Introduction

The Center for Community Engagement and Impact(CEI) at VCU deepens and ignites collaboration between organizations focused on community development and the university. These collaborations lead to partnerships, community-engaged research, service-learning classes, and much more to help solve complex challenges in our communities that also embrace educational experiences for all participating stakeholders. Because of this, there is a need to maintain and track these collaborations on ongoing projects that the CEI have had with their external relationships. The purpose for tracking such data comes in three ways:

1. The collection of data centered around community partnerships are vital so that we can accurately see which VCU entities are partnered with which organizations to see if certain schools at VCU have the same partner.(i.e. A lot of researchers partner with Richmond Public Schools).

2. Several departments at VCU struggle to have a way to track partnership data, so this provides a way to conveniently report on this data. Additionally, researchers who want to partner with Richmond Public Schools for example might be able to benefit knowing if other researchers have done the same, and even possibly on a similar research topic. This makes it easier to see a network of who engages with who and to expose some potential unique overlap.

3. Data on community partners will provide the backbone of a proposed single point-of-entry "Front Door" to the university and health system.

The "front door" can be depicted as a business, faculty/staff member of VCU, non-profit community organization or whomever who is looking for assistance, information, or opportunities involving these partnership activity types:

- Programs/initiatives at VCU, public service/volunteerism, research, teaching/service learning, and/or internships. Which would connect to the proposed partnership data source.
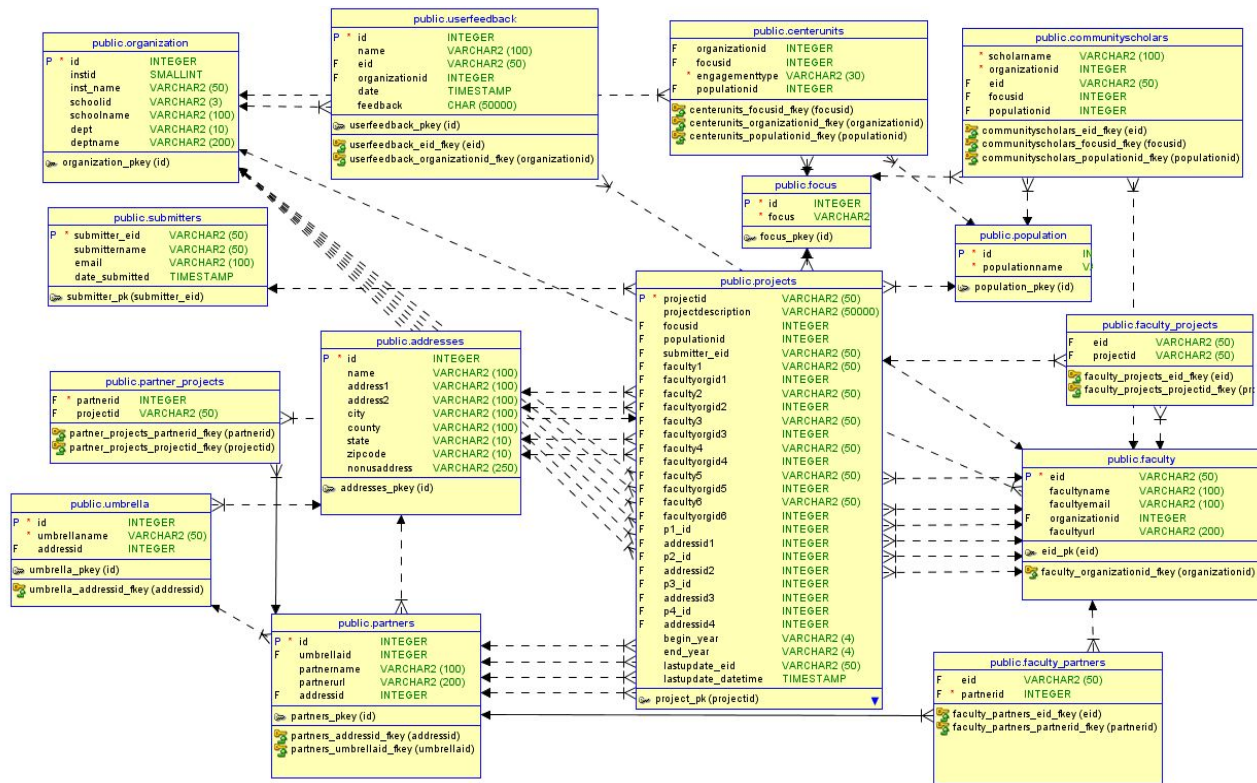
There is a database designed for this already, but it is not fully implemented and it needs to be

