Kaiser Permanente Downey

IS380 Group: Minh Ton, Kelley Vo, Agnes Chang, Alexis Zavaleta, Justin Somsookh

General Description

Kaiser Permanente is one of the nation's largest not-for-profit health plans serving approximately 12.4 million members. Kaiser owns 39 hospitals and in 8 states and the District of Columbia. Kaisers mission state is to offer high-quality, affordable health care services that will help improve the health of communities. A few services that Kaiser Permanente offers include pediatric services, obstetrics, gynecology, and of course primary care. Kaiser continues to build clinics in communities that offer all these great services. Recently, Kaiser has been using technology to simplify the experience of members by redesigning mobile applications and even using video conferences where a patient can video chat with their doctor. Another significant use of technology that Kaiser has begun to implement to better the experience is the use of wearables so a patient's doctor can monitor them in real time. Kaiser is always trying to create new ways in which they can simplify the overall experience of a Kaiser Permanente member. With the current Covid-19 pandemic happening Kaiser has adapted and has been able to continue with day-to-day operations why till being able to provide Covid-19 testing to patients. Recently Kaiser is in the process of opening a new hospital tower in the city of Downey which not only offers medical services but also provides families with the necessary resources to live a healthy life. At the moment the new medical center in the city of Downey offers Covid-19 testing and Flu shots. Kaiser Permanente is also offering programs and classes that are even designed to help individuals with emotions or even addiction.

With this project Kaiser is planning on opening a new tower. Before they can open, they need to staff each department for the new tower. In addition to staffing, staff needs to fill out forms and go through training. Kaiser needs a way to track the progress of each department and staff through the orientation and form filling process.

User Requirements

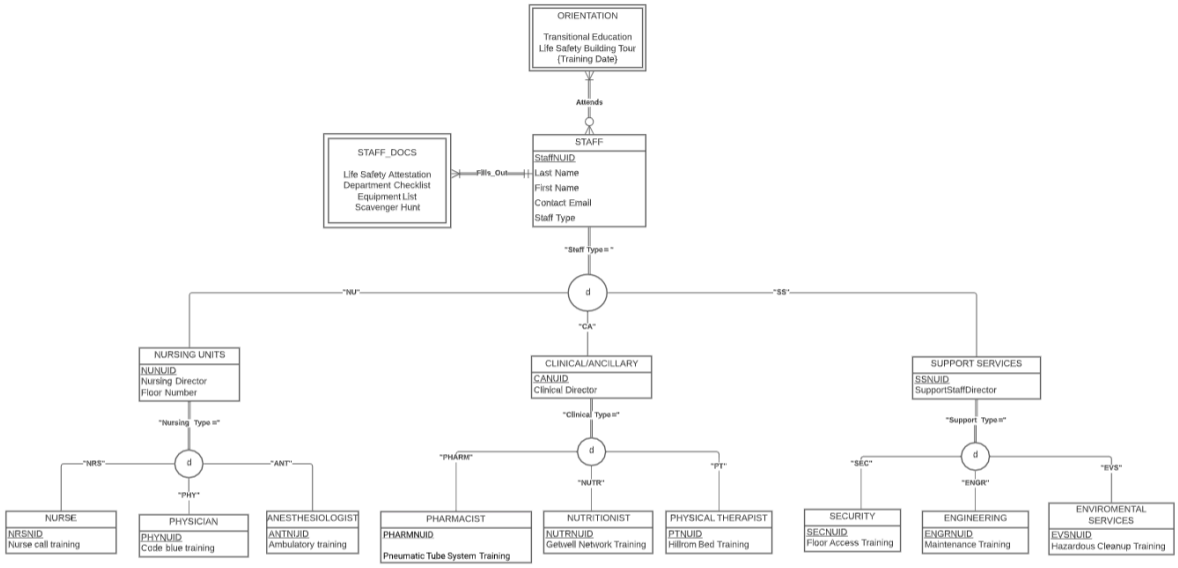
In order for our database to be able to manage the increase in data coming from multiple sources such as the new Kaiser Permanente facility in Downey there is a set of requirements that must be met and implemented to function properly. The requirements listed ahead will be essential for our database be able to store complex relationships in data, yet maintaining an organized structure that maintains the integrity of the data.

