University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ___2___

Date: ___15 Oct, 2024___

Group Number: ___97___

| Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Vamsi Nadella	60336625	p3c3i	nadellavamsi06@gmail.com
Jagathi Moturi	81887028	x4n6s	jagathi.moturi@gmail.com
Hans Chen	46387841	i2c5g	hanschen516@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project Overview:

Our project focuses on Renovation Project Management, designed to help owners, contractors, and supervisors coordinate tasks for both residential and commercial renovations. The system handles key aspects like project details, stakeholder roles, resource allocation, and budget tracking. It also includes a review system where owners can provide feedback upon project completion, enhancing accountability and service improvement.

Changes made to the ER Diagram:
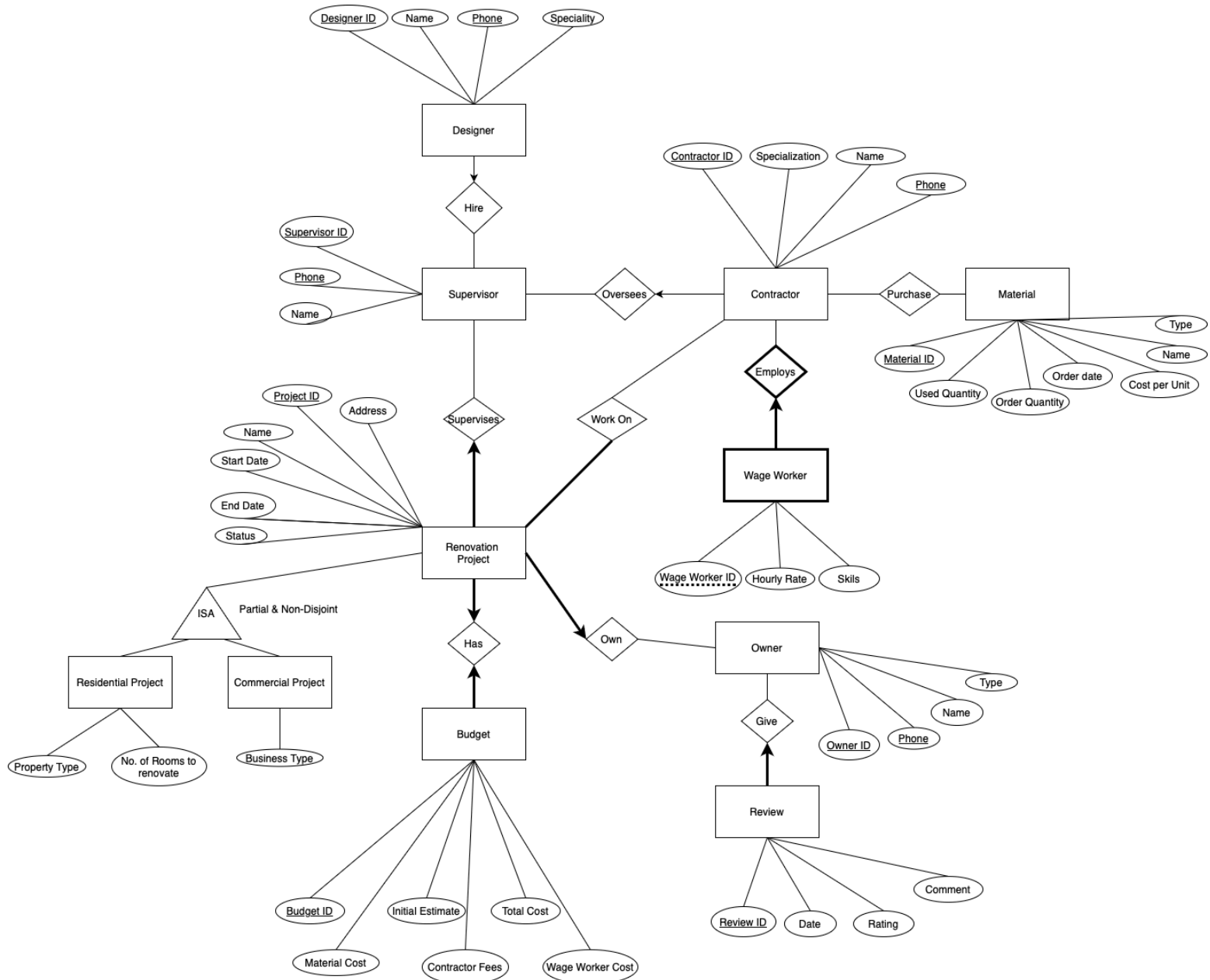
Phone number is made a primary key for Supervisor,Contractor,Designer entities as suggested by the TA.The ISA Relation between Renovation Project,Residential and Commercial Project is partial and non-disjoint.

4. Relational Schema:

- Designer (<u>Designer_ID</u> : VARCHAR[20],
  Designer_Name : VARCHAR[25],
  <u>Designer_Phone</u> : CHAR[10],
  Designer_Speciality : VARCHAR[20],
  Supervisor_ID : VARCHAR[20]),
  Designer_Name NOT NULL

- Supervisor (<u>Supervisor_ID</u> : VARCHAR[20],
  Supervisor_Name : VARCHAR[25],
  <u>Supervisor_Phone</u> : CHAR[10]),
  Supervisor_Name NOT NULL

- Contractor (<u>Contractor_ID</u> : VARCHAR[20],
  Contractor_Name : VARCHAR[25],
  <u>Contractor_Phone</u> : CHAR[10],
  Contractor_Specialization : VARCHAR[20],
  Supervisor_ID : VARCHAR[20]),
  Contractor_Name NOT NULL

