UTEP
Student Employment Application Database

TEAM 9
LUIS MEDINA – ERICK AVALOS – GERMAN VIEZCAS – KEVIN OBREGON

Assignment 3-v1.0

Table of Contents

Table of Figures

SCOPE

The purpose of the system is to allow Students to fill out applications in order to be considered for paid positions at UTEP. The Student position types include IA, TA, and Peer Leader. The three users will be Students (Applicants), Coordinators and Administrators. Every user type will have an account, each with different privileges. The Administrator is the only user who shall create positions and offer a position to the Student. The Administrator and Coordinators will be able to see the Student's current courses, create and read reports, and to review an application. Positions will include what semester they are for, the type of position it is for, the status of the position and a job ID. All users will be greeted by a UTEP-themed welcome page asking for login credentials or an option to create a new account. Every new user account will need their first name, last name, email, password and identifying username. As part of the new account, Students shall fill out additional sections with information such as GPA, classification, gender, and date of birth. Student's courses will include the course name, semester, hours, and CRN number. Once logged in, Students shall fill out a new application or have the option to see the status of pending applications. The webpage for a new application will allow the student to attach an optional resume and personal statemen. If Students want to see the status of pending applications, they will see a list of submitted application numbers and the status of the applications, which include under review or approved. Once an application has been accepted for a position, the Student will have the ability to read, print, and sign their contract for employment. When a Coordinator logs in they will see a list of Student's first and last names who have applied along with their email, GPA, classification, gender and application status.

REQUIREMENTS

Requirement	Satisfied By	Implemented In
1. The system requires a host server.	Computer Science Department @ UTEP.	The system is to be hosted on a server of the Computer Science Department @ UTEP.
2. The system will require new Students to create a new account using their first name, last name, email, password, username, grade level, DOB, GPA, class, and gender.	The webpage to create a new account.	The website provides a section for students to submit the *Profile* and stores it on the database thorough SQL Queries.
3. The system requires one admin and several coordinators with special privileges to oversee applications.	Pre-applied users to the database that will have special privileges on the website.	The database will have the administrator and coordinator username and password information pre-applied in the database.
4. The system will require the administrator and coordinators to login with pre-created usernames and passwords.	Storing the administrators and coordinators credentials in the database.	The database will hold the administrator's and coordinator's username and password.
5. The system will require the administrator to create a new position with a position type, position status, position ID, position semester and Admin ID.	A create new application webpage.	The database will hold all information on new applications entered by the administrator.
6. The system will require students to submit and edit applications based on their classification. Students can attach a resume or a personal statement to the application.	The application webpage will have an application form webpage with all their information and an 'attach' button for a resume and personal statement documents.	The application webpage will store information entered on the application on the database through SQL Queries and documents in an external system.

7. The system will require the administrator to accept/reject a student's application.	A dropdown menu on a student's application to change the status between under review, rejected and approved.	The review student application webpage for administrator only.
8. Administrator will be required to transfer his role to another administrator when replacement is needed.	The transfer role webpage accessed only by the administrator account.	The database will hold the new administrator's username and password.

ASSUMPTIONS

We assume all users have been given a UTEP ID number.

We assume all users have been granted single sign-on credentials by UTEP.

We assume all UTEP students are eligible to be considered for employment.

We assume all users have a valid internet connection.

We assume a personal statement and resume must be separate documents such as pdf or docx

We assume the database has access to an external system that can handle documents such as resumes and personal statements.

