ARMY ROTC COMPANY LEVEL INFORMATION SYSTEM (ARCLIS)

DESIGNED BY JOSEPH SCHMIDT

TABLE OF CONTENTS

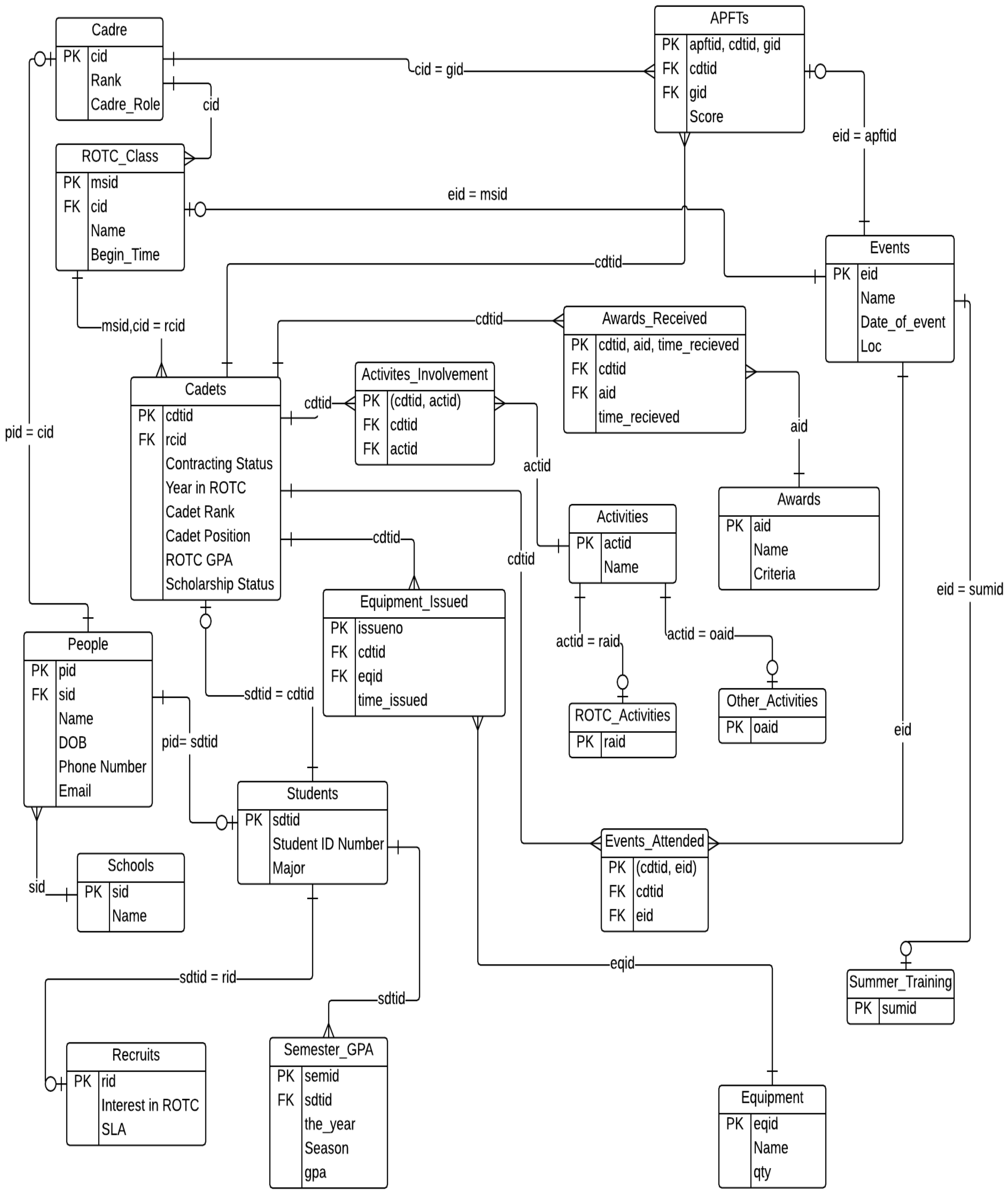
EXECUTIVE SUMMARY:

A big part of the military and ROTC (Reserve Officer Training Corp) is accountability, which means keeping track of everything from personnel, to equipment, to attendance at events. This document shows a design for an information system, ARCLIS, which keeps track of these things for those in an Army ROTC company.

It will be able to keep track of what personnel are involved in the program and in what capacity (i.e. instructor or a cadet) at the company level. For cadets, it will be able to track what equipment cadets have (if any), attendance at events, and other cadet information such as name, email, phone number, host school, contracting status, year in ROTC, APFT scores, GPA, cadet rank, position, activities they're involved in, and DOB. For instructors, or cadre, it will keep track of key information such as name, rank, host school, DOB, email, and the classes they teach. Additionally, there will be a way to keep track of potential recruits for the program, and keep basic information about them such as name, DOB, GPA, Phone Number, Email address, why they are interested in the program, and what makes them SLA (a scholar, leader, and athlete).

This design has been comprised based on the requirements presented by my current ROTC instructors, and with this system, instructors and cadets will have a better method of keeping track of all aspects of the program under one, consistent, relational database (instead of a sort of messy and inconsistent excel spreadsheet). Further, instructors will be able to efficiently update information for cadets on CCIMS (Cadet Command Information Management System), which has been the main inspiration for designing this.

ER DIAGRAM:


Cadre
PK cid
Rank
Cadre_Role
cid = gid
cid
APFTs
PK apftid, cdtid, gid
FK cdtid
FK gid
Score
eid = apftid
ROTC_Class
PK msid
FK cid
Name
Begin_Time
eid = msid
cdtid
Events
PK eid
Name
Date_of_event
Loc
msid,cid = rcid
cdtid
Awards_Received
PK cdtid, aid, time_recieved
FK cdtid
FK aid
time_recieved
aid
pid = cid
Cadets
PK cdtid
FK rcid
Contracting Status
Year in ROTC
Cadet Rank
Cadet Position
ROTC GPA
Scholarship Status
cdtid
Activites_Involvement
PK (cdtid, actid)
FK cdtid
FK actid
actid
Awards
PK aid
Name
Criteria
Activities
PK actid
Name
cdtid
cdtid
Equipment_Issued
PK issueno
FK cdtid
FK eqid
time_issued
eid = sumid
actid = raid
actid = oaid
People
PK pid
FK sid
Name
DOB
Phone Number
Email
sdtid = cdtid
ROTC_Activities
PK raid
Other_Activities
PK oaid
pid= sdtid
eid
Students
PK sdtid
Student ID Number
Major
Events_Attended
PK (cdtid, eid)
FK cdtid
FK eid
Schools
PK sid
Name
sid
eqid
sdtid = rid
sdtid
Summer_Training
PK sumid
Recruits
PK rid
Interest in ROTC
SLA
Semester_GPA
PK semid
FK sdtid
the_year
Season
gpa
Equipment
PK eqid
Name
qty

Tables

People:

Purpose: This will keep of all people associated with the ROTC Company, including the cadets, cadre, and recruits.