- RenovationProject (<u>Project_ID</u> : VARCHAR[20],
  Project_Name : VARCHAR[25],
  Project_Address : VARCHAR[50],
  Project_Status : VARCHAR[20],
  Project_Start_Date : DATE,
  Project_End_Date : DATE,
  Supervisor_ID : VARCHAR[20],
  Budget_ID : VARCHAR[20],
  Owner_ID : VARCHAR[20]),
  Supervisor_ID NOT NULL,
  Budget_ID NOT NULL,
  Owner_ID NOT NULL,
  Project_Name NOT NULL

- ResidentialProject (**<u>Project_ID</u>** : VARCHAR[20],
  Property_Type : VARCHAR[20],
  No_of_Rooms_To_Renovate : INTEGER)

- CommercialProject (**Project_ID** : VARCHAR[20],
    Business_Type : VARCHAR[20])

- Budget (Budget_ID : VARCHAR[20],
    Budget_Initial_Estimate : NUMERIC[10, 2],
    Budget_Material_Cost : NUMERIC[10, 2],
    Budget_ Contractor_Fees : NUMERIC[10, 2],
    Budget_Wage_Worker_Cost : NUMERIC[10, 2],
    Budget_Total_Cost : NUMERIC[10, 2]),
    Budget_Initial_Estimate NOT NULL

- Owner (Owner_ID : VARCHAR[20],
    Owner_Name : VARCHAR[25],
    Owner_Phone : CHAR[10],
    Owner_Type : VARCHAR[20]),
    Owner_Name NOT NULL

- Review (Review_ID : VARCHAR[20],
    Review_Date : DATE,
    Review_Rating : CHAR[1],
    Review_Comment : VARCHAR[100],
    Owner_ID : VARCHAR[20]),
    Owner_ID NOT NULL

- Material (Material_ID : VARCHAR[20],
    Material_Name : VARCHAR[25],
    Material_Type : VARCHAR[25],
    Material_Order_Quantity : VARCHAR[25],
    Material_Used_Quantity : VARCHAR[25],
    Material_Order_Date : DATE,
    Material_Cost_Per_Unit : NUMERIC[5, 2]),
    Material_Name NOT NULL

- Purchase (**Material_ID** : VARCHAR[20],
    Contractor_ID : VARCHAR[20])

- Wage Worker (Wage_Worker_ID : VARCHAR[20],
  Contractor_ID : VARCHAR[20],
  Wage_Worker_Hourly_Rate : Numeric[4,2],
  Wage_Worker_Skills : VARCHAR[20]),
  Wage_Worker_Hourly_Rate NOT NULL

- Work On (**Contractor_ID** : VARCHAR[20],
  Project_ID : VARCHAR[20])

5. **Primary Key FDs:**
Purple → newly added attributes for normalization

1. Hire (Designer_ID, Designer_Name, Designer_Phone, Designer_Specialty, Supervisor_ID, Supervisor_Phone, Supervisor_Name, Designer_Experience_ Level, Designer_License_Number, Designer_Hourly_Rate)

   - Designer_ID → Designer_Name, Designer_Phone, Designer_Specialty, Supervisor_ID
   - Supervisor_ID → Supervisor_Phone, Supervisor_Name
   - Designer_Phone → Designer_ID
   - Supervisor_Phone → Supervisor_ID
   - Designer_Experience_Level → Designer_Hourly_Rate
   - Designer_License_Number → Designer_ID

2. Oversees (Contractor_ID, Contractor_Specialization, Contractor_Name, Contractor_Phone, Supervisor_ID, Supervisor_Phone, Supervisor_Name, Contractor_License_Number)

   - Contractor_ID -> Contractor_Specialization, Contractor_Name, Contractor_Phone, Supervisor_ID
   - Supervisor_ID -> Supervisor_Phone, Supervisor_Name
   - Supervisor_Phone → Supervisor_ID
   - Contractor_Phone -> Contractor_ID
   - Contractor_License_Number -> Contractor_ID

3. Purchase (Material_ID, Material_Type, Material_Name, Material_Cost_per_unit, Material_Order_date, Material_Order_quantity, Material_Used_quantity)

- Material_ID -> Material_Type, Material_Name, Material_Cost_Per_Unit, Material_Order_Date, Material_Order_Quantity, Material_Used_quantity

4. Project (Project_ID, Project_Address, Project_Name, Project_Start_Date, Project_ End_Date, Project_Status, Supervisor_ID, Supervisor_Phone, Supervisor_Name, Budget_ID, Budget_Material_Cost, Budget_Initial_Estimate, Budget_Contractor_fees, Budget_Total_cost, Budget_Wage_Worker_Cost, Owner_ID, Owner_Name, Owner_Type, Owner_Phone, Property_Type, No_of_Rooms_To_Renovate, Business_Type)

   - Project_ID -> Project_ID, Project_Address, Project_Name, Project_Start_ Date, Project_End_Date, Project_Status, Supervisor_ID, Budget_ID, Owner_ID
   - Project_ID -> Property_Type, No_of_Rooms_To_Renovate
   - Project_ID -> Business_Type
   - Supervisor_ID -> Supervisor_Phone, Supervisor_Name
   - Supervisor_Phone → Supervisor_ID
   - Budget_ID -> Budget_Material_Cost, Budget_Initial_Estimate, Budget_Contractor_Fees, Budget_Total_cost, Budget_Wage_worker _cost
   - Owner_ID -> Owner_Name, Owner_Type, Owner_Phone
   - Owner_Phone → Owner_ID

5. Give (Review_ID, Review_Date, Review_Rating, Review_Comment, Owner_ Phone, Owner_Name, Owner_Type, Owner_ID)

   - Review_ID -> Review_Date, Review_Rating, Review_Comment, Owner_ ID
   - Owner_ID → Owner_Phone, Owner_Name, Owner_Type
   - Owner_Phone -> Owner_ID

6. Employs (Wage_Worker_ID, Contractor_ID, Wage_Worker_Hourly_Rate, Wage_Worker_Skills, Contractor_Specialization, Contractor_Name, Contractor_Phone, Contractor_License_Number)