ENTITY-RELATIONSHIP DIAGRAM

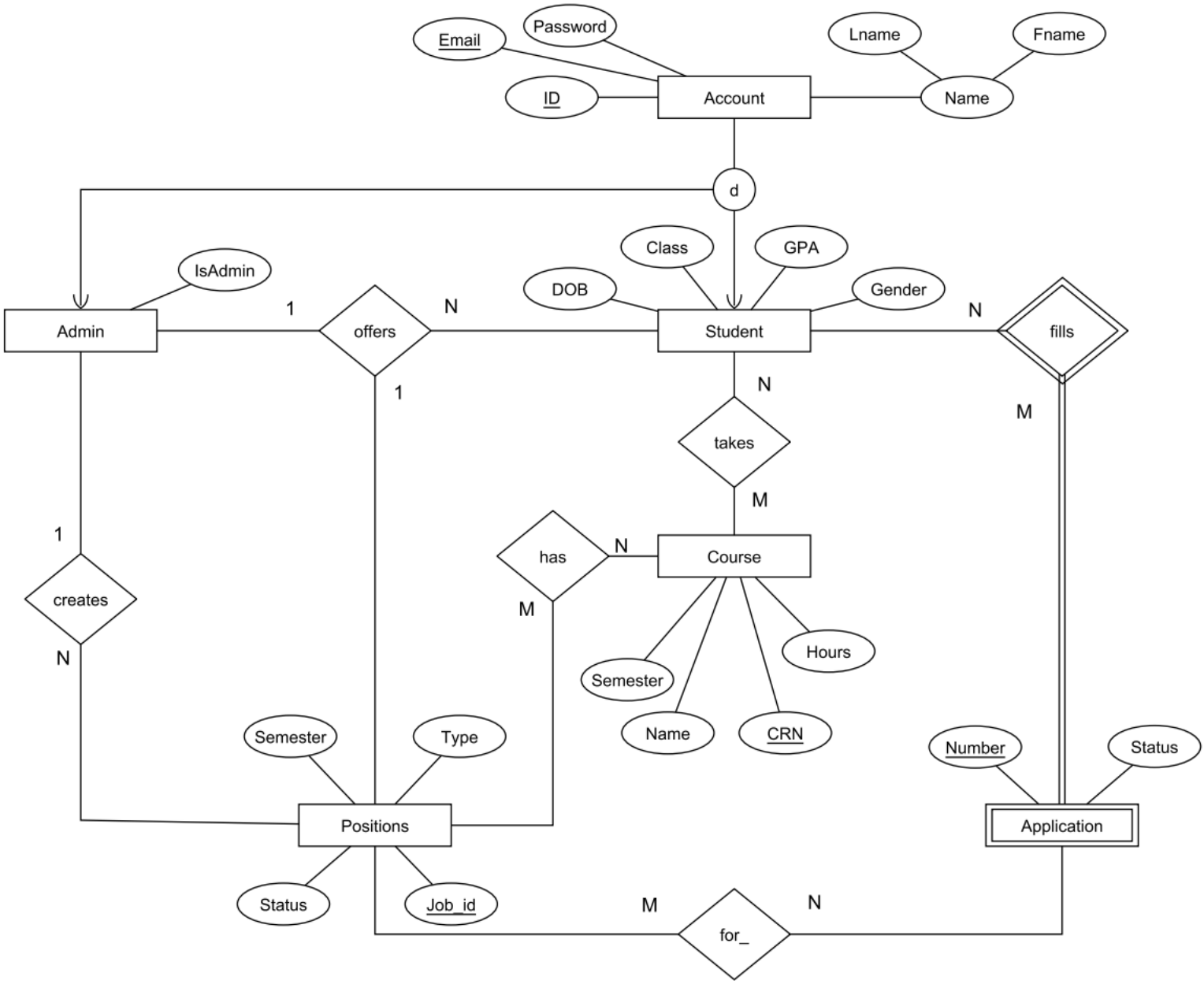

Figure 1-ER Diagram

RELATIONAL MODEL

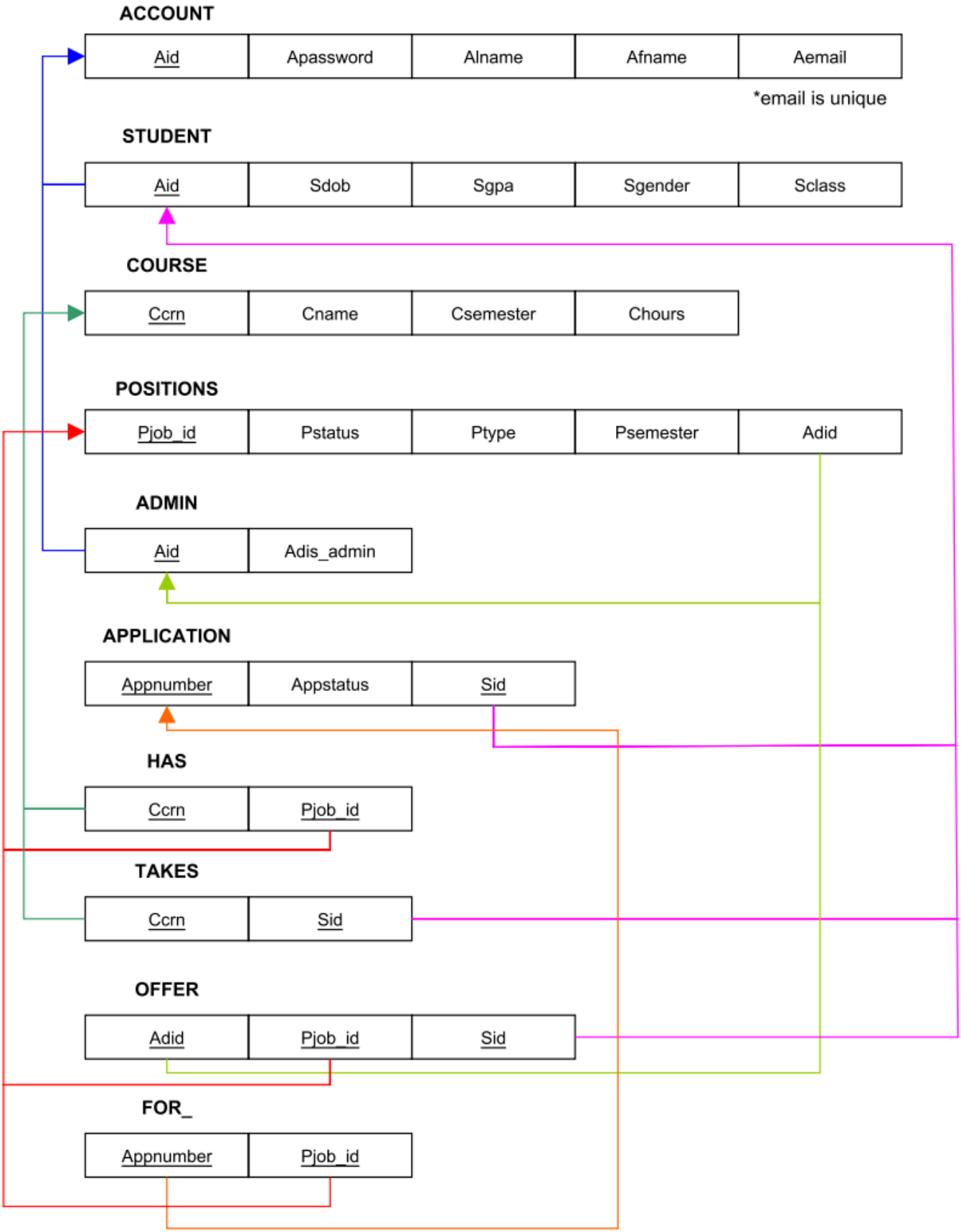

Figure 2-Relational Model

NORMALIZED SCHEMA

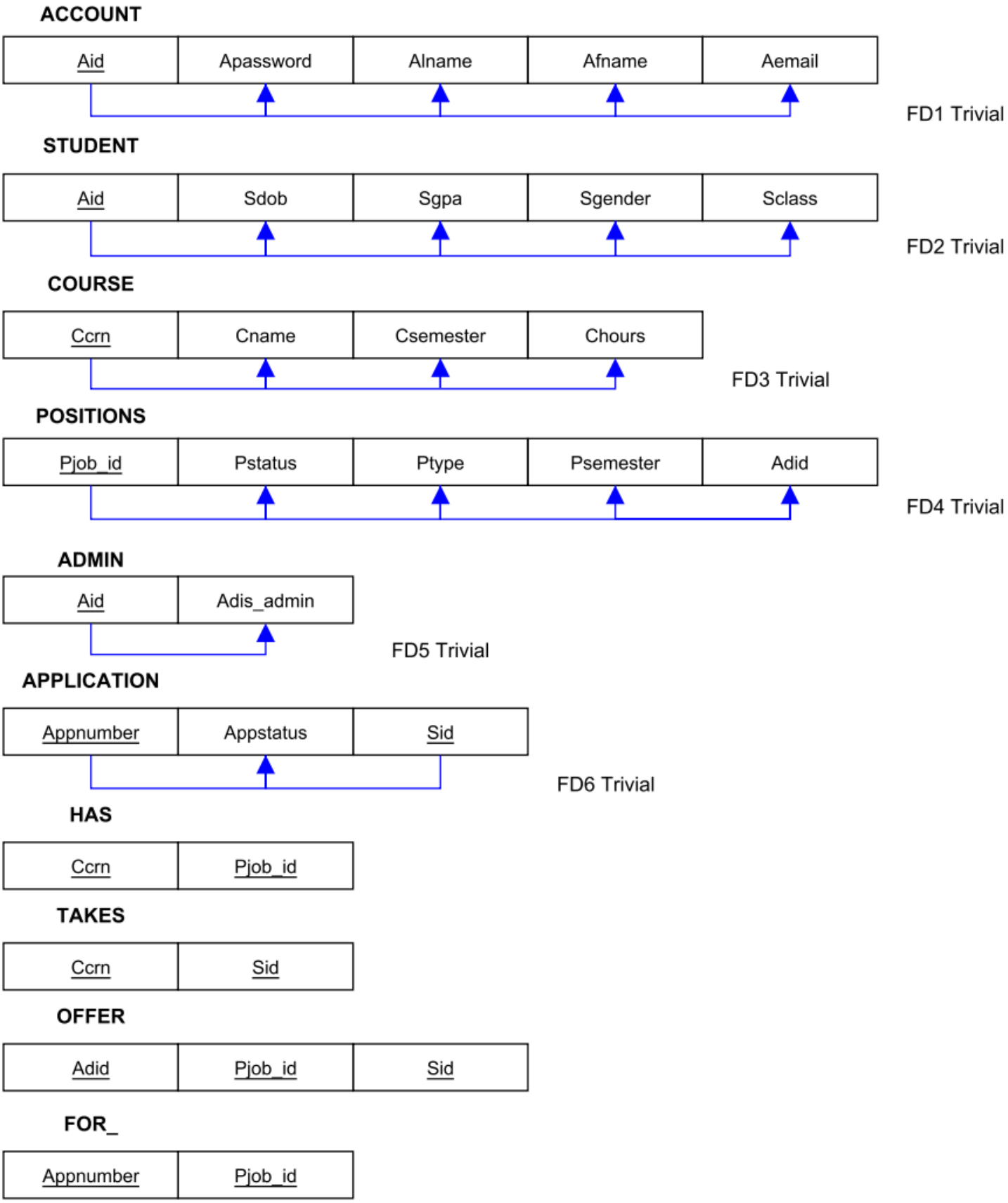

Figure 3-Functional Dependencies

ACCOUNT

The relation ACCOUNT is in 1NF because all its attributes are atomic.
The relation ACCOUNT is in 2NF because it is in 1NF and all non-prime attributes are fully dependent on the PK.
The relation ACCOUNT is in 3NF because it is in 2NF and no non-prime attributes are transitively dependent on the PK.

STUDENT

The relation STUDENT is in 1NF because all its attributes are atomic.
The relation STUDENT is in 2NF because it is in 1NF and all non-prime attributes are fully dependent on the PK.
The relation STUDENT is in 3NF because it is in 2NF and no non-prime attributes are transitively dependent on the PK.

COURSE

The relation COURSE is in 1NF because all its attributes are atomic.
The relation COURSE is in 2NF because it is in 1NF and all non-prime attributes are fully dependent on the PK.
The relation COURSE is in 3NF because it is in 2NF and no non-prime attributes are transitively dependent on the PK.

POSITIONS

The relation POSITIONS is in 1NF because all its attributes are atomic.
The relation POSITIONS is in 2NF because it is in 1NF and all non-prime attributes are fully dependent on the PK.
The relation POSITIONS is in 3NF because it is in 2NF and no non-prime attributes are transitively dependent on the PK.

ADMIN