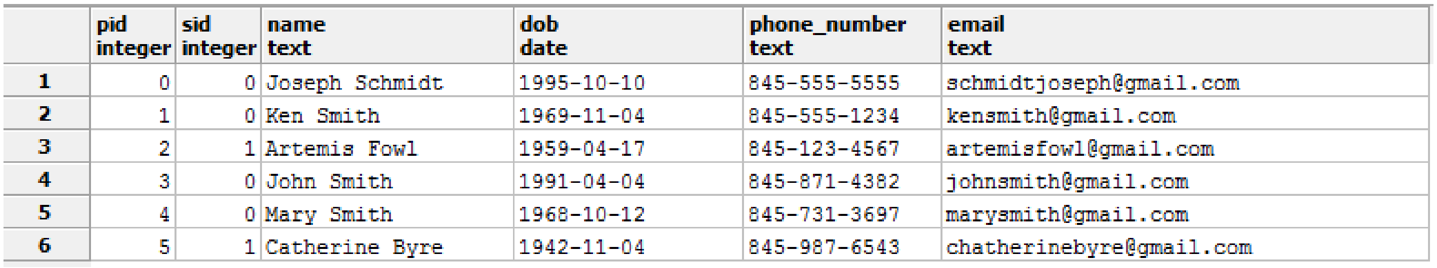
Create Statement:

```
CREATE TABLE people (
        pid int not null,
        sid int not null references schools(sid),
        name text not null,
        dob date not null,
        phone_number text not null,
        email text,
        primary key (pid)
);
```

Functional Dependcies:

pid → sid, name, dob, phone_number, email

Sample Data:



	pid integer	sid integer	name text	dob date	phone_number text	email text
1	0	0	Joseph Schmidt	1995-10-10	845-555-5555	schmidtjoseph@gmail.com
2	1	0	Ken Smith	1969-11-04	845-555-1234	kensmith@gmail.com
3	2	1	Artemis Fowl	1959-04-17	845-123-4567	artemisfowl@gmail.com
4	3	0	John Smith	1991-04-04	845-871-4382	johnsmith@gmail.com
5	4	0	Mary Smith	1968-10-12	845-731-3697	marysmith@gmail.com
6	5	1	Catherine Byre	1942-11-04	845-987-6543	chatherinebyre@gmail.com

Students:

Purpose: This table will keep track of what people are also students.

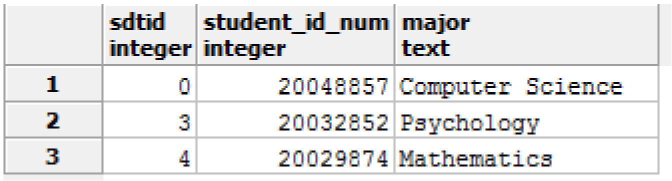
Create Statement:

```
CREATE TABLE students (

        sdtid int not null references people(pid),

        student_id_num int not null,

        major text not null,

        primary key (sdtid)

);
```

Functional Dependencies:

sdtid → student_id_num, major

Sample Data:



	sdtid integer	student_id_num integer	major text
1	0	20048857	Computer Science
2	3	20032852	Psychology
3	4	20029874	Mathematics

CADETS:

PURPOSE: This table will keep track of the cadets that are in the program. There are a lot of entities associated with this table since cadets are the core of the program. They are also students, so this is also a sub-type. Some cadets could be in a different ROTC class despite their year in ROTC. ROTC GPA is cumulative and will have to be updated (academic GPA is more important, which is why it is handled differently).

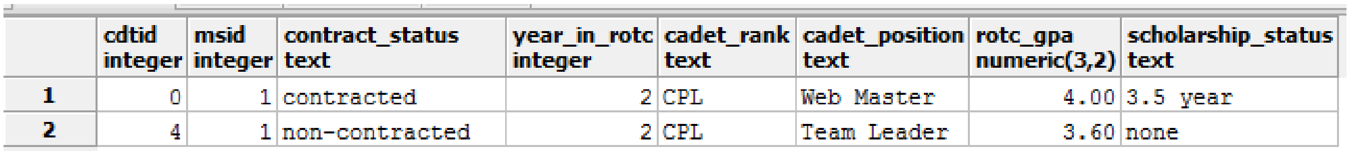
CREATE STATEMENT:

```
        CREATE TABLE cadets (
                cdtid int not null references students(sdtid),
                msid int not null references rotc_class(msid),
                contract_status text not null check
                        (contract_status in ('contracted', 'non-contracted')),
                year_in_rotc int not null,
                cadet_rank text,
                cadet_position text,
                rotc_gpa numeric(3,2) not null,
                scholarship_status text not null,
                primary key (cdtid)
        );
```

FUNCTIONAL DEPENDENCIES:

cdtid → msid, contract_status, year_in_rotc, cader_rank, cadet_position, rotc_gpa, scholarship_status

SAMPLE DATA:



	cdtid	msid	contract_status	year_in_rotc	cadet_rank	cadet_position	rotc_gpa	scholarship_status
	integer	integer	text	integer	text	text	numeric(3,2)	text
1	0	1	contracted	2	CPL	Web Master	4.00	3.5 year
2	4	1	non-contracted	2	CPL	Team Leader	3.60	none

SCHOOLS:

PURPOSE: This tracks what colleges or universities that people are associated with.