re-designed to better align with what we are trying to accomplish. The following list will be what the database track:

- Locations table of community partners

- Community Engaged Contacts table

- Centers and Units table(similar to https://engagement.umn.edu/resources/centers-andunits)

- Up to date info on VCU faculty that are "Community Engaged Scholars" (similar to https://engagement.umn.edu/resources/scholar-directory)

- Community Organizations table

- Project Description table

- Projects ID table

Some of these ideas might end up being edited or changed in the normalization stage, but this gives a brief overview of what needs to be captured. There will be communication between my boss and I for new ideas or issues, but in general the data being tracked will be information about the project, the organizations partnered for the project, their locations, department information, faculty information, if the projects were updated(and when), dates and times of projects, and other things I can think of.

Advanced ERD Diagram

public.organization

P	*	id	INTEGER
		instid	SMALLINT
		inst_name	VARCHAR2 (50)
		schoolid	VARCHAR2 (3)
		schoolname	VARCHAR2 (100)
		dept	VARCHAR2 (10)
		deptname	VARCHAR2 (200)

organization_pkey (id)

public.submitters

P	*	submitter_eid	VARCHAR2 (50)
		submittername	VARCHAR2 (50)
		email	VARCHAR2 (100)
		date_submitted	TIMESTAMP

submitter_pk (submitter_eid)

public.userfeedback

P	*	id	INTEGER
		name	VARCHAR2 (100)
F		eid	VARCHAR2 (50)
F		organizationid	INTEGER
		date	TIMESTAMP
		feedback	CHAR (50000)

userfeedback_pkey (id)
userfeedback_eid_fkey (eid)
userfeedback_organizationid_fkey (organizationid)

public.centerunits

F		organizationid	INTEGER
F		focusid	INTEGER
	*	engagementtype	VARCHAR2 (30)
F		populationid	INTEGER

centerunits_focusid_fkey (focusid)
centerunits_organizationid_fkey (organizationid)
centerunits_populationid_fkey (populationid)

public.communityscholars

	*	scholarname	VARCHAR2 (100)
	*	organizationid	INTEGER
F		eid	VARCHAR2 (50)
F		focusid	INTEGER
F		populationid	INTEGER

communityscholars_eid_fkey (eid)
communityscholars_focusid_fkey (focusid)
communityscholars_populationid_fkey (populationid)

public.focus

P	*	id	INTEGER
	*	focus	VARCHAR2

focus_pkey (id)

public.population

P	*	id	IN
	*	populationname	V

population_pkey (id)

public.addresses

P	*	id	INTEGER
		name	VARCHAR2 (100)
		address1	VARCHAR2 (100)
		address2	VARCHAR2 (100)
		city	VARCHAR2 (100)
		county	VARCHAR2 (100)
		state	VARCHAR2 (10)
		zipcode	VARCHAR2 (10)
		nonusaddress	VARCHAR2 (250)

addresses_pkey (id)

public.partner_projects

F	*	partnerid	INTEGER
F		projectid	VARCHAR2 (50)

partner_projects_partnerid_fkey (partnerid)
partner_projects_projectid_fkey (projectid)

public.umbrella

P	*	id	INTEGER
	*	umbrellaname	VARCHAR2 (50)
F		addressid	INTEGER

umbrella_pkey (id)
umbrella_addressid_fkey (addressid)