   - Contractor_Phone -> Contractor_ID
   - Wage_Worker_ID → Wage_Worker_Hourly_rate, Wage_Worker_Skills

- Contractor_ID → Contractor_Specialization, Contractor_Name, Contractor_ Phone
- Contractor_License_Number -> Contractor_ID

6. Normalization:

1. Hire (Designer_ID, Designer_Name, Designer_Phone, Designer_Specialty, Supervisor_ID, Supervisor_Phone, Supervisor_Name, Designer_Experience_ Level, Designer_License_Number, Designer_Hourly_Rate)

    - Designer_ID → Designer_Name, Designer_Phone, Designer_Specialty, Supervisor_ID
    - Supervisor_ID → Supervisor_Phone, Supervisor_Name
    - Designer_Phone → Designer_ID
    - Supervisor_Phone → Supervisor_ID
    - Designer_Experience_Level → Designer_Hourly_Rate
    - Designer_License_Number → Designer_ID

Finding All the Minimal Keys:
- Left: Designer_Experience_Level, Designer_License_Number
- Middle: Designer_ID, Supervisor_ID, Designer_Phone, Supervisor_Phone
- Right: Designer_Name, Designer_Specialty, Designer_Hourly_Rate, Supervisor_Name

Closure:
- Designer_ID+ = {Designer_ID, Designer_Name, Designer_Phone, Designer_Specialty, Supervisor_ID, Supervisor_Phone, Supervisor_Name}

- Supervisor_ID+ = {Supervisor_Phone, Supervisor_ID, Supervisor_Name}

- Designer_Phone+ = {Designer_Phone, Designer_ID, Designer_Name, Designer_Speciality, Supervisor_ID, Supervisor_Phone, Supervisor_Name}

- Supervisor_Phone+ = {Supervisor_Phone, Supervisor_ID, Supervisor_Name}

- Designer_Experience_Level+ = {Designer_Experience_Level, Designer_Hourly_Rate}
- Designer_License_Number = {Designer_License_Number, Designer_ID, Designer_Name, Designer_Phone, Designer_Specialty, Supervisor_ID, Supervisor_Phone, Supervisor_Name}

- {Designer_Experience_Level, Designer_License_Number}+ = {Designer_Experience_Level, Designer_Hourly_Rate, Designer_License_Number, Designer_ID, Designer_Name, Designer_Phone, Designer_Specialty, Supervisor_ID, Supervisor_Phone, Supervisor_Name}

Key: {Designer_Experience_Level, Designer_License_Number}

BCNF:
- **Designer_Experience_Level → Designer_Hourly_Rate** violates BCNF because Designer_Experience_Level is not a super key for the relation

   Decompose (Designer_Experience_Level → Designer_Hourly_Rate):
   Left: Designer_License_Number, Designer_ID, Designer_Name, Designer_Phone, Designer_Specialty, Supervisor_ID, Supervisor_Phone, Supervisor_Name
   Middle: Designer_Experience_Level
   Right: Designer_hourly_Rate

R1 (Designer_Experience_Level, Designer_hourly_rate)

R2 (Designer_License_Number, Designer_ID, Designer_Name, Designer_Phone, Designer_Specialty, Supervisor_ID, Supervisor_Phone, Supervisor_Name, Designer_Experience_level)

- **Designer_ID → Designer_Name, Designer_Phone, Designer_Specialty, Supervisor_ID** violates BCNF because Designer_ID is not a superkey in relation R2

   Decompose (Designer_ID → Designer_Name, Designer_Phone, Designer_Specialty, Supervisor_ID):
   Left: Designer_License_Number, Supervisor_Phone, Supervisor_Name, Designer_Experience_level

Middle: Designer_ID
Right: Designer_Name, Designer_Phone, Designer_Specialty, Supervisor_ID

R3 (Designer_ID, Designer_Name, Designer_Phone, Designer_Speciality, Supervisor_ ID)

R4 (Designer_License_Number, Supervisor_Phone, Supervisor_Name, Designer_ Experience_Level, Designer_ID)

- **Designer_License_Number → Designer_ID** violates BCNF because Designer License Number is not a superkey for R4

    Decompose (Designer_License_Number → Designer_ID):
    Left: Supervisor_Name, Designer_Experience_Level, Supervisor_Phone
    Middle: Designer_Licence_Number
    Right: Designer_ID

R5 (Designer_License_Number, Designer_ID)

R6 (Supervisor_Name, Designer_Experience_Level, Supervisor_Phone, Designer_ Licence_Number)

Final Answer: R1 (<u>Designer_Experience_level</u>, Designer_hourly_rate);
              R3 (<u>Designer_ID</u>, Designer_Name, <u>Designer_Phone</u>,
                 Designer_Speciality, **Supervisor_ID**),
              R5 (<u>Designer_License_Number</u>, Designer_ID),
              R6 (Supervisor_Name, **<u>Designer_Experience_Level</u>**,
               <u>Supervisor_Phone</u>, **<u>Designer_Licence_Number</u>**)

Tables for Hire:

Hire_R1 (<u>Designer_Experience_Level</u> : VARCHAR[30],
            Designer_Hourly_Rate : NUMERIC[4,2]),
            Designer_Hourly_Rate NOT NULL

Hire_R3 (<u>Designer_ID</u> : VARCHAR[20],
            Designer_Name : VARCHAR[25],
            <u>Designer_Phone</u> : CHAR[10],
            Designer_Speciality : VARCHAR[20],

Supervisor_ID : VARCHAR[20]),
Designer_Name NOT NULL

Hire_R5 (Designer License_Number : VARCHAR[20],
Designer_ID : VARCHAR[20]),
Designer_ID NOT NULL

Hire_R6 (Supervisor_Name : VARCHAR[25],
Designer_Experience_Level : VARCHAR[30],
Supervisor_Phone : CHAR[10],
Designer Licence Number : VARCHAR[20]),
Supervisor_Name NOT NULL