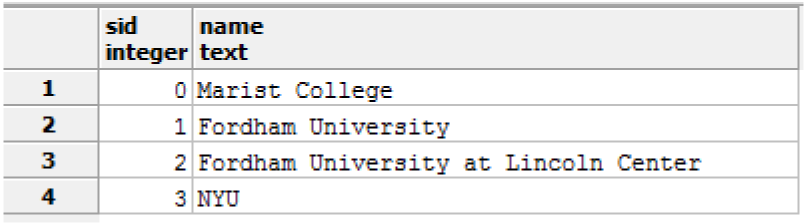
CREATE STATEMENT:

```
CREATE TABLE schools (

sid int not null,

name text not null,

primary key (sid)

);
```

FUNCTIONAL DEPENDENCIES:

sid → name

SAMPLE DATA:



	sid integer	name text
1	0	Marist College
2	1	Fordham University
3	2	Fordham University at Lincoln Center
4	3	NYU

Recruits:

PURPOSE: This will keep track of the potential recruits for the program, including their interest in the program, and what makes them an SLA (scholar, leader, athlete). SLA and their interest in ROTC, will be brief blurbs on their respective subjects. SLA represents the several qualities of what an ROTC cadet should be (a scholar, leader, and athlete).

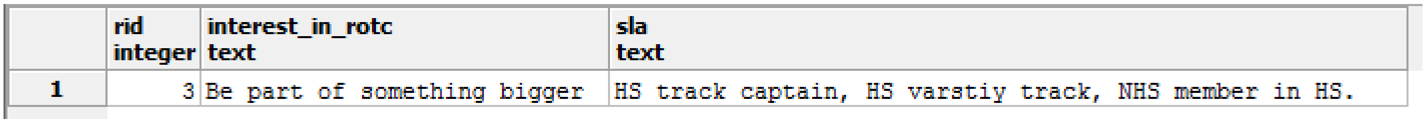
Create Statement:

```
CREATE TABLE recruits (
        rid int not null references students(sdtid),
        interest_in_rotc text not null,
        sla text not null,
        primary key (rid)
);
```

Functional Dependencies:

rid → interest_in_rotc, sla

Sample Data:



	rid integer	interest_in_rotc text	sla text
1	3	Be part of something bigger	HS track captain, HS varstiy track, NHS member in HS.

SEMESTER_GPA:

PURPOSE: This is so the GPAs for students, cadets and recruits, can be tracked. There will be a calculation for cumulGPA which takes the average gpa based on all of the rows for that cadet.

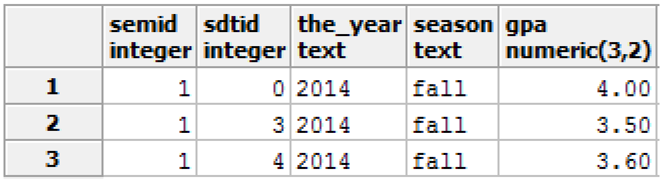
CREATE STATEMENT:

```
CREATE TABLE semester_gpa (
        semid int not null,
        sdtid int not null references students(sdtid),
        the_year text not null,
        season text not null check
                (season in ('fall', 'spring')),
        gpa numeric(3,2) not null,
        cumul_gpa numeric(3,2) not null,
        primary key (semid)
);
```

FUNCTIONAL DEPENDCIES:

semid → sdtid, the_year, season, gpa, cumul_gpa

SAMPLE DATA:



	semid integer	sdtid integer	the_year text	season text	gpa numeric(3,2)
1	1	0	2014	fall	4.00
2	1	3	2014	fall	3.50
3	1	4	2014	fall	3.60

Cadre:

Purpose: This will keep track of the Cadre, who are senior military professionals that help train cadets in the program. Potentially, they could be students at a college, but it is irrelevant since they are instructors for the program and not participants in it.

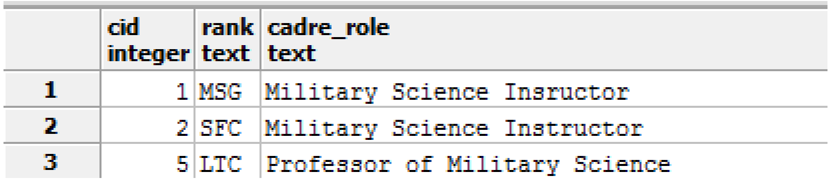
Create Statement:

```
CREATE TABLE cadre (
        cid int not null references people(pid),
        rank text not null,
        cadre_role text not null,
        primary key (cid)
);
```

Functional Dependencies:

cid → rank, cadre_role

Sample Data:



	cid integer	rank text	cadre_role text
1	1	MSG	Military Science Insructor
2	2	SFC	Military Science Instructor
3	5	LTC	Professor of Military Science

EVENTS:

PURPOSE: This entity will keep track of the events that ROTC conducts. It will help keep track of all events from summer training, to class, to APFTs.

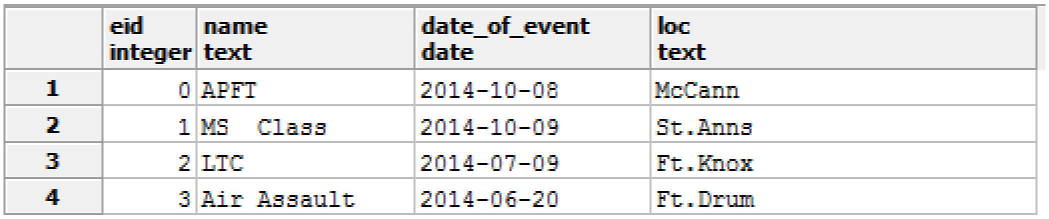
CREATE STATEMENT:

```
CREATE TABLE events (
        eid int not null,
        name text not null,
        date_of_event date not null,
        loc text not null,
        primary key (eid)
);
```

FUNCTIONAL DEPENDENCIES:

eid → name, date_of_event, loc

SAMPLE DATA:



	eid integer	name text	date_of_event date	loc text
1	0	APFT	2014-10-08	McCann
2	1	MS Class	2014-10-09	St.Anns
3	2	LTC	2014-07-09	Ft.Knox
4	3	Air Assault	2014-06-20	Ft.Drum

EVENTS_ATTENDED:

PURPOSE: This entity will keep track of what cadets attend events. It is useful for determining who is deserving of certain awards, and can show cadets are actually showing up for events.

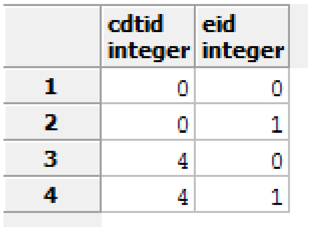
CREATE STATEMENT:

```
    CREATE TABLE events_attended (
        cdtid int not null references cadets(cdtid),
        eid int not null references events(eid),
        primary key (cdtid, eid)
    );
```

FUNCTIONAL DEPENDENCIES:

cdtid,eid →

SAMPLE DATA:



	cdtid integer	eid integer
1	0	0
2	0	1
3	4	0
4	4	1

ROTC_CLASS:

PURPOSE: This will keep track of the ROTC classes that are held and what cadre member teaches the class, along with what time the class starts.

