, 'Seth', 21, 'M', 'M.S.', null, 100);

INSERT INTO Grad\_Student values(222, 'Beth', 21, 'M', 'Ph.D', null, 300);

INSERT INTO Grad\_Student values(333, 'Tom', 21, 'M', 'Ph.D', 222, 300);

INSERT INTO Grad\_Student values(444, 'Pam', 21, 'M', 'M.S.', 111, 100);

INSERT INTO Grad\_Student values(555, 'Mike', 21, 'M', 'M.S.', null, 100);

INSERT INTO Grad\_Student values(666, 'Jann', 21, 'M', 'Ph.D', null, 200);

INSERT INTO Grad\_Student values(777, 'Jim', 21, 'M', 'M.S.', 555, 100);

INSERT INTO Grad\_Student values(888, 'Tim', 21, 'M', 'Ph.D', 666, 200);

CREATE TABLE Project(

project\_num Integer Primary key,

start\_date Date Not Null,

end\_date Date Not Null,

budget Integer Not Null,

sponsor varchar(32) Not Null,

principal\_invest integer Not NULL

);

INSERT INTO Project values(10, '2014-01-25', '2014-04-25', 3000, 'NASA', 123);

INSERT INTO Project values(20, '2014-05-01', '2014-06-01', 4000, 'CCP', 456);

INSERT INTO Project values(30, '2014-10-30', '2014-11-30', 5000, 'NSA', 654);

INSERT INTO Project values(40, '2013-09-30', '2013-12-15', 6000, 'SpaceX', 321);

CREATE TABLE Professor(

ssn Integer Primary key,

name varchar(32) Not Null,

specialty varchar(32),

rank varchar(32),

age Integer Not Null,

gender char(1) Not Null,

works\_in Integer Not Null

);

INSERT INTO Professor values(123, 'Bill', 'A.I.', 'Rank 1', 55, 'M', 100);

INSERT INTO Professor values(321, 'Jill', 'Database', 'Rank 2', 57, 'F', 100);

INSERT INTO Professor values(789, 'Paul', 'Compilers', 'Rank 2', 42, 'M', 100);

INSERT INTO Professor values(456, 'Kate', 'English', 'Rank 1', 51, 'F', 200);

INSERT INTO Professor values(890, 'Jeff', 'English', 'Rank 2', 62, 'M', 200);

INSERT INTO Professor values(654, 'Lisa', 'History', 'Rank 1', 45, 'F', 300);

INSERT INTO Professor values(198, 'John', 'History', 'Rank 2', 39, 'M', 300);

INSERT INTO professor values(190, 'Juan', 'Embedded', 'Rank 2', 59, 'M', 100);

INSERT INTO professor values(291, 'Jane', 'Algorithms', 'Rank 3', 44, 'F', 100);

INSERT INTO professor values(374, 'Lane', 'History', 'Rank 3', 37, 'M', 200);

INSERT INTO professor values(697, 'Anne', 'English', 'Rank 2', 63, 'F', 200);

INSERT INTO professor values(156, 'Dave', 'History', 'Rank 2', 41, 'M', 300);

INSERT INTO professor values(784, 'Katy', 'English', 'Rank 3', 33, 'F', 300);

CREATE TABLE Department(

dept\_num Integer primary key,

name varchar(32) Not Null,

main\_office varchar(32) Not Null,

run\_by integer Not Null

);

INSERT INTO Department values(100, 'Computer Science', 'Office 1', 123);

INSERT INTO Department values(200, 'History', 'Office 2', 654);

INSERT INTO Department values(300, 'English', 'Office 3', 456);

CREATE TABLE Grad\_works(

project\_num Integer Not Null,

ssn Integer Not Null,

Primary Key(project\_num,ssn)

);

CREATE TABLE Works\_on(

project\_num Integer Not Null,

ssn Integer Not Null,

Primary Key(project\_num,ssn)

);

INSERT INTO Grad\_works values(10,111);

INSERT INTO Works\_on values(10,123);

INSERT INTO Grad\_works values(20,222);

INSERT INTO Works\_on values(20,456);

INSERT INTO Grad\_works values(30,666);

INSERT INTO Works\_on values(30,654);

INSERT INTO Grad\_works values(40,555);

INSERT INTO Works\_on values(40,321);

-- Adding foreign keys

ALTER TABLE Grad\_Student

ADD Foreign Key (major) References Department(dept\_num) On Delete Restrict ON Update Cascade;

ALTER TABLE Grad\_Student

ADD Foreign Key (adviser) References Grad\_Student(ssn) ON Delete Restrict ON Update Cascade;