public.partners

P	*	id	INTEGER
F		umbrellaid	INTEGER
		partnername	VARCHAR2 (100)
		partnerurl	VARCHAR2 (200)
F		addressid	INTEGER

partners_pkey (id)
partners_addressid_fkey (addressid)
partners_umbrellaid_fkey (umbrellaid)

public.projects

P	*	projectid	VARCHAR2 (50)
		projectdescription	VARCHAR2 (50000)
F		focusid	INTEGER
F		populationid	INTEGER
F		submitter_eid	VARCHAR2 (50)
F		faculty1	VARCHAR2 (50)
F		facultyorgid1	INTEGER
F		faculty2	VARCHAR2 (50)
F		facultyorgid2	INTEGER
F		faculty3	VARCHAR2 (50)
F		facultyorgid3	INTEGER
F		faculty4	VARCHAR2 (50)
F		facultyorgid4	INTEGER
F		faculty5	VARCHAR2 (50)
F		facultyorgid5	INTEGER
F		faculty6	VARCHAR2 (50)
F		facultyorgid6	INTEGER
F		p1_id	INTEGER
F		addressid1	INTEGER
F		p2_id	INTEGER
F		addressid2	INTEGER
F		p3_id	INTEGER
F		addressid3	INTEGER
F		p4_id	INTEGER
F		addressid4	INTEGER
		begin_year	VARCHAR2 (4)
		end_year	VARCHAR2 (4)
		lastupdate_eid	VARCHAR2 (50)
		lastupdate_datetime	TIMESTAMP

project_pk (projectid)

public.faculty_projects

F		eid	VARCHAR2 (50)
F		projectid	VARCHAR2 (50)

faculty_projects_eid_fkey (eid)
faculty_projects_projectid_fkey (pr

public.faculty

P	*	eid	VARCHAR2 (50)
		facultyname	VARCHAR2 (100)
		facultyemail	VARCHAR2 (100)
F		organizationid	INTEGER
		facultyurl	VARCHAR2 (200)

eid_pk (eid)
faculty_organizationid_fkey (organizationid)

public.faculty_partners

F		eid	VARCHAR2 (50)
F	*	partnerid	INTEGER

faculty_partners_eid_fkey (eid)
faculty_partners_partnerid_fkey (partnerid)