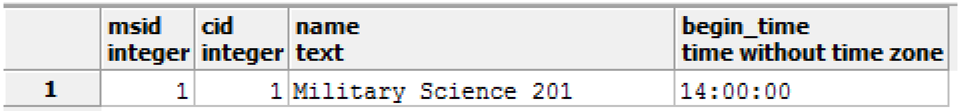
CREATE STATEMENT:

```
CREATE TABLE rotc_class (
        msid int not null references events(eid),
        cid int not null references cadre(cid),
        name text not null,
        begin_time time not null,
        primary key (msid)
);
```

FUNCTIONAL DEPENDENCIES:

msid → cid, name, begin_time

SAMPLE DATA:



	msid integer	cid integer	name text	begin_time time without time zone
1	1	1	Military Science 201	14:00:00

EQUIPMENT:

PURPOSE: This will keep track of the equipment that the company has in its possession. Quantitiy will be calculated (subtracted from) based on the rows inserted or deleted from the table. The equipment that has been issued appears in an entity that is explained on the next page. Contracted and non-contracted can receive equipment, and there will be an alert for when qty reaches five and zero.

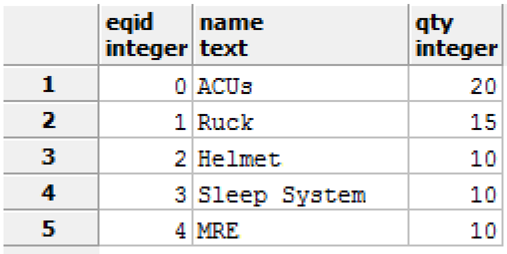
CREATE STATEMENT:

```
CREATE TABLE equipment (
        eqid int not null,
        name text not null,
        qty int not null,
        primary key (eqid)
);
```

FUNCTIONAL DEPENDENCIES:

eqid → name, qty

SAMPLE DATA:



	eqid integer	name text	qty integer
1	0	ACUs	20
2	1	Ruck	15
3	2	Helmet	10
4	3	Sleep System	10
5	4	MRE	10

EQUIPMENT_ISSUED:

PURPOSE: This keeps track of the equipment issued by combining the cadet’s id and the equipment id, and helps determine the quantity in the equipment table.

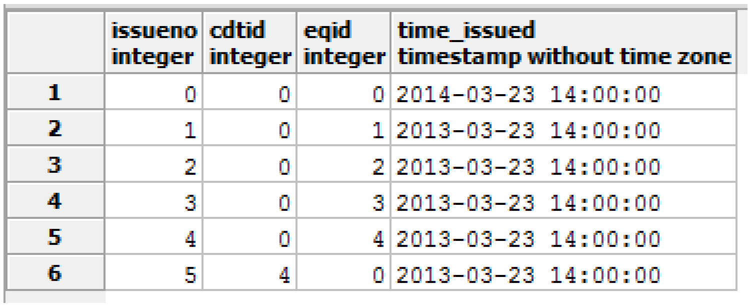
CREATE STATEMENT:

```
CREATE TABLE equipment_issued (
        issueno int not null,
        cdtid int not null references cadets(cdtid),
        eqid int not null references equipment(eqid),
        time_issued timestamp not null,
        primary key (issueno)
);
```

FUNCTIONAL DEPENDENCIES:

issueno → cdtid, eqid, time_issued

SAMPLE DATA:



	issueno	cdtid	eqid	time_issued
	integer	integer	integer	timestamp without time zone
1	0	0	0	2014-03-23 14:00:00
2	1	0	1	2013-03-23 14:00:00
3	2	0	2	2013-03-23 14:00:00
4	3	0	3	2013-03-23 14:00:00
5	4	0	4	2013-03-23 14:00:00
6	5	4	0	2013-03-23 14:00:00

ACTIVITIES:

PURPOSE: This keeps track of the activities that cadets can be involved in; from ROTC activities to other activities across campus, which are reflected in other tables (see ROTC_activities and Other_activities).

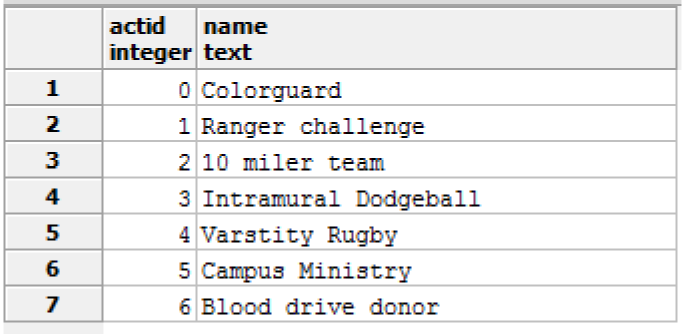
CREATE STATEMENT:

```
        CREATE TABLE activities (
                actid int not null,
                name text not null,
                primary key (actid)
        );
```

FUNCTIONAL DEPENDENCIES:

actid →name

SAMPLE DATA:



	actid integer	name text
1	0	Colorguard
2	1	Ranger challenge
3	2	10 miler team
4	3	Intramural Dodgeball
5	4	Varstity Rugby
6	5	Campus Ministry
7	6	Blood drive donor

ROTC_ACTIVITIES:

PURPOSE: This keeps track of the ROTC activities that cadets can be involved in. Examples include color guard, ranger challenge, and the 10 miler team.

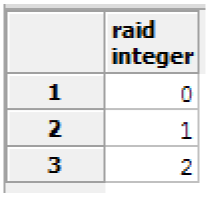
CREATE STATEMENT:

```
        CREATE TABLE rotc_activities (
                raid int not null references activities(actid) ,
                primary key (raid)
        );
```

FUNCTIONAL DEPENDENCIES:

raid →

SAMPLE DATA:



	raid integer
1	0
2	1
3	2

OTHER_ACTIVITIES:

PURPOSE: This keeps track of the other activities that cadets can be involved in. Examples include intramural dodgeball, campus ministry, and being a blood drive donor.