ALTER TABLE Project

ADD Foreign Key (principal\_invest) References Professor(ssn) ON Delete Restrict On Update Cascade;

ALTER TABLE Professor

ADD Foreign Key (works\_in) References Department(dept\_num) ON Delete Restrict ON Update Cascade;

ALTER TABLE Department

ADD Foreign Key (run\_by) References Professor(ssn) ON Delete Restrict On Update Cascade;

ALTER TABLE Grad\_works

ADD Foreign Key (project\_num) References Project(project\_num) ON Delete Cascade On Update Cascade;

ALTER TABLE Grad\_works

ADD Foreign Key (ssn) References Grad\_Student(ssn) ON Delete Cascade On Update Cascade;

ALTER TABLE Works\_on

ADD Foreign Key (project\_num) References Project(project\_num) ON Delete Cascade On Update Cascade;

ALTER TABLE Works\_on

ADD Foreign Key (ssn) References Professor(ssn) ON Delete Cascade On Update Cascade;

**Stored Procedures and Triggers**

Next we created two stored procedures and two triggers.

CREATE FUNCTION female\_faculty() RETURNS real AS'

DECLARE

count REAL := 0;

total REAL := 0;

BEGIN

SELECT COUNT(gender) INTO count FROM professor WHERE gender=''F'';

SELECT COUNT(\*) INTO total FROM professor;

RETURN (count/total) \* 100;

END;

' LANGUAGE 'plpgsql';

CREATE OR REPLACE FUNCTION total\_people(Integer) RETURNS Integer AS '

DECLARE

project\_id ALIAS FOR $1;

total\_grad Integer := 0;

total\_prof Integer := 0;

PI Integer := 0;

total Integer := 0;

BEGIN

SELECT count(project\_num) INTO total\_grad FROM Grad\_works

WHERE project\_num = project\_id;

SELECT count(project\_num) INTO total\_prof FROM Works\_on

WHERE project\_num = project\_id;

total := total\_grad + total\_prof;

SELECT count(project\_num) INTO PI FROM Works\_on

WHERE project\_num = project\_id and ssn=(select ssn from Project where project\_num = project\_id);

IF PI = 0

THEN

total := total+1;

END IF;

RETURN total;

END;

' LANGUAGE 'plpgsql';

CREATE FUNCTION faculty\_restrict() RETURNS trigger AS'

DECLARE

ssncheck INTEGER;

count INTEGER := 0;

row\_data works\_on%ROWTYPE;

BEGIN

FOR row\_data IN SELECT \* FROM works\_on WHERE project\_num = NEW.project\_num

LOOP

SELECT ssn into ssncheck FROM professor WHERE SSN=row\_data.SSN;

IF FOUND THEN

count := count + 1;

END IF;

END LOOP;

RAISE NOTICE ''Professors working on project is %'', count;

IF count < 4 THEN

IF TG\_OP = ''UPDATE''

THEN UPDATE works\_on SET project\_num = NEW.project\_num WHERE ssn=NEW.ssn;

END IF;

ELSE

RAISE EXCEPTION ''Too many professors working on that project.'';

END IF;

RETURN NEW;

END;

' LANGUAGE 'plpgsql';

CREATE TRIGGER faculty\_restrict

BEFORE INSERT OR UPDATE ON works\_on

FOR EACH ROW EXECUTE PROCEDURE faculty\_restrict();

CREATE OR REPLACE FUNCTION student\_restrict() RETURNS trigger AS'

DECLARE

total INTEGER := 0;

BEGIN

SELECT count(project\_num) INTO total FROM Grad\_works

WHERE ssn = NEW.ssn;

IF total > 2 THEN

RAISE EXCEPTION ''Student cannot work on more than 2 projects'';

END IF;

RETURN NEW;

END;

' LANGUAGE 'plpgsql';

CREATE TRIGGER student\_restrict

AFTER INSERT ON Grad\_works

FOR EACH ROW EXECUTE PROCEDURE student\_restrict();

**Member Contribution**

We split the work as best we could. Abraham created the ER diagram and Benjamin helped revise it. Benjamin created the tables and the SQL statements and Abraham helped revise it.

We each created one procedure and one trigger.

We collaborated on the JDBC part of the project.

We believe the contribution to be a 50/50 split between the two group members.

**User Manual**

1. Download the DBPanel.java and JDBCProject.java from CSNS
2. Open your Eclipse Java IDE.
3. Create a new project.
4. Place the DBPanel.java and JDBCProject.java into the src folder
5. Right click the project, hover over build path and select add external archives
6. Navigate to where the postgresql jdbc jar file is located on your local machine and select it and press ok
7. In the package explorer right click the JDBCProject.java file hover over run as and select java application