The relation ADMIN is in 1NF because all its attributes are atomic.
The relation ADMIN is in 2NF because it is in 1NF and all non-prime attributes are fully dependent on the PK.
The relation ADMIN is in 3NF because it is in 2NF and no non-prime attributes are transitively dependent on the PK.

APPLICATION

The relation APPLICATION is in 1NF because all its attributes are atomic.
The relation APPLICATION is in 2NF because it is in 1NF and all non-prime attributes are fully dependent on the PK.
The relation APPLICATION is in 3NF because it is in 2NF and no non-prime attributes are transitively dependent on the PK.

HAS

The relation HAS is in 1NF because all its attributes are atomic.
The relation HAS is in 2NF because it is in 1NF and all attributes form the PK.
The relation HAS is in 3NF because it is in 2NF and all attributes form the PK.

TAKES

The relation TAKES is in 1NF because all its attributes are atomic.
The relation TAKES is in 2NF because it is in 1NF and all attributes form the PK.
The relation TAKES is in 3NF because it is in 2NF and all attributes form the PK.

OFFER

The relation OFFER is in 1NF because all its attributes are atomic.
The relation OFFER is in 2NF because it is in 1NF and all attributes form the PK.
The relation OFFER is in 3NF because it is in 2NF and all attributes form the PK.

FOR_

The relation FOR is in 1NF because all its attributes are atomic.
The relation FOR is in 2NF because it is in 1NF and all attributes form the PK.
The relation FOR is in 3NF because it is in 2NF and all attributes form the PK.