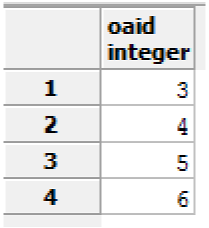
CREATE STATEMENT:

```
CREATE TABLE other_activities (
        oaid int not null references activities(actid),
        primary key (oaid)
);
```

FUNCTIONAL DEPENDENCIES:

oaid →

SAMPLE DATA:



	oaid integer
1	3
2	4
3	5
4	6

ACTIVITIES_INVOLVEMENT:

PURPOSE: This keeps track of the activities that cadets are involved in; from ROTC activities to other activities across campus.

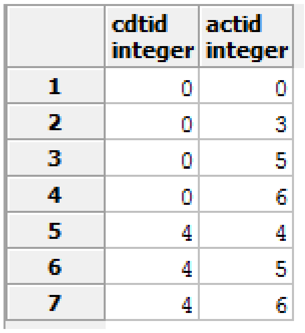
CREATE STATEMENT:

```
CREATE TABLE activities_involvement (
        cdtid int not null references cadets(cdtid),
        actid int not null references activities(actid),
        primary key (cdtid, actid)
);
```

FUNCTIONAL DEPENDENCIES:

cdtid, actid →

SAMPLE DATA:



	cdtid integer	actid integer
1	0	0
2	0	3
3	0	5
4	0	6
5	4	4
6	4	5
7	4	6

Awards:

Purpose: This keeps track of the awards that cadets can receive. There are a variety of rewards, and they are at the discretion of the instructor to give out awards, so joins will be necessary to determine what awards a cadet is entitled to (this specifically will be explained later).

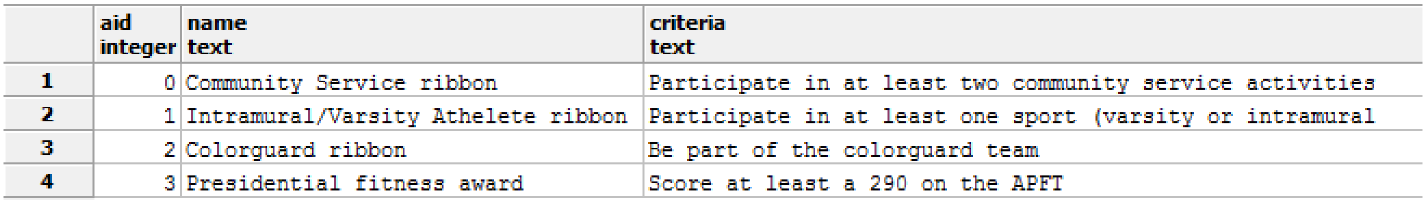
Create Statement:

```
        CREATE TABLE awards (
                aid int not null ,
                name text not null,
                criteria text not null,
                primary key (aid)
        );
```

Functional Dependencies:

aid →name, criteria

Sample Data:



	aid integer	name text	criteria text
1	0	Community Service ribbon	Participate in at least two community service activities
2	1	Intramural/Varsity Athelete ribbon	Participate in at least one sport (varsity or intramural
3	2	Colorguard ribbon	Be part of the colorguard team
4	3	Presidential fitness award	Score at least a 290 on the APFT

AWARDS_RECEIVED:

PURPOSE: This keeps track of the awards that cadets have received.

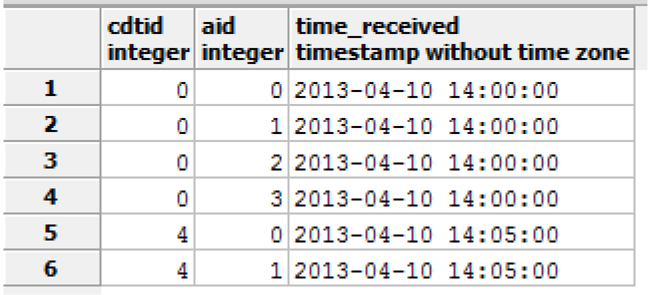
CREATE STATEMENT:

```
CREATE TABLE awards_received (
        cdtid int not null references cadets(cdtid),
        aid int not null references awards(aid) ,
        time_received timestamp not null,
        primary key (cdtid, aid, time_received )
);
```

FUNCTIONAL DEPENDENCIES:

cdtid, aid, time_received →

SAMPLE DATA:



	cdtid integer	aid integer	time_received timestamp without time zone
1	0	0	2013-04-10 14:00:00
2	0	1	2013-04-10 14:00:00
3	0	2	2013-04-10 14:00:00
4	0	3	2013-04-10 14:00:00
5	4	0	2013-04-10 14:05:00
6	4	1	2013-04-10 14:05:00

SUMMER_TRAINING:

PURPOSE: This keeps track of the events that are summer training events which cadets can go to and participate in.

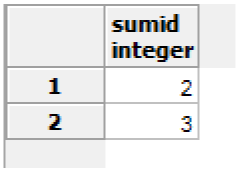
CREATE STATEMENT:

```
        CREATE TABLE summer_training (
                sumid int not null references events(eid),
                primary key (sumid)
        );
```

FUNCTIONAL DEPENDENCIES:

sumid →

SAMPLE DATA:



	sumid integer
1	2
2	3

APFTS:

PURPOSE: This keeps track of the events that are APFTs or army physical fitness tests. Each cadet that takes an APFT has a grader (gid), and it is more important than just normal PT (physical training) since cadets get graded or scored for how well they do on the test (0-300 scale).

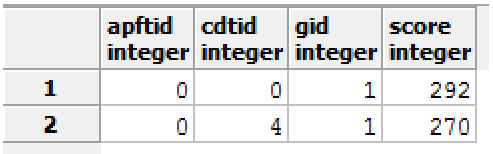
CREATE STATEMENT:

```
        CREATE TABLE apfts (
                apftid int not null references events(eid),
                cdtid int not null references cadets(cdtid),
                gid int not null references cadre(cid),
                score int not null check (score >= 0 and score <= 300),
                primary key (apftid, cdtid, gid)
        );
```

FUNCTIONAL DEPENDENCIES:

apftid, cdtid, gid → score

SAMPLE DATA:



	apftid integer	cdtid integer	gid integer	score integer
1	0	0	1	292
2	0	4	1	270

Views

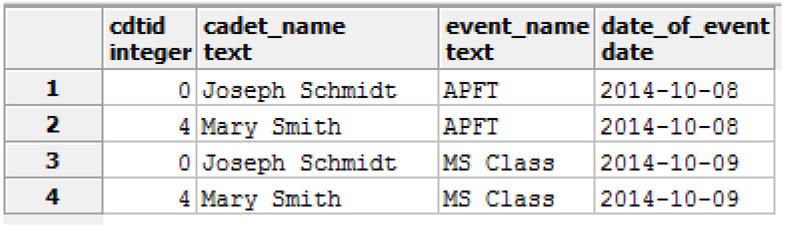
Attendance:

Purpose: This view will allow instructors or cadet officers to see what events cadets have been attending. This will be especially useful for instructors who want to know who is or is not attending events, as to be an official ROTC cadet you have to participate, and attendance for events (or accountability) is extremely important in the military, and is something instructors and cadet officers, conscientiously keep track of. This can also be queried of a specific event if necessary.

