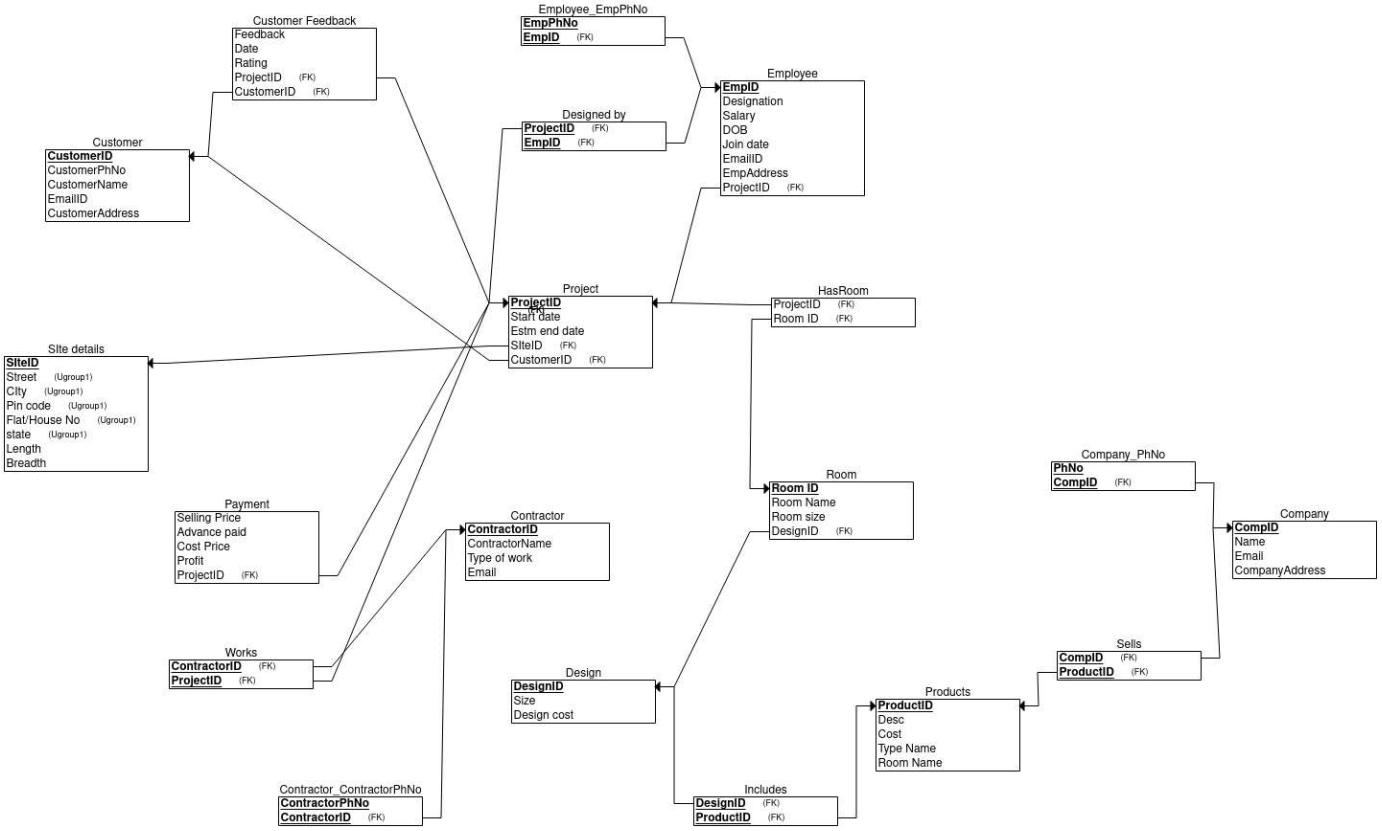
ASSIGNMENT - 2

1. Mapping conceptual schema to relational schema



2. Justifying the choice of model

The model chosen here is Relational database. This model fits the needs of the project adequately and fulfills the requirements needed from the model. Since the schema of the project is fixed and the need for change in the schema is minimal, RDBMS model has been preferred over NoSQL model.

Few advantages of RDBMS model that benefits the project are:

a. Checking for constraints: We have inbuilt support from the model to check for certain explicit constraints, such as
   - Null constraints
   - Unique constraints
   - Primary key constraints

- Foreign key constraints
- Domain constraints
- Integrity constraints

The above mentioned constraints are useful for the project, as it will help ensure there is no discrepancy in the data. For example, it will help ensure that each project is uniquely identifiable.

b. Flexibility
This feature of RDBMS saves a lot of time as updating data in one place is enough. For example, if we update one project, then it is updated in all files through their relationships (i.e. foreign keys). We do not need to individually update each.

c. Maintainability
It provides easy usability. It allows database admins to maintain, control, update data into the database easily. With RDBMS, backing up the data becomes easy. Automation tools of RBDMS automate these tasks.