FINAL RELATIONS

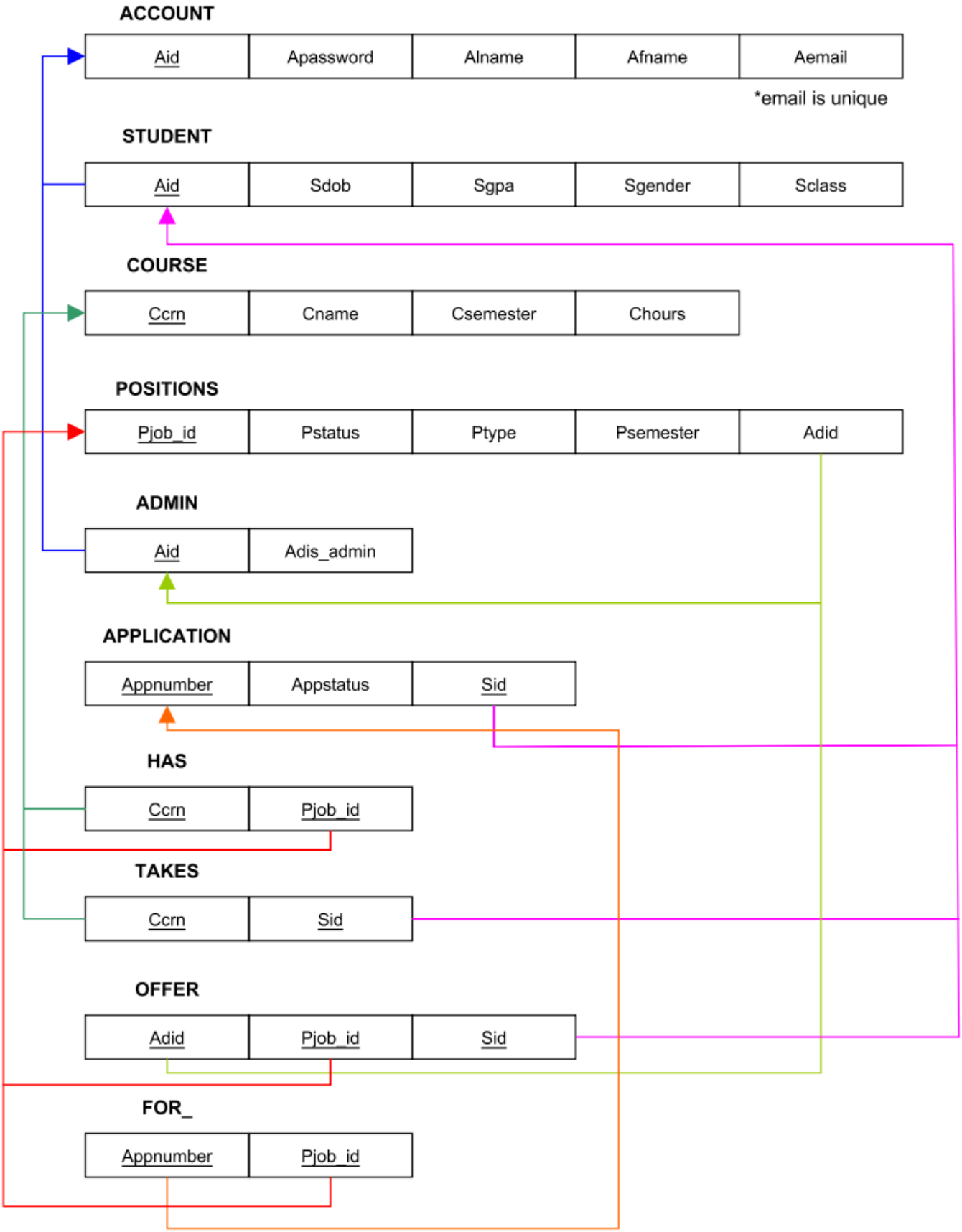

Figure 4-Finalized Schema

Database Schema in MySQL

```
CREATE TABLE ACCOUNT(
Aid CHAR(20) NOT NULL, Apassword CHAR(20), Alname CHAR(20), Afname CHAR(20), Aemail CHAR(50),
PRIMARY KEY(Aid)) Engine=InnoDB;

CREATE TABLE STUDENT(
Aid CHAR(20) NOT NULL, Sdob DATE, Sgpa FLOAT, Sgender CHAR(20), Sclass CHAR(20), PRIMARY KEY(Aid),
FOREIGN KEY (Aid) REFERENCES ACCOUNT(Aid) ON UPDATE CASCADE ON DELETE CASCADE) Engine=InnoDB;

CREATE TABLE COURSE(
Ccrn INT NOT NULL, Cname CHAR(20), Csemester CHAR(20), Chours CHAR(20), PRIMARY KEY(Ccrn)) Engine=InnoDB;

CREATE TABLE ADMIN(
Aid CHAR(20) NOT NULL, Ais_admin BOOLEAN NOT NULL, PRIMARY KEY(Aid), FOREIGN KEY (Aid) REFERENCES
ACCOUNT(Aid) ON UPDATE CASCADE ON DELETE CASCADE) Engine=InnoDB;

CREATE TABLE POSITIONS(
Pjob_id INT NOT NULL, Pstatus CHAR(20), Ptype CHAR(20), Psemester CHAR(20), Adid CHAR(20),
PRIMARY KEY(Pjob_id), FOREIGN KEY (Adid) REFERENCES ADMIN(Aid) ON UPDATE CASCADE ON DELETE CASCADE)
Engine=InnoDB;

CREATE TABLE APPLICATION(
Appnumber INT NOT NULL, Appstatus CHAR(20), Sid CHAR(20) NOT NULL, PRIMARY KEY(Appnumber, Sid),
FOREIGN KEY (Sid) REFERENCES STUDENT(Aid) ON UPDATE CASCADE ON DELETE CASCADE) Engine=InnoDB;

CREATE TABLE HAS(
Ccrn INT NOT NULL, Pjob_id INT NOT NULL, PRIMARY KEY(Ccrn, Pjob_id), FOREIGN KEY (Ccrn) REFERENCES
COURSE(Ccrn) ON UPDATE CASCADE ON DELETE CASCADE, FOREIGN KEY (Pjob_id) REFERENCES POSITIONS(Pjob_id) ON
UPDATE CASCADE ON DELETE CASCADE) Engine=InnoDB;

CREATE TABLE TAKES(
Ccrn INT NOT NULL, Sid CHAR(20) NOT NULL, PRIMARY KEY(Ccrn, Sid), FOREIGN KEY (Ccrn) REFERENCES COURSE(Ccrn)
ON UPDATE CASCADE ON DELETE CASCADE, FOREIGN KEY (Sid) REFERENCES STUDENT(Aid) ON UPDATE CASCADE ON
DELETE CASCADE) Engine=InnoDB;


CREATE TABLE OFFER(
Adid CHAR(20) NOT NULL, Pjob_id INT NOT NULL, Sid CHAR(20) NOT NULL, PRIMARY KEY(Adid, Pjob_id, Sid),
FOREIGN KEY (Adid) REFERENCES ADMIN(Aid) ON UPDATE CASCADE ON DELETE CASCADE, FOREIGN KEY (Pjob_id)
REFERENCES POSITIONS(Pjob_id) ON UPDATE CASCADE ON DELETE CASCADE, FOREIGN KEY (Sid) REFERENCES
STUDENT(Aid) ON UPDATE CASCADE ON DELETE CASCADE) Engine=InnoDB;


CREATE TABLE FOR_(
Appnumber INT NOT NULL, Pjob_id INT NOT NULL, PRIMARY KEY(Appnumber, Pjob_id), FOREIGN KEY (Appnumber)
REFERENCES APPLICATION(Appnumber) ON UPDATE CASCADE ON DELETE CASCADE, FOREIGN KEY (Pjob_id)
REFERENCES POSITIONS(Pjob_id) ON UPDATE CASCADE ON DELETE CASCADE) Engine=InnoDB;
```