Create Statement: (view output on next page)

```
CREATE VIEW attendance AS
SELECT c.cdtid, p.name AS Cadet_Name,  e.name AS Event_name,
e.date_of_event
FROM events_attended ea INNER JOIN events e
ON ea.eid = e.eid
INNER JOIN cadets c
ON ea.cdtid = c.cdtid
INNER JOIN students s
ON c.cdtid = s.sdtid
INNER JOIN people p
ON s.sdtid = p.pid
ORDER BY e.name asc,
         c.cdtid asc
```

VIEW OUTPUT:



	cdtid	cadet_name	event_name	date_of_event
	integer	text	text	date
1	0	Joseph Schmidt	APFT	2014-10-08
2	4	Mary Smith	APFT	2014-10-08
3	0	Joseph Schmidt	MS Class	2014-10-09
4	4	Mary Smith	MS Class	2014-10-09

REPORTS

EQUIPMENT CHECK:

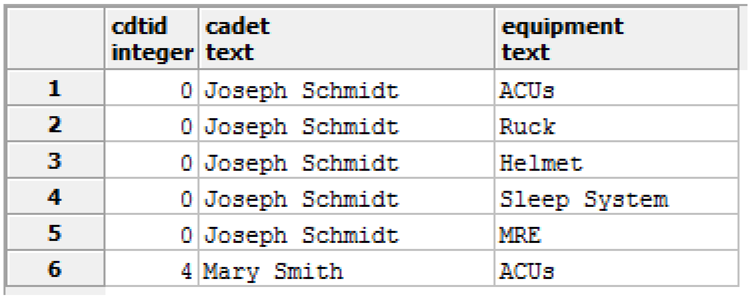
PURPOSE: Equipment checks are always done before and after training events to keep accountability of equipment for all cadets in the company. This will make it much easier to do such checks, since everyone's issued equipment will be in one place (which is how equipment checks work), and if a cadet doesn't have something but they were issued it, this report will prove that they are responsible for that equipment. The importance lies with who was issued what; any other information for this report would superfluous.

QUERY:

```
SELECT c.cdtid, p.name AS Cadet, e.name AS Equipment
FROM cadets c INNER JOIN equipment_issued ei ON ei.cdtid = c.cdtid
INNER JOIN students s ON c.cdtid = s.sdtid
INNER JOIN people p ON s.sdtid = p.pid
INNER JOIN equipment e ON ei.eqid = e.eqid
```

(Changed the format slightly so I could keep it on one page.)

QUERY OUTPUT:



	cdtid integer	cadet text	equipment text
1	0	Joseph Schmidt	ACUs
2	0	Joseph Schmidt	Ruck
3	0	Joseph Schmidt	Helmet
4	0	Joseph Schmidt	Sleep System
5	0	Joseph Schmidt	MRE
6	4	Mary Smith	ACUs

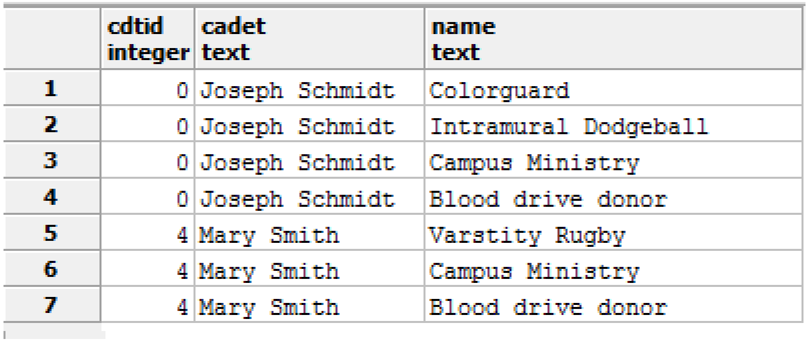
AWARDS FOR ACTIVITIES:

PURPOSE: To see what awards cadets have qualified for, this report will let instructors know what awards cadets are entitled to base on their participation in certain activities. This will require the instructor to compare the awards to this report, though this will allow the instructor to take into other factors that are a basis for whether or not the cadet deserves this award, such as their character and their overall attitude (which cannot be tracked by the database).

QUERY:

```
SELECT c.cdtid, p.name AS Cadet, a.name
FROM cadets c INNER JOIN activities_involvement ai
ON ai.cdtid = c.cdtid
INNER JOIN students s
ON c.cdtid = s.sdtid
INNER JOIN people p
ON s.sdtid = p.pid
INNER JOIN activities a
ON ai.actid = a.actid
```

QUERY OUTPUT:



	cdtid integer	cadet text	name text
1	0	Joseph Schmidt	Colorguard
2	0	Joseph Schmidt	Intramural Dodgeball
3	0	Joseph Schmidt	Campus Ministry
4	0	Joseph Schmidt	Blood drive donor
5	4	Mary Smith	Varstity Rugby
6	4	Mary Smith	Campus Ministry
7	4	Mary Smith	Blood drive donor

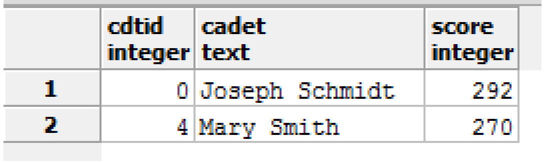
AWARDS FOR PT:

PURPOSE: To see what awards cadets have qualified for in terms of physical training, mainly the army physical fitness test, this report will work like the “Awards for activities” report in that the awards will still be given out at the discretion of the instructor and based on award criteria.