- Staff are identified by their NUID with attributes such as Last Name, First Name and staff type.
    - Staff is separated by type.
    - Staff in Nursing Unit, Clinical/Ancillary, or in Support Services.
    - Nursing units contain the following attributes: A unique NUNUID, Nursing Director, and Floor number. Nursing Units Staff can only be Nurses, Physicians, or Anesthesiologist.
    - Nurse staff type contains the attributes NURSNID and Nurse Call Training, while physician entity contains PHYNUID and Code Blue Training, and Anesthesiologist entity contains ANTNUID and Ambulatory Training.

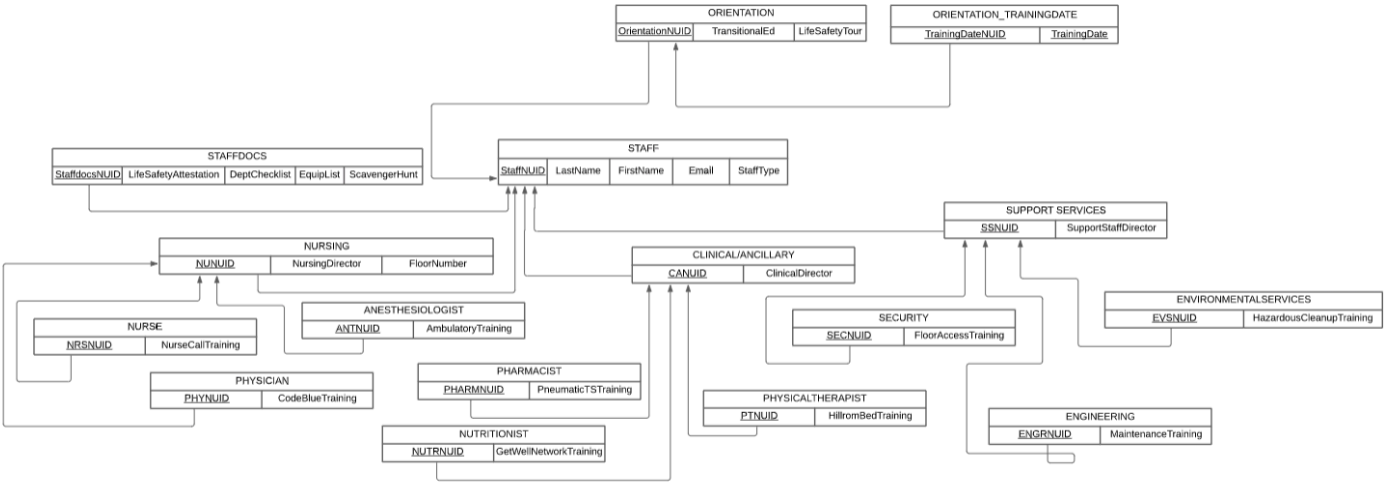
- Support Services contains the following attributes: SSNUID and Support Staff Director. Support Service Staff Type are limited to: Security, Engineering, and Environmental Services.
- The Security entity contains a SECNUID and Floor Access Training, while the Engineering entity contains a ENGRNUID and Maintenance Training and Environmental Services has a EVNSNUID and Hazardous Cleanup Training.
- Clinical/Ancillary contains the following attribute: CANUID and Clinical Director. Clinical Ancillary staff type is limited to: Pharmacist, Nutritionist, and Physical Therapist each possessing their own unique ID.
- Pharmacist contains the attributes PHARMNUID and Pneumatic Tube System Training.
- Nutritionist contain the attributes NUTRNUID as well as Getwell Network Training.
- Physical Therapist contains the attributes PTNUID and Hillrom Training.

- The required documents for each staff include the Life Safety Attestation, the Department Checklist, the Equipment List, and the Scavenger Hunt.
- Staff members can have one or many of the required documents, but each required document can only belong to one staff member.
- Orientation days include the Transition Life/Safety Building Tour, and training date for orientation. Orientations can include zero or many staff, but staff must attend at least one orientation.