Database Records

ACCOUNT:

INSERT INTO ACCOUNT VALUES (‘admin’, ‘admin1’, ‘Administrator’, ‘Professor’, ‘admin@utep.edu’);

INSERT INTO ACCOUNT VALUES (‘user’, ‘user1’, ‘Avalos’, ’Erick’, ‘avalos@utep.edu’);

INSERT INTO ACCOUNT VALUES (‘coordinator1’, ‘coordinator1’, ‘Garcia’, ‘Joe’, ‘coordinator1@utep.edu);

STUDENT:

INSERT INTO student VALUES (‘user’, ‘2000-12-05’, 4.0, ‘M’, ‘Senior’);

INSERT INTO student VALUES (‘student2’, ‘2002-04-11’, 3.85, ‘M’, ‘Junior’);

INSERT INTO student VALUES (‘student3’, ‘1999-10-17’, 3.92, ‘F’, ‘Junior’);

COURSE:

INSERT INTO course VALUES (4342, ‘Database Management’, ‘Fall 2019’, ‘3:00pm-4:20pm’);

INSERT INTO course VALUES (3432, ‘Computer Architecture’, ‘Fall 2019’, ’12:00pm-1:20pm’);

INSERT INTO course VALUES (4311, ‘Software Engineering 1’, ‘Fall 2019’, 9:00am-10:20am’);

ADMIN:

INSERT INTO admin VALUES (‘admin’, true);

INSERT INTO admin VALUES (‘coordinator1’, false);

INSERT INTO admin VALUES (‘coodinator2’, false);

POSITIONS:

INSERT INTO POSITIONS VALUES (422, ‘Open’, ‘TA’, ‘Spring 2020’, ‘admin’);

INSERT INTO POSITIONS VALUES (423, ‘Open’, ‘IA’, ‘Spring 2020’, ‘admin’);

INSERT INTO POSITIONS VALUES (287, ‘Filled’, ‘PL’, ‘Spring 2020’, ‘admin’);

APPLICATION:

INSERT INTO APPLICATION VALUES (86542, ‘Submitted’, ‘user’);

INSERT INTO APPLICATION VALUES (56242, ‘Incomplete’, ‘student2’);

INSERT INTO APPLICATION VALUES (85210, ‘Rejected’, ‘student3’);

HAS:

INSERT INTO HAS VALUES (4342, 422);

INSERT INTO HAS VALUES (4342, 423);

INSERT INTO HAS VALUES (3432, 287);

TAKES:

```
INSERT INTO TAKES VALUES (4342, 'user');
INSERT INTO TAKES VALUES (3432, 'student2');
INSERT INTO TAKES VALUES (4311, 'user');
```

OFFER:

```
INSERT INTO OFFER VALUES ('admin', 422, 'user');
INSERT INTO OFFER VALUES ('admin', 423, 'student2');
INSERT INTO OFFER VALUES ('admin', 287, 'student3');
```

FOR_:

```
INSERT INTO FOR_ VALUES (86542, 422);
INSERT INTO FOR_ VALUES (56242, 423);
INSERT INTO FOR_ VALUES (85210, 287);
```

SQL Queries

Requirement 1: The system requires a host server.

Connected to the host server @ilinkserver.cs.utep.edu

```
MySQL  JS > \connect gviezcas@ilinkserver.cs.utep.edu
Creating a session to 'gviezcas@ilinkserver.cs.utep.edu'
Fetching schema names for autocompletion... Press ^C to stop.
Your MySQL connection id is 2149 (X protocol)
Server version: 8.0.12 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
MySQL  ilinkserver.cs.utep.edu:33060+ ssl  JS > \use S20am_team9
Default schema `S20am_team9` accessible through db.
MySQL  ilinkserver.cs.utep.edu:33060+ ssl  s20am_team9  JS >
```

Requirement 2: The system will require new Students to create a new account using their first name, last name, email, password, username, grade level, DOB, GPA, class, and gender.

SQL Query of account table that stores account ID, password, last name, first name and email.

```
MySQL  ilinkserver.cs.utep.edu:33060+ ssl  s20am_...  SQL > select * from account;
+--------------+--------------+----------------+-----------+-----------------------+
| Aid          | Apassword    | Alname         | Afname    | Aemail                |
+--------------+--------------+----------------+-----------+-----------------------+
| admin        | admin1       | Administrator  | Professor | admin@utep.edu        |
| coordinator1 | coordinator1 | Garcia         | Joe       | coordinator1@utep.edu |
| coordinator2 | coordinator2 | Lopez          | David     | coordinator2@utep.edu |
| student2     | pw2          | Salas          | Miguel    | salas@utep.edu        |
| student3     | pw3          | Sanchez        | Sara      | sanchez@utep.edu      |
| user         | user1        | Avalos         | Erick     | avalos@utep.edu       |
+--------------+--------------+----------------+-----------+-----------------------+
6 rows in set (0.0822 sec)
```

SQL Query of student database that stores Student account ID, GPA, gender, classification and DOB.

```
MySQL  ilinkserver.cs.utep.edu:33060+ ssl  s20am_team9  SQL > select * from student;
+----------+------------+------+---------+---------+
| Aid      | Sdob       | Sgpa | Sgender | Sclass  |
+----------+------------+------+---------+---------+
| student2 | 2002-04-11 | 3.85 | M       | Junior  |
| student3 | 1999-10-17 | 3.92 | F       | Junior  |
| user     | 2000-12-05 |    4 | M       | Senior  |
+----------+------------+------+---------+---------+
3 rows in set (0.0568 sec)
```

Requirements 3, 4 & 8: The system requires one admin and several coordinators with special privileges to oversee applications. The system will require the administrator and coordinators to login with pre-created usernames and passwords. Administrator will be required to transfer his role to another administrator when replacement is needed.