QUERY:

```
SELECT c.cdtid, p.name AS Cadet, pt.score

FROM cadets c INNER JOIN apfts pt

ON pt.cdtid = c.cdtid

INNER JOIN students s

ON c.cdtid = s.sdtid

INNER JOIN people p

ON s.sdtid = p.pid
```

QUERY OUTPUT:



	cdtid	cadet	score
	integer	text	integer
1	0	Joseph Schmidt	292
2	4	Mary Smith	270

STORED PROCEDURES

EQUIPMENT_CHECK_FOR():

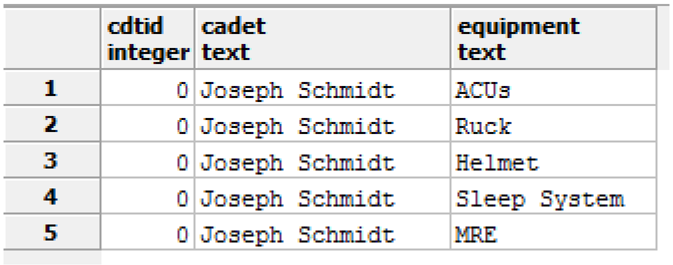
PURPOSE: If a report is needed for individual cadets equipment, this can be done with this procedure, and works similarly to the report “Equipment Check”.

PROCEDURE:

```
CREATE OR REPLACE FUNCTION equipment_check_for(text, REFCURSOR) RETURNS REFCURSOR
AS  $$

DECLARE

                                        cadetName text       :=$1;

                                        resultset REFCURSOR  :=$2;

BEGIN

  OPEN resultset FOR

  SELECT c.cdtid, p.name AS Cadet, e.name AS Equipment

  FROM cadets c INNER JOIN equipment_issued ei ON ei.cdtid = c.cdtid

  INNER JOIN students s ON c.cdtid = s.sdtid

  INNER JOIN people p ON s.sdtid = p.pid

  INNER JOIN equipment e ON ei.eqid = e.eqid

  WHERE p.name LIKE '%'||cadetName||'%';

  RETURN resultset;

END;

$$

LANGUAGE plpgsql;   (Changed the format slightly so I could keep it on one page.)
```

PROCEDURE OUTPUT:

```
SELECT equipment_check_for('Joseph Schmidt', 'results');

FETCH ALL FROM results;
```



	cdtid	cadet	equipment
	integer	text	text
1	0	Joseph Schmidt	ACUs
2	0	Joseph Schmidt	Ruck
3	0	Joseph Schmidt	Helmet
4	0	Joseph Schmidt	Sleep System
5	0	Joseph Schmidt	MRE

SUBTRACT_FROM_QTY():

PURPOSE: When equipment is issued, this stored procedure will keep the quantity of the equipment issued consistent (see the trigger for this procedure for a sample output).

PROCEDURE:

```
CREATE OR REPLACE FUNCTION subtract_from_qty() RETURNS TRIGGER AS
$$
BEGIN
  UPDATE equipment
  SET qty = qty - 1
  WHERE equipment.eqid = (SELECT ei.eqid
                          FROM equipment_issued ei
                          WHERE ei.issueno = (SELECT max(issueno)
                                              FROM equipment_issued
                                             )
                         );
  RETURN NEW;
  IF NEW = 0 THEN
    RETURN 'Last Equipment item issued';
    END IF;
END;
$$
LANGUAGE plpgsql;
```

ADD_TO_QTY():

PURPOSE: When equipment is being returned, this stored procedure will keep the quantity of the equipment issued consistent (see the trigger for this procedure for a sample output).

PROCEDURE: (continues on next page)

```
CREATE OR REPLACE FUNCTION add_to_qty() RETURNS TRIGGER AS
$$
BEGIN
  UPDATE equipment
  SET qty = qty + 1
  WHERE equipment.eqid = (SELECT ei.eqid
                          FROM equipment_issued ei
                          WHERE ei.issueno = (SELECT max(issueno)
                                              FROM equipment_issued
                                             )
                         );
  RETURN NEW;
END;
$$
LANGUAGE plpgsql;
```

CUMULATIVE GPA CALCULATION:

PURPOSE: Sometimes instructors need to know what cumulative GPA a cadet has for CCIMS or some other form and this calculation will make that easy. This will be based on what GPA they received over the semesters.

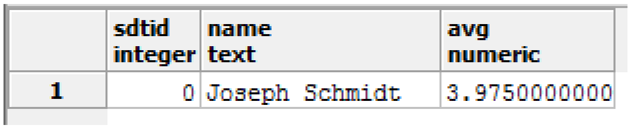
PROCEDURE:

```
CREATE OR REPLACE FUNCTION cumulGPA(text, REFCURSOR) RETURNS REFCURSOR AS
$$
DECLARE
        cadetName text      :=$1;
        resultset REFCURSOR  :=$2;
BEGIN
OPEN resultset FOR
SELECT s.sdtid, p.name, avg(gpa)
FROM semester_gpa sg INNER JOIN students s
ON s.sdtid = sg.sdtid
INNER JOIN people p
ON p.pid = s.sdtid
WHERE p.name like '%'||cadetName||'%'
GROUP BY s.sdtid, p.name;


RETURN resultset;
END;
$$
LANGUAGE plpgsql;
```

EXAMPLE OUTPUT:

(added another entry for this output example to make it a bit more interesting)

```
INSERT INTO semester_gpa(semid, sdtid, the_year, season, gpa)

        VALUES(0, 0, '2013', 'fall', 3.95);

SELECT cumulGPA('Joseph Schmidt', 'results');

FETCH ALL FROM results;
```



	sdtid integer	name text	avg numeric
1	0	Joseph Schmidt	3.9750000000

TRIGGERS

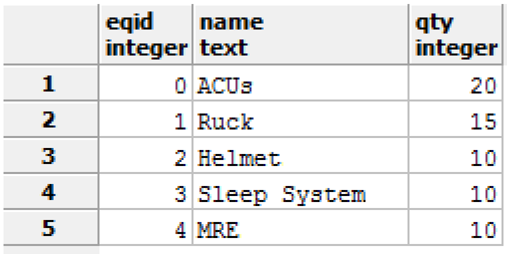
SUBTRACT_FROM_QTY:

PURPOSE: This will keep the equipment quantities consistent by subtracting the quantity by one when a piece of equipment is issued.