3. Specific implementation to be used

The implementation we are using for the project is PostgreSQL. It is an enterprise-class open source database management system. It supports both SQL and JSON for relational and non-relational queries for extensibility and SQL compliance. PostgreSQL supports advanced data types and performance optimization features.

Postgres version installed - 13

4. Create commands

Create.sql:

```
create table project(
 projectID serial,
 siteID int not null,
 customerID int not null,
 startDate date,
 estimatedEndDate date check(estimatedEndDate > startDate),
 primary key (projectID)
);

create table employee(
```

```
 employeeID serial,
 empEmailID varchar(30),
 empAddress text,
 designation varchar(30),
 salary decimal check (salary > '15000'),
 dob date check(date_part('year',AGE(CURRENT_DATE,dob))>18),
 joinDate date check(joinDate > dob),
 projectID int,
 primary key (employeeID),
 FOREIGN KEY(projectID) REFERENCES project(projectID)
);

create table empPhNo(
 employeeID int,
 empPhNo varchar(10) unique,
 primary key(employeeID,empPhNo),
 FOREIGN KEY(employeeID) REFERENCES employee(employeeID)
);

create table designedBy(
 projectID int,
 employeeID int,
 primary key(projectID,employeeID),
 FOREIGN KEY(projectID) REFERENCES project(projectID),
 FOREIGN KEY(employeeID) REFERENCES employee(employeeID)
);

create table customer(
 customerID serial primary key,
 customerName varchar(30),
 customerPhNo varchar(10),
 customerEmailID varchar(30),
 customerAddress text
);

create table customerFeedback(
 customerID int,
 projectID int,
 feedback text,
 feedbackDate date,
 rating numeric(1) check (rating>=1 and rating<=5),
 FOREIGN KEY(projectID) REFERENCES project(projectID),
```

```
 FOREIGN KEY(customerID) REFERENCES customer(customerID)
);

create table design(
 designID serial,
 designCost decimal check (designCost>0),
 size char check(size IN ('S','M','L')),
 primary key (designID)
);

create table room(
 roomID serial,
 roomName varchar(30) not null,
 roomSize char check(roomSize IN('S','M','L')),
 designID int,
 primary key (roomID),
 FOREIGN KEY(designID) REFERENCES design(designID)
);

create table hasRoom(
 projectID int,
 roomID int,
 FOREIGN KEY(projectID) REFERENCES project(projectID),
 FOREIGN KEY(roomID) REFERENCES room(roomID)
);

create table product(
 productID serial,
 typeName varchar(30),
 roomName varchar(30),
 productCost decimal check(productCost>0),
 description text not null,
 primary key (productID)
);

create table designIncludesProducts(
 designID int,
 productID int,
 primary key (designID,productID),
 FOREIGN KEY(designID) REFERENCES design(designID),
 FOREIGN KEY(productID) REFERENCES product(productID)
);
```

```
create table siteDetails(
 siteID serial,
 houseNo varchar(10) not null,
 street text not null,
 pincode varchar(6) not null,
 city varchar(30) not null,
 state varchar(10),
 length decimal not null check(length>0),
 breadth decimal not null check(breadth>0),
 primary key(siteID)
);

create table payment(
 costPrice decimal check(costPrice>0),
 sellingPrice decimal check(sellingPrice>0 and sellingPrice-15000 >= costPrice),
 advancePaid decimal check(advancePaid< sellingPrice),
 profit decimal check(profit > 0),
 projectId int not null,
 FOREIGN KEY(projectID) REFERENCES project(projectID)
);

create table contractor(
 contractorID serial,
 contractorName varchar(30) not null,
 typeOfWork varchar(30) not null,
 contractorEmail varchar(30),
 primary key (contractorID)
);

create table contractorPhNo(
 contractorID int,
 contractorPhNo varchar(12) unique,
 primary key(contractorID,contractorPhNo),
 FOREIGN KEY(contractorID) REFERENCES contractor(contractorID)
);

create table works(
 contractorID int,
 projectID int,
 primary key(contractorID,projectID),
 FOREIGN KEY(projectID) REFERENCES project(projectID),
```

```
 FOREIGN KEY(contractorID) REFERENCES contractor(contractorID)
);

create table company(
 companyID serial,
 companyName varchar(30) not null,
 companyEmailID varchar(30),
 companyAddress text,
 primary key (companyID)
);

create table companyPhNo(
 companyID int,
 companyPhNo varchar(12) unique,
 primary key (companyID,companyPhNo),
 FOREIGN KEY(companyID) REFERENCES company(companyID)
);

create table sells(
 companyID int,
 productID int,
 primary key (companyID,productID)
);

alter table project
ADD FOREIGN KEY(siteID) REFERENCES SiteDetails(siteID);

alter table project
ADD FOREIGN KEY(customerID) REFERENCES customer(customerID);
```