Entity-Relationship Diagram



Relational Data Model



Implementation

With the implementation of MySQL data can stay organized, which will allow us to run queries to perform specific tasks with the data in the database. Such as finding employees who are currently in training and when exactly is their training date.

```
DELIMITER //

CREATE PROCEDURE sp_secnuid_td ()

BEGIN

SELECT staff.StaffFirstName, staff.StaffLastName,orientation_trainingdate.TrainingDate FROM staff, orientation, orientation_trainingdate

WHERE staff.StaffNUID = orientation.OrientationNUID AND orientation.OrientationNUID = orientation_trainingdate.TrainingDateNUID;

END //

DELIMITER ;
```

+ Options

StaffFirstName	StaffLastName	TrainingDate
Luis	Smith	2020-09-26 00:00:00
Humera	Friedman	2020-09-26 00:00:00
Humera	Robert	2020-09-28 00:00:00
Cherie	Wilkes	2020-10-15 00:00:00
Lake	Robyn	2020-10-16 00:00:00
Taliyah	Whittington	2020-10-16 00:00:00
Reiley	Rojin	2020-10-23 00:00:00
Ortega	Sahib	2020-10-23 00:00:00
Thatcher	Arya	2020-10-24 00:00:00
Ayers	Kadie	2020-10-24 00:00:00
Natasha	Maxwell	2020-09-26 00:00:00
Faizaan	Aguirre	2020-09-26 00:00:00
Jones	Aditya	2020-09-28 00:00:00
Carter	Lucas	2020-10-15 00:00:00
Velazquez	Louise	2020-10-16 00:00:00

We are also able to track what important documents are completed and what needs to be done, while still demonstrating the information of who has completed the forms and who needs forms to complete.

```
DELIMITER //

CREATE PROCEDURE req_forms (x varchar(50))

BEGIN

SELECT staff.StaffType, staff.StaffLastName, staff.StaffFirstName, staff_docs.LifeSafetyAttestation,
staff_docs.DeptChecklist, staff_docs.EquipList FROM staff, staff_docs

WHERE staff.StaffNUID = staff_docs.StaffdocsNUID AND staff.StaffType = x;

END //
```

```
SET @p0='pharmacist'; CALL `req_forms`(@p0);
```

Execution results of routine `req_forms`

StaffType	StaffLastName	StaffFirstName	LifeSafetyAttestation	DeptChecklist	EquipList
Pharmacist	Holman	Harrison	No	Yes	No
Pharmacist	Sarina	Whitley	Yes	No	Yes
Pharmacist	Konrad	Mayer	Yes	Yes	No

Another example of implementation would be the ability to keep track of the floors that nurses are available as well as their coordinating staff type. Also it allows us to see which nurses are taking a training.

```
DELIMITER //
CREATE PROCEDURE sp_nursing_floor ()

BEGIN

SELECT staff.StaffNUID, staff.StaffFirstName, nursing.NursingDirector, nursing.FloorNumber,
nurse.NurseCallTraining FROM staff, nurse, nursing
WHERE
staff.StaffNUID = nursing.NUNUID AND
nursing.NUNUID = nurse.NRSNUID;

END //
```

+ Options

StaffNUID	StaffFirstName	NursingDirector	FloorNumber	NurseCallTraining
2001	Harrison	Holman Harrison	6N	No
2002	Whitley	Sarina Whitley	4N	No
2003	Mayer	Holman Harrison	5N	Yes
2004	Washington	Sarina Whitley	4N	No
2005	Grainger	Holman Harrison	5N	Yes

Views with SQL

SQL allows us to use views to specify searches based off employee criteria such as first name, last name, or NUID. We can also use complex statements to track staff document completion

```
CREATE VIEW V_tdocs_staff AS


(SELECT staff.StaffType, staff.StaffFirstName, staff.StaffLastName, staff_docs.DeptChecklist FROM staff, staff_docs

WHERE staff.StaffNUID = staff_docs.StaffdocsNUID AND staff_docs.DeptChecklist = "No"

);
```

+ Options

StaffType	StaffFirstName	StaffLastName	DeptChecklist
Engineering	Humera	Robert	No
Pharmacist	Whitley	Sarina	No
Nurse	Garza	Adina	No
Security	Jones	Aditya	No

Conclusion

Using MySQL allows us to keep data organized which makes it easier to keep track of all the new employees. Using view and stored procedures allows us to see who is on track with the

trainings, which in the long run can be beneficial since it would increase efficiency in bringing up to date all new and existing employees.