CREATE STATEMENT:

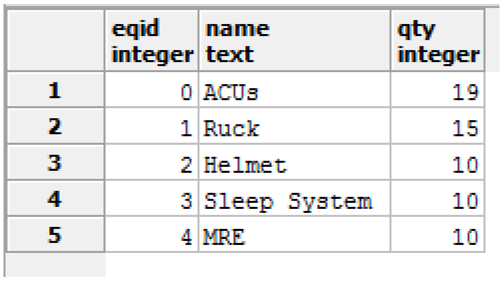
```
CREATE TRIGGER subtract_from_qty
AFTER INSERT ON equipment_issued
FOR EACH STATEMENT EXECUTE PROCEDURE subtract_from_qty();
```

EXAMPLE OUTPUT: (The table below is before the insert statement)



	eqid integer	name text	qty integer
1	0	ACUs	20
2	1	Ruck	15
3	2	Helmet	10
4	3	Sleep System	10
5	4	MRE	10

```
INSERT INTO equipment_issued(issueno, cdtid, eqid, time_issued)
        VALUES(6, 0, 0, '2014-12-02 14:00:00');
```



	eqid integer	name text	qty integer
1	0	ACUs	19
2	1	Ruck	15
3	2	Helmet	10
4	3	Sleep System	10
5	4	MRE	10

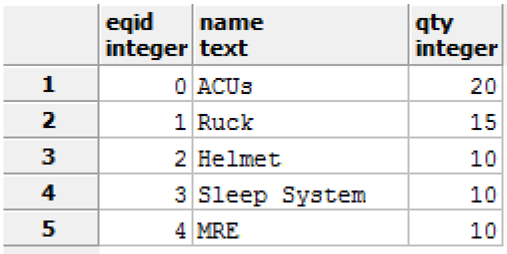
ADD_TO_QTY:

PURPOSE: This will keep the equipment quantities consistent by adding one to the quantity when a piece of equipment is returned (deleted from equipment_issued).

CREATE STATEMENT:

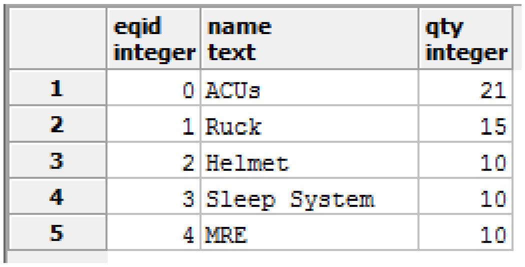
```
CREATE TRIGGER add_to_qty
AFTER DELETE ON equipment_issued
FOR EACH STATEMENT EXECUTE PROCEDURE add_to_qty();
```

EXAMPLE OUTPUT: (The table below is before the delete statement)



	eqid	name	qty
	integer	text	integer
1	0	ACUs	20
2	1	Ruck	15
3	2	Helmet	10
4	3	Sleep System	10
5	4	MRE	10

DELETE FROM equipment_issued WHERE issueno = 0



	eqid	name	qty
	integer	text	integer
1	0	ACUs	21
2	1	Ruck	15
3	2	Helmet	10
4	3	Sleep System	10
5	4	MRE	10

SECURITY

There will be three different types of users for this information system. They include the following:

1. The system administrator, who will have omnipotent control of the database:

   ```
   CREATE ROLE arclis_admin
   GRANT ALL PRIVLIGES ON ALL TABLES IN SCHEMA public TO arclis_admin
   ```

2. The instructors, who will have the ability to modify most tables since they are the main controllers of the system like the cadets make up the core of the system:

   ```
   CREATE ROLE instructor;
   GRANT SELECT ON attendance TO instructor;
   GRANT SELECT, UPDATE, INSERT ON people TO instructor;
   GRANT SELECT, UPDATE, INSERT, DELETE ON cadets TO instructor;
   GRANT SELECT, UPDATE, INSERT, DELETE ON students TO instructor;
   GRANT SELECT, UPDATE, INSERT ON schools TO instructor;
   GRANT SELECT, UPDATE, INSERT, DELETE ON recruits TO instructor;
   GRANT SELECT, UPDATE, INSERT ON semester_gpa TO instructor;
   GRANT SELECT, UPDATE, INSERT, DELETE ON cadre TO instructor;
   GRANT SELECT, UPDATE, INSERT ON events TO instructor;
   GRANT SELECT, UPDATE, INSERT ON events_attended TO instructor;
   GRANT SELECT, UPDATE, INSERT ON rotc_class TO instructor;
   GRANT SELECT, UPDATE, INSERT ON equipment TO instructor;
   GRANT SELECT, UPDATE, INSERT, DELETE ON equipment_issued TO instructor;
   GRANT SELECT, UPDATE, INSERT ON activities TO instructor;
   GRANT SELECT, UPDATE, INSERT ON rotc_activities TO instructor;
   GRANT SELECT, UPDATE, INSERT ON other_activities TO instructor;
   GRANT SELECT, UPDATE, INSERT ON activities_involvement TO instructor;
   GRANT SELECT, UPDATE, INSERT ON awards TO instructor;
   GRANT SELECT, UPDATE, INSERT ON awards_received TO instructor;
   GRANT SELECT, UPDATE, INSERT ON summer_training TO instructor;
   GRANT SELECT, UPDATE, INSERT ON apfts TO instructor;
   ```

3. The cadet leaders or cadet officers, who help manage equipment and attendance:

   ```
   CREATE ROLE cadet_officer
   GRANT SELECT ON attendance TO cadet_officer;
   GRANT SELECT, UPDATE, INSERT ON events TO instructor;
   GRANT SELECT, UPDATE, INSERT ON events_attended TO instructor;
   GRANT SELECT, UPDATE, INSERT ON equipment TO instructor;
   GRANT SELECT, UPDATE, INSERT, DELETE ON equipment_issued TO instructor;
   ```

IMPLEMENTATION NOTES

The whole project went pretty smoothly overall and seems effective for its intent. My ROTC company needs a better way for keeping track of those who are involved in the program in some way, and after talking to my instructor as to what information would be needed to be kept track of, I took on this idea as a way to potentially help my company become more effective at managing its resources. Further, since this is based on the real information needs of my ROTC company, it could be implemented with some training for instructors and cadet officers on how to use SQL, and it could be used by other ROTC companies since their needs are surely similar to the ones covered by this information system.

KNOWN PROBLEMS

- When we conduct APFTs, sometimes cadets are also grader. The database does not have a way to reflect this situation.
- While the instructors should always consider other criteria for awards such as character and attitude, there is no way for the instructor so see what awards cadets qualify for automatically.

FUTURE ENHANCEMENTS

- Have this system work with the much larger database for keeping information on cadets called CCIMS which would make the instructors lives easier.
- Make it possible that cadets can also be graders if needed.
- Automatically generate the report for awards that cadets qualify for based on the criteria of the award.