===

2. Oversees (Contractor_ID, Contractor_Specialization, Contractor_Name, Contractor_Phone, Supervisor_ID, Supervisor_Phone, Supervisor_Name,Contractor_License_Number)
   - Contractor_ID -> Contractor_Specialization, Contractor_Name, Contractor_Phone, Supervisor_ID
   - Supervisor_ID -> Supervisor_Phone, Supervisor_Name
   - Supervisor_Phone → Supervisor_ID
   - Contractor_Phone -> Contractor_ID
   - Contractor_License_Number -> Contractor_ID

Finding All the Minimal Keys:
   - Left:Contractor_License_Number
   - Middle: Contractor_ID, Supervisor_Phone, Contractor_Phone, Supervisor_ID
   - Right: Contractor_Specialization, Contractor_Name, Supervisor_Name

Closures:
   - Contractor_ID+ = {Contractor_ID, Contractor_Specialization, Contractor_Name, Contractor_Phone, Supervisor_ID, Supervisor_Phone, Supervisor_Name}
   - Supervisor_ID+ = {Supervisor_ID, Supervisor_Phone, Supervisor_Name}

- Supervisor_Phone+ = {Supervisor_Phone, Supervisor_ID, Supervisor_Name}
- Contractor_Phone+ = {Contractor_Phone, Contractor_ID, Contractor_Specialization, Contractor_Name, Supervisor_ID, Supervisor_Phone, Supervisor_Name}
- Contractor License Number+ = {Contractor License Number, Contractor_ID, Contractor_Specialization, Contractor_Name, Contractor_Phone, Supervisor_ID, Supervisor_Phone, Supervisor_Name}

Key: Contractor_License_Number

BCNF:

- **Contractor_ID → Contractor_Specialization, Contractor_Name, Contractor_Phone, Supervisor_ID** violates BCNF because Contractor_ID is not a superkey for the relation

Decompose (Contractor_ID → Contractor_Specialization, Contractor_Name, Contractor_Phone, Supervisor_ID):

Left:Contractor_License_Number, Supervisor_Phone, Supervisor_Name
Middle: Contractor_ID
Right: Contractor_Specialization, Contractor_Name, Contractor_Phone, Supervisor_ID

R1 (Contractor_ID, Contractor_Specialization, Contractor_Name, Contractor_Phone, Supervisor_ID)

R2 (Contractor License Number, Supervisor_Phone, Supervisor_Name, Contractor_ID)

Final Answer: R1 (Contractor_ID, Contractor_Specialization, Contractor_Name, Contractor_Phone, **Supervisor_ID**),
R2 (Contractor License Number, Supervisor_Phone, Supervisor_Name, **Contractor_ID**)

Tables for Oversees:
Oversees_R1 (Contractor_ID : VARCHAR[20],

Contractor_Name : VARCHAR[25],
Contractor_Phone : CHAR[10],
Contractor_Specialization : VARCHAR[20],
Supervisor_ID : VARCHAR[20]),
Contractor_Name NOT NULL

Oversees_R2 (Contractor_License_Number : VARCHAR[20],
Supervisor_Phone : CHAR[10],
Supervisor_Name : VARCHAR[25],
Contractor_ID : VARCHAR[20]),
Contractor_ID NOT NULL,
Supervisor_Name NOT NULL

===

3. Purchase (Material_ID, Material_Type, Material_Name, Material_Cost_Per_Unit, Material_Order_Date, Material_Order_Quantity, Material_Used_Quantity)

- Material_ID -> Material_Type, Material_Name, Material_Cost_Per_Unit, Material_Order_Date, Material_Order_Quantity, Material_Used_Quantity

Closure:
- Material_ID+ = {Material_ID, Material_Type, Material_Name, Material_Cost_Per_Unit, Material_Order_Date, Material_Order_Quantity, Material_Used_Quantity}

Key: Material_ID

BCNF:
- Material_ID → Material_Type, Material_Name, Material_Cost_Per_Unit, Material_Order_Date, Material_Order_Quantity, Material_Used_Quantity does not violate BCNF as Material_ID is the superkey for the relation.

Final Answer: R1 {Material_ID, Material_Type, Material_Name, Material_Cost_Per_Unit, Material_Order_Date, Material_Order_Quantity, Material_Used_Quantity}

Table for Purchase:

Purchase_R1 (Material_ID : VARCHAR[20],
            Material_Name : VARCHAR[25],
            Material_Type : VARCHAR[25],
            Material_Order_Quantity : VARCHAR[25],
            Material_Used_Quantity : VARCHAR[25],
            Material_Order_Date : DATE,
            Material_Cost_Per_Unit : NUMERIC[5, 2]),
            Material_Name NOT NULL

Purchase (**Material_ID** : VARCHAR[20],
          Contractor_ID : VARCHAR[20])

===

4. Project (Project_ID, Project_Address, Project_Name, Project_Start_Date, Project_End_Date, Project_Status, Supervisor_ID, Supervisor_Phone, Supervisor_Name, Budget_ID, Budget_Material_Cost, Budget_Initial_Estimate, Budget_Contractor_fees, Budget_Total_Cost, Budget_Wage_Worker_Cost, Owner_ID, Owner_Name, Owner_Type, Owner_Phone, Property_Type, No_of_Rooms_To_Renovate, Business_Type)

   - Project_ID -> Project_ID, Project_Address, Project_Name, Project_Start_Date, Project_End_Date, Project_Status, Supervisor_ID, Budget_ID, Owner_ID
   - Project_ID -> Property_Type, No_of_Rooms_To_Renovate
   - Project_ID -> Business_Type
   - Supervisor_ID -> Supervisor_Phone, Supervisor_Name
   - Supervisor_Phone → Supervisor_ID
   - Budget_ID -> Budget_Material_Cost, Budget_Initial_Estimate, Budget_Contractor_fees, Budget_Total_Cost, Budget_Wage_Worker_Cost
   - Owner_ID -> Owner_Name, Owner_Type, Owner_Phone
   - Owner_Phone → Owner_ID