SQL Query of admin database that stores User ID and admin privileges.

```
MySQL  ilinkserver.cs.utep.edu:33060+ ssl  s20am_team9  SQL > select * from admin;
+--------------+-----------+
| Aid          | Ais_admin |
+--------------+-----------+
| admin        |         1 |
| coordinator1 |         0 |
| coordinator2 |         0 |
+--------------+-----------+
3 rows in set (0.0631 sec)
```

Requirement 5 & 7: The system will require the administrator to create a new position with a position type, position status, position ID, position semester and Admin ID. The system will require the administrator to accept/reject a student's position.

SQL Query of position database that stores position type, position status, position ID, position semester and Admin ID.

```
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > select * from positions;
+---------+---------+-------+-------------+--------+
| Pjob_id | Pstatus | Ptype | Psemester   | Adid   |
+---------+---------+-------+-------------+--------+
|     287 | Filled  | PL    | Spring 2020 | admin |
|     422 | Open    | TA    | Spring 2020 | admin |
|     423 | Open    | IA    | Spring 2020 | admin |
+---------+---------+-------+-------------+--------+
3 rows in set (0.0820 sec)
```

Requirement 6 : The system will require students to submit and edit one application per semester based on their classification. Students can attach a resume to the application and include a personal statement.

SQL Query of application database that stores application number, application status and student ID.

```
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > select * from application;
+-----------+------------+----------+
| Appnumber | Appstatus  | Sid      |
+-----------+------------+----------+
|     56242 | Incomplete | student2 |
|     85210 | Rejected   | student3 |
|     86542 | Submitted  | user     |
+-----------+------------+----------+
3 rows in set (0.0534 sec)
```

Views

#View of gpa_search that lists students with a gpa above 3.4 from student table.

```
MySQL  ilinkserver.cs.utep.edu:33060+ ssl  s20am_team9  SQL > create view gpa_search(Gid, Ggpa, Ggender, Gclass, Gdob)
as select Aid, Sgpa, Sgender, Sclass, Sdob from student where Sgpa > '3.4';
Query OK, 0 rows affected (0.1909 sec)
MySQL  ilinkserver.cs.utep.edu:33060+ ssl  s20am_team9  SQL > select*from gpa_search;
+----------+------+---------+----------+------------+
| Gid      | Ggpa | Ggender | Gclass   | Gdob       |
+----------+------+---------+----------+------------+
| a        |  3.5 | M       | Freshman | 1999-01-01 |
| student2 | 3.85 | M       | Junior   | 2002-04-11 |
| student3 | 3.92 | F       | Junior   | 1999-10-17 |
| student5 |  3.5 | M       | Senior   | 1998-03-23 |
| student6 | 3.56 | F       | Junior   | 1998-09-21 |
| user     |    4 | M       | Senior   | 2000-12-05 |
+----------+------+---------+----------+------------+
6 rows in set (0.0569 sec)
```

#View of report_of_classification that lists students at the Junior level form student table.

```
MySQL  ilinkserver.cs.utep.edu:33060+ ssl  s20am_team9  SQL > create view report_of_classifications(Rid, Rgpa, Rgender,
 Rclass, Rdob) as select Aid, Sgpa, Sgender, Sclass, Sdob from student where Sclass = 'Junior';
Query OK, 0 rows affected (0.2744 sec)
MySQL  ilinkserver.cs.utep.edu:33060+ ssl  s20am_team9  SQL > select*from report_of_classifications;
+----------+------+---------+--------+------------+
| Rid      | Rgpa | Rgender | Rclass | Rdob       |
+----------+------+---------+--------+------------+
| student2 | 3.85 | M       | Junior | 2002-04-11 |
| student3 | 3.92 | F       | Junior | 1999-10-17 |
| student6 | 3.56 | F       | Junior | 1998-09-21 |
+----------+------+---------+--------+------------+
3 rows in set (0.0783 sec)
```

#View of student_accounts that lists all student accounts from student table and account table.

```
MySQL  ilinkserver.cs.utep.edu:33060+ ssl  s20am_team9  SQL > create view student_accounts as select a.Aid, Alname, Afn
ame, Aemail, Sdob, Sgpa, Sgender, Sclass from account a, student b where a.Aid = b.Aid;
Query OK, 0 rows affected (0.2167 sec)
MySQL  ilinkserver.cs.utep.edu:33060+ ssl  s20am_team9  SQL > select * from student_accounts;
+----------+---------+--------+------------------+------------+------+---------+----------+
| Aid      | Alname  | Afname | Aemail           | Sdob       | Sgpa | Sgender | Sclass   |
+----------+---------+--------+------------------+------------+------+---------+----------+
| a        | a       | a      | a@a              | 1999-01-01 |  3.5 | M       | Freshman |
| student2 | Salas   | Miguel | salas@utep.edu   | 2002-04-11 | 3.85 | M       | Junior   |
| student3 | Sanchez | Sara   | sanchez@utep.edu | 1999-10-17 | 3.92 | F       | Junior   |
| student5 | Cinco   | Five   | cinco@utep.edu   | 1998-03-23 |  3.5 | M       | Senior   |
| student6 | Seis    | Six    | seis@utep.edu    | 1998-09-21 | 3.56 | F       | Junior   |
| user     | Avalos  | Erick  | avalos@utep.edu  | 2000-12-05 |    4 | M       | Senior   |
+----------+---------+--------+------------------+------------+------+---------+----------+
6 rows in set (0.0537 sec)
```

Procedures

Procedure 1

This procedure can be used to add a new user to the *account* table.

```
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > delimiter $
```

Create the procedure addAccount to add a new user into the Account table.

```
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > create procedure addAccount(IN AidIn CHAR(20), IN Apasswo
rdIn CHAR(20), IN AlnameIn CHAR(20), IN AfnameIn CHAR(20), IN AemailIn CHAR(50)) BEGIN insert into account(Aid, Apasswor
d, Alname, Afname, Aemail) values(AidIn, ApasswordIn, AlnameIn, AfnameIn, AemailIn); END $
Query OK, 0 rows affected (0.2311 sec)
```