SQL Commands for Creating Tables

Staff

```
CREATE TABLE staff
(
StaffNUID int(11) NOT NULL AUTO_INCREMENT,
StaffLastName varchar(50),
StaffFirstName varchar(50),
StaffEmail varchar(50),
StaffType varchar(50),
CONSTRAINT staff_pk PRIMARY KEY (StaffNUID)
)
ENGINE=INNODB;
```

Anesthesiologist

```
CREATE TABLE anesthesiologist
(
ANTNUID int(11) NOT NULL,
AmbulatoryTraining varchar(50),
CONSTRAINT anesthesiologist_pk PRIMARY KEY (ANTNUID),
CONSTRAINT anesthesiologist_fk FOREIGN KEY (ANTNUID) REFERENCES nursing(NUNUID) ON UPDATE CASCADE
)
ENGINE=INNODB;
```

Clinical

```
CREATE TABLE clinical_ancillary
(
CANUID int(11) NOT NULL,
ClinicalDirector varchar(50),
CONSTRAINT clinical_ancillary_pk PRIMARY KEY (CANUID),
CONSTRAINT clinical_ancillary_fk FOREIGN KEY (CANUID) REFERENCES staff(StaffNUID) ON UPDATE CASCADE
)
ENGINE=INNODB;
```

Engineering

```
CREATE TABLE engineering
(
ENGRNUID int(11) NOT NULL,
MaintenanceTraining varchar(50),
CONSTRAINT engineering_pk PRIMARY KEY (ENGRNUID),
CONSTRAINT engineering_fk FOREIGN KEY (ENGRNUID) REFERENCES support_services(SSNUID) ON UPDATE CASCADE
)
ENGINE=INNODB;
```

Environmental Services

```
CREATE TABLE environmental_services
(
EVSNUID int(11) NOT NULL,
HazardousCleanupTraining varchar(50),
CONSTRAINT environmental_services_pk PRIMARY KEY (EVSNUID),
CONSTRAINT environmental_services_fk FOREIGN KEY (EVSNUID) REFERENCES support_services(SSNUID) ON UPDATE CASCADE
)
ENGINE=INNODB;
```

Nurse

```sql
CREATE TABLE nurse
(
NRSNUID int(11) NOT NULL,
NurseCallTraining varchar(50),
CONSTRAINT nurse_pk PRIMARY KEY (NRSNUID),
CONSTRAINT nurse_fk FOREIGN KEY (NRSNUID) REFERENCES nursing(NUNUID) ON UPDATE CASCADE
)
ENGINE=INNODB;
```

Nursing

```sql
CREATE TABLE nursing
(
NUNUID int(11) NOT NULL,
NursingDirector varchar(50),
FloorNumber varchar(50),
CONSTRAINT nursing_pk PRIMARY KEY (NUNUID),
CONSTRAINT nursing_fk FOREIGN KEY (NUNUID) REFERENCES staff(StaffNUID) ON UPDATE CASCADE
)
ENGINE=INNODB;
```

Nutritionist

```sql
CREATE TABLE nutritionist
(
NUTRNUID int(11) NOT NULL,
GetWellNetworkTraining varchar(50),
CONSTRAINT nutritionist_pk PRIMARY KEY (NUTRNUID),
CONSTRAINT nutritionist_fk FOREIGN KEY (NUTRNUID) REFERENCES clinical_ancillary(CANUID) ON UPDATE CASCADE
)
ENGINE=INNODB;
```

Orientation Training Dates

```sql
CREATE TABLE orientation_trainingdate
(
TrainingdateNUID int(11) NOT NULL,
TrainingDate DATETIME,
CONSTRAINT orientation_trainingdate_pk PRIMARY KEY (TrainingdateNUID, TrainingDate),
CONSTRAINT orientation_trainingdate_fk FOREIGN KEY (TrainingdateNUID) REFERENCES orientation(OrientationNUID) ON UPDATE CASCADE
)
ENGINE=INNODB;
```

Orientation

```sql
CREATE TABLE orientation
(
OrientationNUID int(11) NOT NULL,
TransitionalEd varchar(50),
LifeSafetyTour varchar(50),
CONSTRAINT orientation_pk PRIMARY KEY (OrientationNUID),
CONSTRAINT orientation_fk FOREIGN KEY (OrientationNUID) REFERENCES staff(StaffNUID) ON UPDATE CASCADE
)
ENGINE=INNODB;
```