Closure:

- Project_ID+ = {Project_ID, Project_Address, Project_Name, Project_Start_Date, Project_End_Date, Project_Status, Supervisor_ID, Supervisor_Phone, Supervisor_Name, Budget_ID, Budget_Material_Cost, Budget_Initial_Estimate, Budget_Contractor_fees, Budget_Total_Cost, Budget_Wage_Worker_Cost, Owner_ID, Owner_Name, Owner_Type, Owner_Phone, Property_Type, No_of_Rooms_To_Renovate, Business_Type}
- Supervisor_ID+ = {Supervisor_ID, Supervisor_Phone, Supervisor_Name}
- Supervisor_Phone+ = {Supervisor_Phone, Supervisor_ID, Supervisor_Name}
- Budget_ID+ = {Budget_ID, Budget_Material_Cost, Budget_Initial_Estimate, Budget_Contractor_fees, Budget_Total_Cost, Budget_Wage_Worker_Cost}
- Owner_ID+ = {Owner_ID+, Owner_Name, Owner_Type, Owner_Phone}
- Owner_Phone+ = {Owner_Phone, Owner_ID, Owner_Name, Owner_Type}

Key: Project_ID

BCNF:

- **Supervisor_ID → Supervisor_Phone, Supervisor_Name** violates BCNF because that is not a super key for this relation

    Decompose (Supervisor_ID -> Supervisor_Phone, Supervisor_Name):
    Left: Project_ID, Project_Address, Project_Name, Project_Start_Date, Project_End_Date, Project_Status, Budget_ID, Budget_Material_Cost, Budget_Initial_Estimate, Budget_Contractor_fees, Budget_Total_Cost, Budget_Wage_Worker_Cost, Owner_ID, Owner_Name, Owner_Type, Owner_Phone, Property_Type, No_of_Rooms_To_Renovate, Business_Type
    Middle: Supervisor_ID
    Right: Supervisor_Phone, Supervisor_Name

R1 (Supervisor_ID, Supervisor_Phone, Supervisor_Name)

R2 (Project_ID, Project_Address, Project_Name, Project_Start_Date, Project_End_Date, Project_Status, Budget_ID, Budget_Material_Cost,

Budget_Initial_Estimate, Budget_Contractor_fees, Budget_Total_Cost,
Budget_Wage_Worker_Cost, Owner_ID, Owner_Name, Owner_Type, Owner_Phone,
Supervisor_ID, Property_Type, No_of_Rooms_To_Renovate, Business_Type)

- **Budget_ID → Budget_Material_Cost, Budget_Initial_Estimate,
  Budget_Contractor_fees, Budget_Total_Cost,
  Budget_Wage_Worker_Cost** violates BCNF because Budget_ID is not a
  superkey for R2
  Decompose:
  Left: Project_ID, Project_Address, Project_Name, Project_Start_Date,
      Project_End_Date, Project_Status, Owner_ID, Owner_Name,
      Owner_Type, Owner_Phone, Supervisor_ID, Property_Type,
       No_of_Rooms_To_Renovate, Business_Type
  Middle: Budget_ID
  Right: Budget_Material_Cost, Budget_Initial_Estimate,
       Budget_Contractor_fees, Budget_Total_Cost,
       Budget_Wage_Worker_Cost

R3 (Budget_ID, Budget_Material_Cost, Budget_Initial_Estimate,
    Budget_Contractor_fees, Budget_Total_Cost, Budget_Wage_Worker_Cost)

R4 (Project_ID, Project_Address, Project_Name, Project_Start_Date,
    Project_End_Date, Project_Status, Owner_ID, Owner_Name, Owner_Type,
    Owner_Phone, Supervisor_ID, Budget_ID, Property_Type,
    No_of_Rooms_To_Renovate, Business_Type)

- **Owner_ID -> Owner_Name, Owner_Type, Owner_Phone** violates BCNF
  because it is not a superkey for the relation R4
  Decompose:
  Left: Project_ID, Project_Address, Project_Name, Project_Start_Date,
      Project_End_Date, Project_Status, Supervisor_ID, Budget_ID,
      Property_Type, No_of_Rooms_To_Renovate, Business_Type
  Middle: Owner_ID
  Right: Owner_Name, Owner_Type, Owner_Phone

R5 (Owner_ID, Owner_Name, Owner_Type, Owner_Phone)

R6 (Project_ID, Project_Address, Project_Name, Project_Start_Date,
    Project_End_Date, Project_Status, Supervisor_ID, Budget_ID, Owner_ID,
    Property_Type, No_of_Rooms_To_Renovate, Business_Type)

Final Answer: R1 (<u>Supervisor_ID</u>, <u>Supervisor_Phone</u>, Supervisor_Name),
R3 (<u>Budget_ID</u>, Budget_Material_Cost, Budget_Initial_Estimate,
Budget_Contractor_fees, Budget_Total_Cost,
Budget_Wage_Worker_Cost),
R5 (<u>Owner_ID</u>, Owner_Name, Owner_Type, <u>Owner_Phone</u>),
R6 (<u>Project_ID</u>, Project_Address, Project_Name, Project_Start_Date,
Project_End_Date, Project_Status, Property_Type, No_of_Rooms_To_Renovate,
Business_Type, **Supervisor_ID, Budget_ID, Owner_ID**)

Tables for Project:

Project_R1(<u>Supervisor_ID</u> : VARCHAR[20],
Supervisor_Name : VARCHAR[25],
<u>Supervisor_Phone</u> : CHAR[10]),
Supervisor_Name NOT NULL