SQL Statements

CREATE Table Statements

```
DROP TABLE IF EXISTS
addresses,projects,submitters,focus,population,organization,partners,umbrel
la,faculty, centerUnits, communityScholars, projects,userFeedback
,partner_projects, faculty_projects, faculty_partners CASCADE
;
CREATE TABLE submitters(
      submitter_eID varchar(50) NULL,
      CONSTRAINT submitter_pk PRIMARY KEY (submitter_eID),
      submitterName varchar(50) NULL,
      email varchar(100) NULL,
      date_submitted timestamp NULL
    );
```

```sql
CREATE TABLE focus(
    id serial PRIMARY KEY,
    focus varchar(100) NOT NULL
);

CREATE TABLE population(
    id serial PRIMARY KEY,
    populationName varchar(250) NOT NULL
);

CREATE TABLE organization(
    id serial PRIMARY KEY,
    InstID smallint NULL,
    Inst_Name varchar(50) NULL,
    schoolID varchar(3) NULL,
    schoolName varchar(100) NULL,
    Dept varchar(10) NULL,
    deptName varchar(200) NULL
);

CREATE TABLE addresses (
    id serial PRIMARY KEY,
    Name varchar(100) NULL,
    Address1 varchar(100) NULL,
    Address2 varchar(100) NULL,
    City varchar(100) NULL,
    County varchar(100) NULL,
    State  varchar(10) NULL,
    ZipCode varchar(10) NULL,
    NonUSAddress  varchar(250) NULL
);

CREATE TABLE umbrella(
    id serial PRIMARY KEY,
    umbrellaName varchar(50) NOT NULL,
    addressID int REFERENCES addresses(id)
);

CREATE TABLE partners(
    id serial PRIMARY KEY,
    umbrellaID int REFERENCES umbrella(id),
    partnerName varchar(100) NULL,
```

```sql
        partnerURL varchar(200) NULL,
        addressID int REFERENCES addresses(id)
    );

CREATE TABLE faculty(
        eID varchar(50) NULL,
        CONSTRAINT eID_pk PRIMARY KEY (eID),
        facultyName varchar(100) NULL,
        facultyEmail varchar(100) NULL,
        organizationID int REFERENCES organization(id),
        facultyURL varchar(200) NULL
    );

CREATE TABLE centerUnits(
        organizationID int REFERENCES organization(id),
        focusID int REFERENCES focus(id),
        engagementType varchar(30) NOT NULL,
        populationID int REFERENCES population(id)
    );

CREATE TABLE communityScholars(
        scholarName varchar(100) NOT NULL,
        organizationID int NOT NULL,
        eID varchar(50) REFERENCES faculty(eID),
        focusID int REFERENCES focus(id),
        populationID int REFERENCES population(id)
    );

CREATE TABLE userFeedback(
        id serial PRIMARY KEY,
        name varchar(100) NULL,
        eID varchar(50) REFERENCES faculty(eID),
        organizationID int REFERENCES organization(id),
        date timestamp NULL,
        feedback char(50000) NULL
    );

CREATE TABLE projects(
        projectID varchar(50)  NOT NULL,
        projectDescription varchar(50000) NULL,
        focusID int REFERENCES focus(id),
        populationID int REFERENCES population(id),
        CONSTRAINT project_pk PRIMARY KEY (projectID),
```

```sql
        submitter_eID varchar(50) REFERENCES submitters(submitter_eID),
        Faculty1 varchar(50) REFERENCES faculty(eID),
        FacultyOrgID1 int REFERENCES organization(id),
        Faculty2 varchar(50) REFERENCES faculty(eID),
        FacultyOrgID2 int REFERENCES organization(id),
        Faculty3 varchar(50) REFERENCES faculty(eID)NULL,
        FacultyOrgID3 int REFERENCES organization(id)NULL,
        Faculty4 varchar(50) REFERENCES faculty(eID)NULL,
        FacultyOrgID4 int REFERENCES organization(id)NULL,
        Faculty5 varchar(50) REFERENCES faculty(eID)NULL,
        FacultyOrgID5 int REFERENCES organization(id)NULL,
        Faculty6 varchar(50) REFERENCES faculty(eID)NULL,
        FacultyOrgID6 int REFERENCES organization(id)NULL,
        P1_ID int REFERENCES partners(id),
        addressID1 int REFERENCES addresses(id),
        P2_ID int REFERENCES partners(id),
        addressID2 int REFERENCES addresses(id),
        P3_ID int REFERENCES partners(id),
        addressID3 int REFERENCES addresses(id),
        P4_ID int REFERENCES partners(id),
        addressID4 int REFERENCES addresses(id),
        Begin_Year varchar(4) NULL,
        End_Year varchar(4) NULL,
        LastUpdate_eID varchar(50) NULL,
        LastUpdate_DateTime timestamp NULL
    );

CREATE TABLE partner_projects(
        partnerID serial REFERENCES partners(id),
        projectID varchar(50)  REFERENCES projects(projectID)
    );

CREATE TABLE faculty_projects(
        eID varchar(50) REFERENCES faculty(eid),
        projectID varchar(50)  REFERENCES projects(projectID)
    );

CREATE TABLE faculty_partners(
        eID varchar(50) REFERENCES faculty(eid),
        partnerID serial  REFERENCES partners(id)
    );
```