Pharmacist

```
CREATE TABLE pharmacist
(
PHARMNUID int(11) NOT NULL,
PneumaticTSTraining varchar(50),
CONSTRAINT pharmacist_pk PRIMARY KEY (PHARMNUID),
CONSTRAINT pharmacist_fk FOREIGN KEY (PHARMNUID) REFERENCES clinical_ancillary(CANUID) ON UPDATE CASCADE
)
ENGINE=INNODB;
```

Physical Therapist

```
CREATE TABLE physical_therapist
(
PTNUID int(11) NOT NULL,
HillromBedTraining varchar(50),
CONSTRAINT physical_therapist_pk PRIMARY KEY (PTNUID),
CONSTRAINT physical_therapist_fk FOREIGN KEY (PTNUID) REFERENCES clinical_ancillary(CANUID) ON UPDATE CASCADE
)
ENGINE=INNODB;
```

Physician

```
CREATE TABLE physician
(
PHYNUID int(11) NOT NULL,
CodeBlueTraining varchar(50),
CONSTRAINT physician_pk PRIMARY KEY (PHYNUID),
CONSTRAINT physician_fk FOREIGN KEY (PHYNUID) REFERENCES nursing(NUNUID) ON UPDATE CASCADE
)
ENGINE=INNODB;
```

Security

```
CREATE TABLE security
(
SECNUID int(11) NOT NULL,
FloorAccessTraining varchar(50),
CONSTRAINT security_pk PRIMARY KEY (SECNUID),
CONSTRAINT security_fk FOREIGN KEY (SECNUID) REFERENCES support_services(SSNUID) ON UPDATE CASCADE
)
ENGINE=INNODB;
```

Staff Docs

```
CREATE TABLE staff_docs
(
StaffdocsNUID int(11) NOT NULL,
LifeSafetyAttestation varchar(50),
DeptChecklist varchar(50),
EquipList varchar(50),
ScavengerHunt varchar(50),
CONSTRAINT staffdocs_pk PRIMARY KEY (StaffdocsNUID),
CONSTRAINT staffdocs_fk FOREIGN KEY (StaffdocsNUID) REFERENCES staff(StaffNUID) ON UPDATE CASCADE
)
ENGINE=INNODB;

CREATE TABLE orientation
```

Support Services

```sql
CREATE TABLE support_services
(
SSNUID int(11) NOT NULL,
SupportStaffDirector varchar(50),
CONSTRAINT support_services_pk PRIMARY KEY (SSNUID),
CONSTRAINT support_services_fk FOREIGN KEY (SSNUID) REFERENCES staff(StaffNUID) ON UPDATE CASCADE
)
ENGINE=INNODB;
```

SQL Commands for Inserting Data into Tables

Inserting to Staff table

```sql
INSERT INTO staff(StaffNUID,StaffLastName,StaffFirstName,StaffEmail,StaffT
ype)
VALUES
(1001,'Smith','Luis','Smith.Luis@gmail.com','EVS'),
(1002,'Friedman','Humera','Humera.Friedman@gmail.com','Engineering'),
(1003,'Robert','Humera','Robert.Friedman@gmail.com','Engineering'),
(1004,'Wilkes','Cherie','Cherie.Wilkes@gmail.com','Physical Therapist'),
(1005,'Robyn','Lake','Robyn.Lake@gmail.com','Nutritionist'),
(1006,'Whittington','Taliyah','Taliyah.Whittington@gmail.com','Anesthesiol
ogist'),
(1007,'Rojin','Reiley','Sahib.Ortega@kp.org','EVS'),
(1008,'Sahib','Ortega','Reiley.Rojin@yahoo.com',' Nutritionist '),
(1009,'Arya','Thatcher','Arya.Thatcher@kp.org','Physician'),
(1010,'Kadie','Ayers','Kadie.Ayers@kp.org','Physician'),
(2001,'Holman','Harrison','Harrison.Holman@gmail.com','Pharmacist'),
(2002,'Sarina','Whitley','Sarina.Whitley@kp.org','Pharmacist'),
(2003,'Konrad','Mayer','Konrad.Mayer@kp.org','Pharmacist'),
(2004,'Aqib','Washington','Aqib.Washington@kp.org','Pharmacist'),
(2005,'Kendall','Grainger','Kendall.Grainger@kp.org','Pharmacist'),
(3001,'Sabah','Lewis','Sabah.Lewis@gmail.com','Nurse'),
(3002,'Kaydee','Dolan','Kaydee.Dolan@kp.org','Nurse'),
(3003,'Adina','Garza','Adina.Garza@kp.org','Nurse'),
(3004,'Felix','Horne','Felix.Horne@kp.org','Nurse'),
(3005,'Ralphie','Driscoll','Ralphie.Driscoll@kp.org','Nurse'),
(3006,'Jaskaran','Rodriquez','Jaskaran.Rodriquez@kp.org','Nurse'),
(4001,'Maxwell','Natasha','Natasha.Maxwell@gmail.com','Security'),
(4002,'Aguirre','Faizaan','Faizaan.Aguirre@yahoo.com','Security'),
(4003,'Aditya','Jones','Aditya.Jones@gmail.com','Security'),
(4004,'Lucas','Carter','Lucas.Carter@gmail.com','Security'),
(4005,'Louise','Velazquez','Louise.Velazquez@gmail.com','Security');
```