Project_R3(<u>Budget_ID</u> : VARCHAR[20],
Budget_Initial_Estimate : NUMERIC[10, 2],
Budget_Material_Cost : NUMERIC[10, 2],
Budget_Contractor_Fees : NUMERIC[10, 2],
Budget_Wage_Worker_Cost : NUMERIC[10, 2],
Budget_Total_Cost : NUMERIC[10, 2]),
Budget_Initial_Estimate NOT NULL

Project_R5(<u>Owner_ID</u> : VARCHAR[20],
Owner_Name : VARCHAR[25],
<u>Owner_Phone</u> : CHAR[10],
Owner_Type : VARCHAR[20]),
Owner_Name NOT NULL

Project_R6(<u>Project_ID</u> : VARCHAR[20],
Project_Name : VARCHAR[25],
Project_Address : VARCHAR[50],
Project_Status : VARCHAR[20],
Project_Start_Date : DATE,
Project_End_Date : DATE,
Supervisor_ID : VARCHAR[20],

Budget_ID : VARCHAR[20],
Owner_ID : VARCHAR[20],
Property_Type : VARCHAR[20],
No_of_Rooms_To_Renovate : INTEGER,
Business_Type : VARCHAR[20]),
Supervisor_ID NOT NULL,
Budget_ID NOT NULL,
Owner_ID NOT NULL,
Project_Name NOT NULL

Work On (**Contractor_ID** : VARCHAR[20],
　　　　Project_ID : VARCHAR[20])

==

5. Give (Review_ID, Review_Date, Review_Rating, Review_Comment, Owner_Phone, Owner_Name, Owner_Type, Owner_ID)

   ● Review_ID -> Review_Date, Review_Rating, Review_Comment, Owner_ID
   ● Owner_ID → Owner_Phone, Owner_Name, Owner_Type
   ● Owner_Phone -> Owner_ID

Find all the minimal keys:
　　　Left: Review_ID
　　　Middle: Owner_ID, Owner_Phone
　　　Right: Review_Date, Review_Rating, Review_Comment, Owner_Name,
　　　　　　Owner_Type

Closure:
   ● Review_ID+ = {Review_ID, Review_Date, Review_Rating, Review_Comment, Owner_Phone, Owner_Name, Owner_Type, Owner_ID}
   ● Owner_ID+ = {Owner_ID, Owner_Phone, Owner_Name, Owner_Type}
   ● Owner_Phone+ = {Owner_Phone, Owner_ID, Owner_Name, Owner_Type}

Key: Review_ID

BCNF:
- **Owner_ID → Owner_Phone, Owner_Name, Owner_Type** violates BCNF because it is not a superkey of the relation
  Decompose:
  Left: Review_ID, Review_Date, Review_Rating, Review_Comment
  Middle: Owner_ID
  Right: Owner_Phone, Owner_Name, Owner_Type

R1 (Owner_ID, Owner_Phone, Owner_Name, Owner_Type)

R2 (Review_ID, Review_Date, Review_Rating, Review_Comment, Owner_ID)

Final Answer: R1 (Owner_ID, Owner_Phone, Owner_Name, Owner_Type),
R2 (Review_ID, Review_Date, Review_Rating, Review_Comment, **Owner_ID**)

Tables for Give:

Give_R1(Owner_ID : VARCHAR[20],
   Owner_Name : VARCHAR[25],
   Owner_Phone : CHAR[10],
   Owner_Type : VARCHAR[20]),
   Owner_Name NOT NULL

Give_R2(Review_ID : VARCHAR[20],
   Review_Date : DATE,
   Review_Rating : CHAR[1],
   Review_Comment : VARCHAR[100],
   Owner_ID : VARCHAR[20]),
   Owner_ID NOT NULL

==

6. Employs (Wage_Worker_ID, Contractor_ID, Wage_Worker_Hourly_Rate, Wage_Worker_Skills, Contractor_Specialization, Contractor_Name, Contractor_Phone,Contractor_License_Number)

- Contractor_Phone -> Contractor_ID
- (Wage_Worker_ID, Contractor_ID) → Wage_Worker_Hourly_Rate, Wage_Worker_Skills
- Contractor_ID → Contractor_Specialization, Contractor_Name, Contractor_Phone
- Contractor_License_Number -> Contractor_ID

Find Minimal Keys:
      Left:Contractor_License_Number, Wage_Worker_ID
      Middle: Contractor_Phone, Contractor_ID
      Right: Wage_Worker_Hourly_Rate, Wage_Worker_Skills,
          Contractor_Specialization, Contractor_Name

Closure:
- Contractor_Phone+ = {Contractor_Phone, Contractor_ID, Contractor_Name, Contractor_Specialization}
- Wage_Worker_ID+ = {Wage_Worker_ID, Wage_Worker_Hourly_Rate, Wage_Worker_Skills, Contractor_ID, Contractor_Specialization, Contractor_Name, Contractor_Phone}
- Contractor_ID+ = {Contractor_ID, Contractor_Specialization, Contractor_Name, Contractor_Phone}
- (Wage_Worker_ID, Contractor_Licence_Number)+ = {Wage_Worker_ID, Contractor_ID, Wage_Worker_Hourly_Rate, Wage_Worker_Skills, Contractor_Specialization, Contractor_Name, Contractor_Phone, Contractor_Licence_Number}
- (Wage_Worker_ID, Contractor_ID)+ = {Wage_Worker_ID, Contractor_ID, Wage_Worker_Hourly_Rate, Wage_Worker_Skills, Contractor_Specialization, Contractor_Name, Contractor_Phone}

Key: (Wage_Worker_ID, Contractor_Licence_Number)

BCNF:

- **(Wage_Worker_ID, Contractor_ID) → Wage_Worker_Hourly_Rate, Wage_Worker_Skills** violates BCNF because it is not a super key for this relation

      Decompose:
      Left: Contractor_Specialization, Contractor_Name, Contractor_Phone,
          Contractor_Licence_Number