INSERT Table Statements

```sql
INSERT INTO submitters
(submitter_eID, submitterName, email, date_submitted)
VALUES ('gregbr', 'greg', 'greg@vcu.edu', '03/01/2012'),
('katherinehr', 'katherine', 'katherine@vcu.edu', '03/01/2012'),
('davegh', 'dave', 'dave@vcu.edu', '03/01/2012');

INSERT INTO focus
(id, focus)
VALUES (100, 'Economic and Workforce Development'),
(101, 'Education: Adult Education'),
(102, 'Health and Wellness: Unspecified/General');

INSERT INTO population
(id, populationName)
VALUES (100, 'Inner City'),
(101, 'Ages 65+'),
(102, 'Youth');

INSERT INTO organization
(id, InstID, Inst_Name, schoolID, schoolName, Dept, deptName)
VALUES (100, 100,'VCU Monroe', 100, 'School of Business', 100, 'Economics
Deparment'),
(101, 100,'VCU Monroe',101, 'School of Education', 101,'Adult Education
Department'),
(102, 100,'VCU Monroe',102, 'School of Health Sciences',102,'Exercise
Science Department');

INSERT INTO addresses
(id,Name, Address1, Address2, City, County,State, ZipCode, NonUSAddress)
VALUES (100, 'Thomas Jefferson High School','1212 yeah rd.','3131 yeah
rd.','richmond','Richmond City','VA','23220','NA'),
(101,'Other RPS school','3232 partner2 rd', 'NA','richmond','Richmond
City','VA','23220','NA'),
(102, 'Geriatrics org', '432 yeah rd.','4234 yeah
rd.','richmond','henrico','VA','23220','NA'),
(103,'Another geriatrics org','3232 partner2 rd',
'NA','richmond','henrico','VA','23220','NA'),
(104,'YMCA tuckahoe','209 yeah rd.','0243 yeah
rd.','richmond','henrico','VA','23220','NA'),
(105,'YMCA Glen Allen','4545 partner2
rd','NA','richmond','henrico','VA','23220','NA');
```

```sql
INSERT INTO umbrella
(id, umbrellaName, addressID)
VALUES (100, 'Richmond Public Schools', NULL),
(101, 'Geriatrics Organizations', NULL),
(102, 'Outreach Organizations', NULL);

INSERT INTO partners
(id, umbrellaID,partnerName, partnerURL, addressID)
VALUES (100, 100, 'Thomas Jefferson High School', 'jeffHCPS.com', 100),
(101,100,'Other RPS school', 'otherhcps.com',101),
(102, 101,'Geriatrics org', 'geria.com', 102),
(103,101, 'Another geriatrics org', 'othergeria.com',103),
(104, 102, 'YMCA tuckahoe', 'tuckYMCA.com', 104),
(105,102,'YMCA Glen Allen', 'YMCAGA.com',105);

INSERT INTO faculty
(eid, facultyname,facultyemail, organizationid, facultyurl)
VALUES ('gregbr','greg burtch','greg@vcu.edu',100,'greg.com'),
('katherinehr',   'katherine hines','katherine@vcu.edu', 101
,'SOEVCU.com'),
('davegh',  'dave gregory','dave@vcu.edu',      102,  'dave.com'),
('bobcv','bob cash','bob@vcu.edu',100,'bob.com'),
('timlb','tim lions','tim@vcu.edu', 101   ,'SOEVCU.com'),
('sarahty', 'sarah hamburger','sarah@vcu.edu',101,'SOEVCU.com');

INSERT INTO centerUnits
(organizationID,focusID, engagementType,populationID)
VALUES(100, 102, 'Interships', 101),
(101, 100, 'Health', 102),