Inserting data into Staff Documents table

```
INSERT INTO staff_docs(StaffdocsNUID,LifeSafetyAttestation,DeptChecklist,E
quipList,ScavengerHunt)
VALUES
(1001,'Yes','Yes','No','No'),
(1002,'Yes','Yes','No','Yes'),
(1003,'Yes','No','No','No'),
(2001,'No','Yes','No','No'),
(2002,'Yes','No','Yes','No'),
(2003,'Yes','Yes','No','Yes'),
(3001,'Yes','Yes','Yes','No'),
(3002,'Yes','Yes','No','No'),
(3003,'Yes','No','No','No'),
(4001,'Yes','Yes','No','No'),
(4002,'Yes','Yes','Yes','No'),
(4003,'Yes','No','No','No');
```

Inserting data into Orientation Table

```
INSERT INTO orientation(OrientationNUID,TransitionalEd,LifeSafetyTour)
VALUES
(1001,'Yes','Yes'),
(1002,'Yes','Yes'),
(1003,'Yes','No'),
(1004,'No','Yes'),
(1005,'Yes','No'),
(1006,'Yes','Yes'),
(1007,'Yes','Yes'),
(1008,'No','Yes'),
(1009,'Yes','No'),
(1010,'Yes','Yes'),
(4001,'Yes','Yes'),
(4002,'Yes','Yes'),
(4003,'Yes','No'),
(4004,'No','Yes'),
(4005,'Yes','No');
```

Inserting data into Orientation Training Date Table

```
INSERT INTO orientation_trainingdate(TrainingdateNUID,TrainingDate)
VALUES
(1001,'2020-09-26'),
(1002,'2020-09-26'),
(1003,'2020-09-28'),
(1004,'2020-10-15'),
(1005,'2020-10-16'),
(1006,'2020-10-16'),
(1007,'2020-10-23'),
(1008,'2020-10-23'),
(1009,'2020-10-24'),
(1010,'2020-10-24'),
(4001,'2020-09-26'),
(4002,'2020-09-26'),
(4003,'2020-09-28'),
(4004,'2020-10-15'),
(4005,'2020-10-16');
```

Inserting data into Nursing Table

```
INSERT INTO nursing(NUNUID,NursingDirector,FloorNumber)
VALUES
(2001,'Holman Harrison','6N'),
(2002,'Sarina Whitley','4N'),
(2003,'Holman Harrison','5N'),
(2004,'Sarina Whitley','4N'),
(2005,'Holman Harrison','5N'),
(3001,'Sabah Lewis','4N'),
(3002,'Sabah Lewis','6N'),
(3003,'Sabah Lewis','4N'),
(3004,'Sabah Lewis', '4N'),
(3005,'Sabah Lewis','5N'),
(1001,'Elaine Connor','6N'),
(1002,'Linda Bridges','4N'),
(1003,'Daphne Martins','5N'),
(1004,'Shaine Meyers','4N'),
(1005,'Dylan Mcdonald','5N'),
(1006,'Ceri Beck','6N'),
(1007,'Marcel Herbert','6N'),
(1008,'Timmy Jones Jr.','4N'),
(1009,'Alex Baldwin','5N'),
(1010,'Anthony Cook','4N');
```