Called the addAccount procedure to add a new account to the Account table.

```
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > delimiter ;
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > call addAccount('student8', 'password8', 'Rayos', 'Adrian
', 'rayos@email.com');
```

```
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > select*from account;
+--------------+--------------+---------------+-----------+----------------------+
| Aid          | Apassword    | Alname        | Afname    | Aemail               |
+--------------+--------------+---------------+-----------+----------------------+
|              |              |               |           |                      |
| a            | a            | a             | a         | a@a                  |
| admin        | admin1       | Administrator | Professor | admin@utep.edu       |
| coordinator1 | coordinator1 | Garcia        | Joe       | coordinator1@utep.edu |
| coordinator2 | coordinator2 | Lopez         | David     | coordinator2@utep.edu |
| Maverick     | password     | Cruise        | Tom       | cruise@utep.edu      |
| student2     | pw2          | Salas         | Miguel    | salas@utep.edu       |
| student3     | pw3          | Sanchez       | Sara      | sanchez@utep.edu     |
| student5     | student5     | Cinco         | Five      | cinco@utep.edu       |
| student6     | student6     | Seis          | Six       | seis@utep.edu        |
| student7     | password7    | Dominguez     | Ivan      | dominguez@email.com  |
| student8     | password8    | Rayos         | Adrian    | rayos@email.com      |
| user         | user1        | Avalos        | Erick     | avalos@utep.edu      |
+--------------+--------------+---------------+-----------+----------------------+
13 rows in set (0.0497 sec)
```

Procedure 2

This procedure can be used to list students depending on their class level from the student_accounts view.

The studentClass procedure is created and called to show all Senior students.

```
 MySQL  ilinkserver.cs.utep.edu:33060+ ssl  s20am_team9  SQL > delimiter $
 MySQL  ilinkserver.cs.utep.edu:33060+ ssl  s20am_team9  SQL > create procedure studentClass(IN class CHAR(20)) BEGIN SE
LECT*FROM student_accounts where Sclass = class; END $
Query OK, 0 rows affected (0.1783 sec)
 MySQL  ilinkserver.cs.utep.edu:33060+ ssl  s20am_team9  SQL > delimiter ;
 MySQL  ilinkserver.cs.utep.edu:33060+ ssl  s20am_team9  SQL > call studentClass('Senior');
+----------+--------+--------+----------------+------------+------+---------+--------+
| Aid      | Alname | Afname | Aemail         | Sdob       | Sgpa | Sgender | Sclass |
+----------+--------+--------+----------------+------------+------+---------+--------+
| student5 | Cinco  | Five   | cinco@utep.edu | 1998-03-23 |  3.5 | M       | Senior |
| user     | Avalos | Erick  | avalos@utep.edu | 2000-12-05 | 3.99 | M       | Senior |
+----------+--------+--------+----------------+------------+------+---------+--------+
2 rows in set (0.0547 sec)

Query OK, 0 rows affected (0.0547 sec)
```

Transaction

A transaction can be used when the student is creating an account, the information in each individual field is saved to the transaction table.

```
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > start transaction;
Query OK, 0 rows affected (0.0496 sec)
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > insert into account(Aid, Apassword, Alname, Afname, Aemai
l) values('student7', 'password7', 'Dominguez', 'Ivan', 'dominguez@email.com');
Query OK, 1 row affected (0.0705 sec)
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > select*from account;
+-------------+-------------+--------------+-----------+----------------------+
| Aid         | Apassword   | Alname       | Afname    | Aemail               |
+-------------+-------------+--------------+-----------+----------------------+
|             |             |              |           |                      |
| a           | a           | a            | a         | a@a                  |
| admin       | admin1      | Administrator| Professor | admin@utep.edu       |
| coordinator1| coordinator1| Garcia       | Joe       | coordinator1@utep.edu|
| coordinator2| coordinator2| Lopez        | David     | coordinator2@utep.edu|
| Maverick    | password    | Cruise       | Tom       | cruise@utep.edu      |
| student2    | pw2         | Salas        | Miguel    | salas@utep.edu       |
| student3    | pw3         | Sanchez      | Sara      | sanchez@utep.edu     |
| student5    | student5    | Cinco        | Five      | cinco@utep.edu       |
| student6    | student6    | Seis         | Six       | seis@utep.edu        |
| student7    | password7   | Dominguez    | Ivan      | dominguez@email.com  |
| user        | user1       | Avalos       | Erick     | avalos@utep.edu      |
+-------------+-------------+--------------+-----------+----------------------+
12 rows in set (0.0529 sec)
```

If student cancels or does not finish an application, a rollback can undo the previous insertions.

```
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > rollback;
Query OK, 0 rows affected (0.2002 sec)
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > select*from account;
+-------------+-------------+--------------+-----------+----------------------+
| Aid         | Apassword   | Alname       | Afname    | Aemail               |
+-------------+-------------+--------------+-----------+----------------------+
|             |             |              |           |                      |
| a           | a           | a            | a         | a@a                  |
| admin       | admin1      | Administrator| Professor | admin@utep.edu       |
| coordinator1| coordinator1| Garcia       | Joe       | coordinator1@utep.edu|
| coordinator2| coordinator2| Lopez        | David     | coordinator2@utep.edu|
| Maverick    | password    | Cruise       | Tom       | cruise@utep.edu      |
| student2    | pw2         | Salas        | Miguel    | salas@utep.edu       |
| student3    | pw3         | Sanchez      | Sara      | sanchez@utep.edu     |
| student5    | student5    | Cinco        | Five      | cinco@utep.edu       |
| student6    | student6    | Seis         | Six       | seis@utep.edu        |
| user        | user1       | Avalos       | Erick     | avalos@utep.edu      |
+-------------+-------------+--------------+-----------+----------------------+
11 rows in set (0.0542 sec)
```