Middle: Wage_Worker_ID, Contractor_ID
Right: Wage_Worker_Hourly_Rate, Wage_Worker_Skills

R1 (Wage_Worker_ID, Contractor_ID, Wage_Worker_Hourly_Rate, Wage_Worker_Skills)

R2 (Contractor Specialization, Contractor_Name, Contractor_Phone, Contractor_Licence_Number, Wage_Worker_ID, Contractor_ID)

- **Contractor_ID -> Contractor_Specialization, Contractor_Name, Contractor_Phone** violates BCNF because it is not a superkey for this relation R2

  Decompose:
  Left: Contractor_Licence_Number, Wage_Worker_ID
  Middle: Contractor_ID
  Right: Contractor_Specialization, Contractor_Name, Contractor_Phone

R3 (Contractor_ID, Contractor_Specialization, Contractor_Name, Contractor_Phone)

R4 (Contractor_Licence_Number, Wage_Worker_ID, Contractor_ID)

Final Answer: R1 (<u>Wage_Worker_ID</u>, **<u>Contractor_ID</u>**, Wage_Worker_Hourly_Rate, Wage_Worker_Skills),
R3 (<u>Contractor_ID</u>, Contractor_Specialization, Contractor_Name, Contractor_Phone),
R4 (<u>Contractor_Licence_Number,</u> **Wage_Worker_ID**, Contractor_ID)

Tables for Employs:

Employs_R1 (<u>Wage_Worker_ID</u> : VARCHAR[20],
<u>Contractor_ID</u> : VARCHAR[20],
Wage_Worker_Hourly_Rate : Numeric[4,2],
Wage_Worker_Skills : VARCHAR[20]),
Wage_Worker_Hourly_Rate NOT NULL

Employs_R3 (<u>Contractor_ID</u> : VARCHAR[20],
          Contractor_Name : VARCHAR[25],
          Contractor_Phone : CHAR[10],
          Contractor_Specialization : VARCHAR[20]),
          Contractor_Name NOT NULL)

Employs_R4 (<u>Contractor_License_Number</u> : VARCHAR[20],
          Wage_Worker_ID : VARCHAR[20],
          Contractor_ID : VARCHAR[20])

==

7. SQL DDL:

CREATE TABLE DesignerExperienceLevel (
      Designer_Experience_Level VARCHAR(30),
      Designer_Hourly_Rate NUMERIC(4,2) NOT NULL,
      PRIMARY KEY (Designer_Experience_Level)

)

CREATE TABLE DesignerDetails (
      Designer_ID VARCHAR(20),
      Designer_Name VARCHAR(25) NOT NULL,
      Designer_Phone CHAR(10),
      Designer_Specialty VARCHAR(20),
      Supervisor_ID VARCHAR(20),
      PRIMARY KEY (Designer_ID,Designer_Phone),
      FOREIGN KEY (Supervisor_ID) REFERENCES Supervisor
)

```sql
CREATE TABLE DesignerLicense (
        Designer_License_Number VARCHAR(20),
        Designer_ID VARCHAR(20) NOT NULL,
        PRIMARY KEY (Designer_License_Number)
)


CREATE TABLE SupervisorDetails(
        Supervisor_Name VARCHAR(25) NOT NULL,
        Supervisor_Phone CHAR(10),
        Designer_Experience_Level VARCHAR(30) NOT NULL,
        Designer_License_Number VARCHAR(20) NOT NULL,
        PRIMARY KEY (Supervisor_Phone),

        FOREIGN KEY (Designer_License_Number) REFERENCES
        DesignerLicense,
        FOREIGN KEY (Designer_Experience_Level) REFERENCES
        DesignerExperienceLevel
)


CREATE TABLE Supervisor (
    Supervisor_ID VARCHAR(20),
    Supervisor_Name VARCHAR(25) NOT NULL,
    Supervisor_Phone CHAR(10),
    PRIMARY KEY (Supervisor_ID,Supervisor_Phone)
)


CREATE TABLE Contractor (
    Contractor_ID VARCHAR(20),
    Contractor_Name VARCHAR(25) NOT NULL,
    Contractor_Specialization VARCHAR(20),
    Contractor_Phone CHAR(10),
    Supervisor_ID VARCHAR(20),
    PRIMARY KEY (Contractor_ID,Contractor_Phone),
    FOREIGN KEY (Supervisor_ID) REFERENCES Supervisor
)
```

```sql
CREATE TABLE ContractorLicense (
    Contractor_License_Number VARCHAR(20),
    Supervisor_Name VARCHAR(25) NOT NULL,
    Supervisor_Phone CHAR(10),
    Contractor_ID VARCHAR(20) NOT NULL,
    PRIMARY KEY (Contractor_License_Number, Supervisor_Phone),
    FOREIGN KEY (Contractor_ID) REFERENCES Contractor
)


CREATE TABLE Material (

        Material_ID VARCHAR(20),
        Material_Type VARCHAR(25),
        Material_Name VARCHAR(25) NOT NULL,
        Material_Order_Quantity VARCHAR(25),
        Material_Used_Quantity VARCHAR(25),
        Material_Order_Date DATE,
        Material_Cost_Per_Unit NUMERIC(5,2),
        PRIMARY KEY (Material_ID)
)

CREATE TABLE Purchase (
        Material_ID VARCHAR(20),
        Contractor_ID VARCHAR(20),
        PRIMARY KEY (Material_ID,Contractor_ID),
        FOREIGN KEY (Material_ID) REFERENCES Material,
        FOREIGN KEY (Contractor_ID) REFERENCES Contractor
)


CREATE TABLE Project (
        Project_ID VARCHAR(20),
        Project_Address VARCHAR(50),
        Project_Name VARCHAR(25) NOT NULL,
        Project_Start_Date DATE,
        Project_End_Date DATE,
        Project_Status VARCHAR(20),
        Supervisor_ID VARCHAR(20) NOT NULL,
```

```
        Budget_ID VARCHAR(20) NOT NULL,
        Owner_ID VARCHAR(20) NOT NULL,
        PRIMARY KEY (Project_ID),
        FOREIGN KEY (Supervisor_ID) REFERENCES Supervisor,
        FOREIGN KEY (Budget_ID) REFERENCES Budget,
        FOREIGN KEY (Owner_ID) REFERENCES Owner
)

CREATE TABLE ResidentialProject (
        Project_ID VARCHAR(20),
        Property_Type VARCHAR(20),
        No_of_rooms_To_Renovate INTEGER,
        PRIMARY KEY (Project_ID),
        FOREIGN KEY (Project_ID) REFERENCES Project
)

CREATE TABLE CommercialProject (
        Project_ID VARCHAR(20),
        Business_Type VARCHAR(20),
        PRIMARY KEY (Project_ID),
        FOREIGN KEY (Project_ID) REFERENCES Project
)


CREATE TABLE Budget (
        Budget_ID VARCHAR(20),
        Budget_Material_Cost NUMERIC (10,2),
        Budget_Initial_Estimate NUMERIC (10,2) NOT NULL,
        Budget_Contractor_Fees NUMERIC(10,2),
        Budget_Total_Cost NUMERIC(10,2),
        Budget_Wage_Worker_Cost NUMERIC(10,2),
        PRIMARY KEY (Budget_ID)
)

CREATE TABLE Owner (
        Owner_ID VARCHAR(20),
        Owner_Name VARCHAR(25) NOT NULL,
        Owner_Type VARCHAR(20),
        Owner_Phone CHAR(10),
        PRIMARY KEY (Owner_ID, Owner_Phone)
```

```sql
)

CREATE TABLE WorkOn (
        Contractor_ID VARCHAR(20),
        Project_ID VARCHAR(20),
        PRIMARY KEY (Contractor_ID, Project_ID),
        FOREIGN KEY (Contractor_ID) REFERENCES Contractor ON DELETE
                                                                CASCADE,
        FOREIGN KEY (Project_ID) REFERENCES Project ON DELETE CASCADE
)

CREATE TABLE WageWorker (

        Wage_Worker_ID VARCHAR(20),
        Contractor_ID VARCHAR(20),
        Wage_Worker_Hourly_Rate NUMERIC(4, 2) NOT NULL,
        Wage_Worker_Skills VARCHAR(20),
        PRIMARY KEY (Wage_Worker_ID,Contractor_ID),
        FOREIGN KEY (Contractor_ID) REFERENCES Contractor, ON DELETE
                                                                CASCADE
)

CREATE TABLE  WageWorkerContractor (
        Contractor_License_Number VARCHAR(20),
        Wage_Worker_ID VARCHAR(20),
        Contractor_ID VARCHAR(20),
        PRIMARY KEY (Contractor_License_Number),
        FOREIGN KEY (Wage_Worker_ID) REFERENCES WageWorker
)


CREATE TABLE Review (
    Review_ID VARCHAR(20) NOT NULL,
    Review_Date DATE,
    Review_Rating CHAR(1),
    Review_Comment VARCHAR(100),
    Owner_ID VARCHAR(20) NOT NULL,
    PRIMARY KEY (Review_ID),
    FOREIGN KEY (Owner_ID) REFERENCES Owner
```