Inserting data into Clinical Ancillary Table

```
INSERT INTO clinical_ancillary(CANUID,ClinicalDirector)
VALUES
(2001,'Tj Ward'),
(2002,'Tj Ward'),
(2003,'Tj Ward'),
(2004,'Tj Ward'),
(2005,'Tj Ward'),
(1001,'Mick jo'),
(1002,'Mick Jo'),
(1003,'Rachel Adams'),
(1004,'Marcel Dickens'),
(1005,'Tyler Green'),
(1006,'Sequoia Parker'),
(1007,'Peter Liberty'),
(1008,'Andrew Star'),
(1009,'Joe Jo'),
(1010,'Rae Jay');
```

Insert data into Support Service Table

```
INSERT INTO support_services(SSNUID,SupportStaffDirector)
VALUES
(4001,'Sol Bennett'),
(4002,'Dilara Lowry'),
(4003,'Jayden Villa'),
(4004,'Elinor Martin'),
(4005,'Samah Leonard'),
(1001,'Sol Bennett'),
(1002,'Dilara Lowry'),
(1003,'Jayden Villa'),
(1004,'Elinor Martin'),
(1005,'Samah Leonard'),
(1006,'Jackson Forrest'),
(1007,'August Frank'),
(1008,'Keith Barron'),
(1009,'Matt Wickens'),
(1010,'Ella-Grace Yu');
```

Insert data into Nurse Table

```
INSERT INTO nurse(NRSNUID,NurseCallTraining)
VALUES
(2001,'No'),
(2002,'No'),
(2003,'Yes'),
(2004,'No'),
(2005,'Yes');
```

Insert data into Physician Table

```
INSERT INTO physician(PHYNUID,CodeBlueTraining)
VALUES
(1001,'Yes'),
(1002,'Yes'),
(1003,'Yes'),
(1004,'Yes'),
(1005,'Yes'),
(1006,'Yes'),
(1007,'No'),
(1008,'Yes'),
(1009,'Yes'),
(1010,'Yes');
```

Insert data into Anesthesiologist Table

```
INSERT INTO anesthesiologist(ANTNUID,AmbulatoryTraining)
VALUES
(1001,'Yes'),
(1002,'No'),
(1003,'Yes'),
(1004,'Yes'),
(1005,'Yes'),
(1006,'Yes'),
(1007,'No'),
(1008,'Yes'),
(1009,'Yes'),
(1010,'Yes');
```

Insert data into Pharmacist Table

```
INSERT INTO pharmacist(PHARMNUID,PneumaticTSTraining)
VALUES
(2001,'Yes'),
(2002,'Yes'),
(2003,'Yes'),
(2004,'Yes'),
(2005,'Yes');
```

Insert data into Nutritionist Table

```
INSERT INTO nutritionist(NUTRNUID,GetWellNetworkTraining)
VALUES
(1001,'No'),
(1002,'Yes'),
(1003,'Yes'),
(1004,'Yes'),
(1005,'Yes'),
(1006,'Yes'),
(1007,'No'),
(1008,'Yes'),
(1009,'No'),
(1010,'Yes');
```

Insert data into Physical Therapist Table

```
INSERT INTO physical_therapist(PTNUID,HillromBedTraining)
VALUES
(1001,'Yes'),
(1002,'Yes'),
(1003,'Yes'),
(1004,'Yes'),
(1005,'No'),
(1006,'Yes'),
(1007,'Yes'),
(1008,'No'),
(1009,'Yes'),
(1010,'Yes');
```

Insert data into Security Table

```
INSERT INTO security(SECNUID,FloorAccessTraining)
VALUES
(4001,'Yes'),
(4002,'Yes'),
(4003,'No'),
(4004,'Yes'),
(4005,'No'),
(1006,'Yes'),
(1007,'Yes'),
(1008,'No'),
(1009,'Yes'),
(1010,'Yes');
```

Insert data into Engineering Table

```
INSERT INTO engineering(ENGRNUID,MaintenanceTraining)
VALUES
(1001,'No'),
(1002,'No'),
(1003,'No'),
(1004,'Yes'),
(1005,'No'),
(1006,'Yes'),
(1007,'Yes'),
(1008,'No'),
(1009,'No'),
(1010,'Yes');
```

Insert data into Environmental Services Table

```
INSERT INTO environmental_services(EVSNUID,HazardousCleanupTraining)
VALUES
(1001,'Yes'),
(1002,'No'),
(1003,'Yes'),
(1004,'Yes'),
(1005,'No'),
(1006,'Yes'),
(1007,'No'),
(1008,'Yes'),
(1009,'No'),
(1010,'Yes');
```