5. Insert commands:

Insert.sql :

```
create table project(
 projectID serial,
 siteID int not null,
 customerID int not null,
 startDate date,
 estimatedEndDate date check(estimatedEndDate > startDate),
 primary key (projectID)
```

```
);

create table employee(
 employeeID serial,
 empEmailID varchar(30),
 empAddress text,
 designation varchar(30),
 salary decimal check (salary > '15000'),
 dob date check(date_part('year',AGE(CURRENT_DATE,dob))>18),
 joinDate date check(joinDate > dob),
 projectID int,
 primary key (employeeID),
 FOREIGN KEY(projectID) REFERENCES project(projectID)
);

create table empPhNo(
 employeeID int,
 empPhNo varchar(10) unique,
 primary key(employeeID,empPhNo),
 FOREIGN KEY(employeeID) REFERENCES employee(employeeID)
);

create table designedBy(
 projectID int,
 employeeID int,
 primary key(projectID,employeeID),
 FOREIGN KEY(projectID) REFERENCES project(projectID),
 FOREIGN KEY(employeeID) REFERENCES employee(employeeID)
);

create table customer(
 customerID serial primary key,
 customerName varchar(30),
 customerPhNo varchar(10),
 customerEmailID varchar(30),
 customerAddress text
);

create table customerFeedback(
 customerID int,
 projectID int,
 feedback text,
```

```
 feedbackDate date,
 rating numeric(1) check (rating>=1 and rating<=5),
 FOREIGN KEY(projectID) REFERENCES project(projectID),
 FOREIGN KEY(customerID) REFERENCES customer(customerID)
);

create table design(
 designID serial,
 designCost decimal check (designCost>0),
 size char check(size IN ('S','M','L')),
 primary key (designID)
);

create table room(
 roomID serial,
 roomName varchar(30) not null,
 roomSize char check(roomSize IN('S','M','L')),
 designID int,
 primary key (roomID),
 FOREIGN KEY(designID) REFERENCES design(designID)
);

create table hasRoom(
 projectID int,
 roomID int,
 FOREIGN KEY(projectID) REFERENCES project(projectID),
 FOREIGN KEY(roomID) REFERENCES room(roomID)
);

create table product(
 productID serial,
 typeName varchar(30),
 roomName varchar(30),
 productCost decimal check(productCost>0),
 description text not null,
 primary key (productID)
);

create table designIncludesProducts(
 designID int,
 productID int,
 primary key (designID,productID),
```

```
 FOREIGN KEY(designID) REFERENCES design(designID),
 FOREIGN KEY(productID) REFERENCES product(productID)
);

create table siteDetails(
 siteID serial,
 houseNo varchar(10) not null,
 street text not null,
 pincode varchar(6) not null,
 city varchar(30) not null,
 state varchar(10),
 length decimal not null check(length>0),
 breadth decimal not null check(breadth>0),
 primary key(siteID)
);

create table payment(
 costPrice decimal check(costPrice>0),
 sellingPrice decimal check(sellingPrice>0 and sellingPrice-15000 >= costPrice),
 advancePaid decimal check(advancePaid< sellingPrice),
 profit decimal check(profit > 0),
 projectId int not null,
 FOREIGN KEY(projectID) REFERENCES project(projectID)
);

create table contractor(
 contractorID serial,
 contractorName varchar(30) not null,
 typeOfWork varchar(30) not null,
 contractorEmail varchar(30),
 primary key (contractorID)
);

create table contractorPhNo(
 contractorID int,
 contractorPhNo varchar(12) unique,
 primary key(contractorID,contractorPhNo),
 FOREIGN KEY(contractorID) REFERENCES contractor(contractorID)
);

create table works(
 contractorID int,
```

```
 projectID int,
 primary key(contractorID,projectID),
 FOREIGN KEY(projectID) REFERENCES project(projectID),
 FOREIGN KEY(contractorID) REFERENCES contractor(contractorID)
);

create table company(
 companyID serial,
 companyName varchar(30) not null,
 companyEmailID varchar(30),
 companyAddress text,
 primary key (companyID)
);

create table companyPhNo(
 companyID int,
 companyPhNo varchar(12) unique,
 primary key (companyID,companyPhNo),
 FOREIGN KEY(companyID) REFERENCES company(companyID)
);

create table sells(
 companyID int,
 productID int,
 primary key (companyID,productID)
);

alter table project
ADD FOREIGN KEY(siteID) REFERENCES SiteDetails(siteID);

alter table project
ADD FOREIGN KEY(customerID) REFERENCES customer(customerID);
```

6. Work done:

The whole team got on call and each member has contributed equally to the project, for approximately 6.5 hours each.