When the student does finish and submit the application, the table commits the information to the application table.

```
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > rollback;
Query OK, 0 rows affected (0.2002 sec)
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > select*from account;
+--------------+--------------+---------------+-----------+-----------------------+
| Aid          | Apassword    | Alname        | Afname    | Aemail                |
+--------------+--------------+---------------+-----------+-----------------------+
|              |              |               |           |                       |
| a            | a            | a             | a         | a@a                   |
| admin        | admin1       | Administrator | Professor | admin@utep.edu        |
| coordinator1 | coordinator1 | Garcia        | Joe       | coordinator1@utep.edu |
| coordinator2 | coordinator2 | Lopez         | David     | coordinator2@utep.edu |
| Maverick     | password     | Cruise        | Tom       | cruise@utep.edu       |
| student2     | pw2          | Salas         | Miguel    | salas@utep.edu        |
| student3     | pw3          | Sanchez       | Sara      | sanchez@utep.edu      |
| student5     | student5     | Cinco         | Five      | cinco@utep.edu        |
| student6     | student6     | Seis          | Six       | seis@utep.edu         |
| user         | user1        | Avalos        | Erick     | avalos@utep.edu       |
+--------------+--------------+---------------+-----------+-----------------------+
11 rows in set (0.0542 sec)
```

When the student does finish and submit the application, the table commits the information to the application table.

```
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > insert into account(Aid, Apassword, Alname, Afname, Aemai
l) values('student7', 'password7', 'Dominguez', 'Ivan', 'dominguez@email.com');
Query OK, 1 row affected (0.1961 sec)
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > commit;
Query OK, 0 rows affected (0.0519 sec)
MySQL ilinkserver.cs.utep.edu:33060+ ssl s20am_team9 SQL > select*from account;
+--------------+--------------+---------------+-----------+-----------------------+
| Aid          | Apassword    | Alname        | Afname    | Aemail                |
+--------------+--------------+---------------+-----------+-----------------------+
|              |              |               |           |                       |
| a            | a            | a             | a         | a@a                   |
| admin        | admin1       | Administrator | Professor | admin@utep.edu        |
| coordinator1 | coordinator1 | Garcia        | Joe       | coordinator1@utep.edu |
| coordinator2 | coordinator2 | Lopez         | David     | coordinator2@utep.edu |
| Maverick     | password     | Cruise        | Tom       | cruise@utep.edu       |
| student2     | pw2          | Salas         | Miguel    | salas@utep.edu        |
| student3     | pw3          | Sanchez       | Sara      | sanchez@utep.edu      |
| student5     | student5     | Cinco         | Five      | cinco@utep.edu        |
| student6     | student6     | Seis          | Six       | seis@utep.edu         |
| student7     | password7    | Dominguez     | Ivan      | dominguez@email.com   |
| user         | user1        | Avalos        | Erick     | avalos@utep.edu       |
+--------------+--------------+---------------+-----------+-----------------------+
12 rows in set (0.0532 sec)
```

Triggers

Trigger 1

This trigger is used to return an error if the password field is empty when creating an account.

```
MySQL  localhost:3306  gviezcas_db  SQL > delimiter $
MySQL  localhost:3306  gviezcas_db  SQL > create trigger passwordError before insert on account for each row begin if n
ew.Apassword = '' then set new.Apassword = NULL; end if; end; $
Query OK, 0 rows affected (0.0262 sec)
MySQL  localhost:3306  gviezcas_db  SQL > delimiter ;
MySQL  localhost:3306  gviezcas_db  SQL > INSERT INTO ACCOUNT VALUES ('student', '', 'Rayos', 'Angel', 'rayos2@utep.edu
');
ERROR: 1048 (23000): Column 'Apassword' cannot be null
```

Trigger 2

This trigger archives positions that have been deleted by the administrator.

```
MySQL  localhost:3306  gviezcas_db  SQL > create table archive_positions(id INT PRIMARY KEY AUTO_INCREMENT, Pjob_id INT
 NOT NULL, Pstatus CHAR(20), Ptype CHAR(20), Psemester CHAR(20), Adid CHAR(20), deletedAt TIMESTAMP DEFAULT NOW());
Query OK, 0 rows affected (0.0910 sec)
MySQL  localhost:3306  gviezcas_db  SQL > delimiter $
MySQL  localhost:3306  gviezcas_db  SQL > create trigger before_positions_delete before delete on positions for each ro
w begin insert into archive_positions(Pjob_id, Pstatus, Ptype, Psemester, Adid) values(OLD.Pjob_id, OLD.Pstatus, OLD.Pty
pe, OLD.Psemester, OLD.Adid); END$
Query OK, 0 rows affected (0.0358 sec)
MySQL  localhost:3306  gviezcas_db  SQL > delimiter ;
MySQL  localhost:3306  gviezcas_db  SQL > delete from positions where Pstatus = 'Filled';
Query OK, 1 row affected (0.0114 sec)
MySQL  localhost:3306  gviezcas_db  SQL > select*from archive_positions;
+----+---------+---------+-------+-------------+-------+---------------------+
| id | Pjob_id | Pstatus | Ptype | Psemester   | Adid  | deletedAt           |
+----+---------+---------+-------+-------------+-------+---------------------+
|  1 |     287 | Filled  | PL    | Spring 2020 | admin | 2020-05-05 21:23:53 |
+----+---------+---------+-------+-------------+-------+---------------------+
1 row in set (0.0005 sec)
```

Appendix A

- Database Schema in MySQL – Luis Medina, German Viezcas, Erick Avalos
- Database Records – Erick Avalos
- SQL Queries – German Viezcas
- IIS Web Server and GUI – Kevin Obregon, Erick Avalos

References

Elmasri, R., & Navathe, S. B. (2015). Fundamentals of Database Systems (7th ed.).

Video-Mejia-Ta-Project.mp4. (2020). Retrieved from https://blackboardlearn.utep.edu/bbcswebdav/pid-3036080-dt-content-rid89043959_1/xid-89043959_1

Kevin Apodaca (2020) create_account.php (1.0) [Template Code]. Retrieved from https://blackboardlearn.utep.edu/bbcswebdav/pid-3040893-dt-content-rid89247575_1/xid-89247575_1

Kevin Apodaca (2020) student_login.php (1.0) [Template Code]. Retrieved from https://blackboardlearn.utep.edu/bbcswebdav/pid-3040893-dt-content-rid89247575_1/xid-89247575_1

Kevin Apodaca (2020) config.php (1.0) [Template Code]. Retrieved from https://blackboardlearn.utep.edu/bbcswebdav/pid-3040893-dt-content-rid89247575_1/xid-89247575_1