(102, 101, 'Service Learning', 100),
(100, 100, 'Research', 101),
(101, 101, 'Service/Outreach', 100),
(102, 102, 'Health-related Services', 102),
(100, 102, 'Research', 101),
(101, 100, 'Research', 102),
(102, 101, 'Research', 100),
(100, 100, 'Research', 101),
(101, 101, 'Service/Outreach', 100),
(102, 102, 'Health-related Services', 102);
```

```sql
INSERT INTO communityScholars
(scholarName,organizationID, eID, focusID , populationID)
VALUES ('Greg B.',100, 'gregbr', 100, 100),
('Katherine H.',101, 'katherinehr', 101, 101),
('Dave G.',102, 'davegh', 102, 102),
('Bob C.',100, 'bobcv', 100, 100),
('Tim L.',101, 'timlb', 101, 101),
('Sarah H.',101, 'sarahty', 102, 102);

INSERT INTO userFeedback
(id, name, eID, organizationID, date, feedback)
VALUES (102,'Dave G.', 'davegh', 102, '01/23/2020', 'I love the database
but my name is spelled wrong'),
(100,'Greg B.', 'gregbr', 100, '05/16/2013', 'If you do not have any
partnerships, or you do not have any changes to make...'),
(101,'Katherine H.', 'katherinehr', 101, '10/15/2014', 'I have included all
my information, however I see no "update" or "save" button so unsure if
this informati...');

INSERT INTO projects
(projectID,  projectDescription,  focusID, populationID,
submitter_eID,Faculty1,FacultyOrgID1,Faculty2,FacultyOrgID2,Faculty3,Facult
yOrgID3,Faculty4,
FacultyOrgID4,Faculty5,FacultyOrgID5,Faculty6,FacultyOrgID6,P1_ID,addressID
1,P2_ID,addressID2,P3_ID,addressID3,P4_ID,addressID4,
Begin_Year, End_Year, LastUpdate_eID, LastUpdate_DateTime)
VALUES ('HM100','We plan on helping economic dev...',
100,100,'gregbr','gregbr',100,'bobcv',100,
NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,100,100,101,101,104,104,NULL,NULL,'
2012','2013','NA',NULL),
('HM101','We plan on helping older...',
101,101,'katherinehr','katherinehr',101,'timlb',101,
NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,102,102,103,103,NULL,NULL,NULL,NULL
,'2012','2013','NA',NULL),
('HM102','We plan on helping youth...',
102,102,'davegh','davegh',102,'sarahty',101,
NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,104,104,105,105,101,101,NULL,NULL,'
2012','2013','NA',NULL);

INSERT INTO partner_projects
(partnerID, projectID)
VALUES(100,'HM100'),
(101,'HM100'),
```

```
(102,'HM101'),
(103,'HM101'),
(104,'HM102'),
(105,'HM102'),
(104,'HM100'),
(101,'HM102');

INSERT INTO faculty_projects
(eID, projectID)
VALUES('gregbr','HM100'),
('katherinehr','HM101'),
('davegh','HM102'),
('bobcv','HM100'),
('timlb','HM101'),
('sarahty','HM102');

INSERT INTO faculty_partners
(eID, partnerID)
VALUES('gregbr',100),
('gregbr',101),
('gregbr',104),
('katherinehr',102),
('katherinehr',103),
('davegh',104),
('davegh',105),
('davegh',101),
('bobcv',100),
('bobcv',101),
('bobcv',104),
('timlb',102),
('timlb',103),
('sarahty',104),
('sarahty',105),
('sarahty',101);
```

SELECT Table Statements

```
'''I wanted to show that a department can have multiple engagement types
across multiple focuses and populations, it is select distinct because I
may have duplicated inserting valueS'''
SELECT DISTINCT
organization.inst_name,organization.schoolname,organization.deptname,
centerUnits.engagementType, focus.focus, population.populationName FROM
organization, centerUnits
INNER JOIN focus ON focus.id = centerUnits.focusID
INNER JOIN population ON population.id = centerUnits.populationID;


'''Researchers with their area of research'''
SELECT
communityScholars.scholarname,organization.schoolname,organization.deptname
,focus.focus, population.populationName FROM  communityScholars
INNER JOIN focus ON focus.id = communityScholars.focusID
INNER JOIN organization ON organization.id =
communityScholars.organizationID
INNER JOIN population ON population.id = communityScholars.populationID;


'''partners with their addresses'''
SELECT partners.id,partners.partnername,partners.partnerurl,
umbrella.umbrellaName,addresses.Address1,addresses.County, addresses.state
FROM  partners
INNER JOIN umbrella ON umbrella.id = partners.umbrellaID
INNER JOIN addresses ON addresses.id = partners.addressID;


'''partners with their corresponding project'''
SELECT partner_projects.projectid, partners.partnername,
umbrella.umbrellaname, partners.partnerurl,projects.projectdescription from
partner_projects
INNER JOIN partners ON partners.id= partner_projects.partnerid
INNER JOIN projects on projects.projectid  = partner_projects.projectid
INNER JOIN umbrella ON umbrella.id = partners.umbrellaid
ORDER BY partners.partnername;


'''shows faculty and what project they did'''
SELECT
faculty.facultyname,faculty.facultyemail,faculty_projects.projectid,
projects.projectDescription FROM  faculty_projects
INNER JOIN faculty ON faculty.eid = faculty_projects.eid
INNER JOIN projects ON projects.projectid = faculty_projects.projectid
```

```
ORDER BY faculty_projects.projectid;

'''faculty and who they partner with '''
SELECT  faculty.facultyname,faculty.facultyemail,partners.partnername,
partners.partnerurl from faculty_partners
INNER JOIN faculty ON faculty.eid = faculty_partners.eid
INNER JOIN partners on partners.id = faculty_partners.partnerid
ORDER BY faculty.facultyname;

User Feedback on databases with their org. info
SELECT userFeedback.name, userFeedback.eid, organization.inst_name,
organization.schoolname, organization.deptname, userfeedback.date,
userFeedback.feedback FROM  organization
INNER JOIN userFeedback on userFeedback.organizationid=organization.id;

'''Searching for researchers who focus in economic and workforce dev'''
SELECT
communityScholars.scholarname,organization.schoolname,organization.deptname
,focus.focus, population.populationName FROM  communityScholars
INNER JOIN focus ON focus.id = communityScholars.focusID
INNER JOIN organization ON organization.id =
communityScholars.organizationID
INNER JOIN population ON population.id = communityScholars.populationID
WHERE focus.focus = 'Economic and Workforce Development';

'''Counting by county and grouping them'''
Select addresses.county, count(addresses.county) from addresses
GROUP BY addresses.county;

'''Counting number of times a partner appeared in a project'''
Select partners.partnername,count(partner_projects.projectid)from
partner_projects
left JOIN partners on partner_projects.partnerid = partners.id
group by partners.partnername
ORDER BY count(partner_projects.projectid);

'''number of faculty on a project'''
Select projects.projectID, projects.projectdescription,
count(faculty_projects.eid)as numberoffaculty from faculty_projects
left join projects on faculty_projects.projectid = projects.projectid
group by projects.projectID;

'''Number of partners for every project'''
```

```
Select projects.projectid,count(partner_projects.partnerid)from
partner_projects
left JOIN projects on partner_projects.projectid = projects.projectid
group by  projects.projectid
ORDER BY count(partner_projects.partnerid);

'''Showing partner who have appeared in more than 1 project'''
Select partners.partnername,count(partner_projects.projectid)from
partner_projects
left JOIN partners on partner_projects.partnerid = partners.id
group by partners.partnername
having count(partner_projects.projectid) >1;

'''Showing projects that have more than 2 community partners'''
Select projects.projectid,count(partner_projects.partnerid)from
partner_projects
left JOIN projects on partner_projects.projectid = projects.projectid
group by  projects.projectid
HAVING COUNT(partner_projects.partnerid) > 2;

'''Selecting Schools that focus their engagement to anything related to the
word health'''
SELECT DISTINCT
organization.inst_name,organization.schoolname,organization.deptname,
centerUnits.engagementType, focus.focus, population.populationName FROM
organization, centerUnits
INNER JOIN focus ON focus.id = centerUnits.focusID
INNER JOIN population ON population.id = centerUnits.populationID
where  centerUnits.engagementtype like '%Health%';


SELECT *FROM projects;
```