8. INSERT:

1. DesignerExperienceLevel

INSERT INTO DesignerExperienceLevel (Designer_Experience_Level, Designer_Hourly_Rate)
VALUES ('Junior', 30.00),
('Intermediate',40.50),
('Senior', 50.00),
 ('Lead',65.00),
('Junior',30.00)

2. DesignerDetails

INSERT INTO DesignerDetails (Designer_ID, Designer_Name, Designer_Phone, Designer_Specialty, Supervisor_ID)
VALUES ('D001', 'Alice Johnson', '1234567890', 'Interior Design', 'S001'),
('D002', 'Bob Smith', '2345678901', 'Landscape Design', 'S002'),
('D003', 'Cathy T', '3456789012', NULL, 'S003'),
('D004', 'David Brown', '4567890123', 'Sustainable Design', 'S001'),
('D005', 'Eva Brown', '5678901234', 'Urban Design', 'S004');

3. DesignerLicense

INSERT INTO DesignerLicense (Designer_License_Number, Designer_ID)
VALUES ('DL001', 'D001'),
('DL002', 'D002'),
('DL003', 'D003'),
('DL004', 'D004'),
('DL005', 'D005');

4. SupervisorDetails

INSERT INTO SupervisorDetails (Supervisor_Name, Supervisor_Phone, Designer_Experience_Level, Designer_License_Number)
VALUES
('Michael Lee', '6789012345', 'Senior', 'DL001'),
('Sarah Miller', '7890123456', 'Lead', 'DL002'),
('James Wilson', '8901234567', 'Junior', 'DL003'),
('Nancy White', '9012345678', 'Intermediate', 'DL004'),
('Tom Harris', '0123456789', 'Junior', 'DL005');

5. Supervisor

INSERT INTO Supervisor (Supervisor_ID, Supervisor_Name, Supervisor_Phone)
VALUES
('S001', 'Michael Lee', '6789012345'),
('S002', 'Sarah Miller', '7890123456'),
('S003', 'James Wilson', '8901234567'),
('S004', 'Nancy White', '9012345678'),
('S005', 'Tom Harris', '0123456789');

6. Contractor

INSERT INTO Contractor (Contractor_ID, Contractor_Name, Contractor_Specialization, Contractor_Phone, Supervisor_ID)
VALUES
('C001', 'X Construction', 'Lighting', '1122334455', 'S001'),
('C002', 'Y Builders', 'Plumbing', '2233445566', 'S002'),
('C003', 'Z Developers', 'Masonry', '3344556677', 'S003'),
('C004', 'Alpha Construction', 'Electrician', '4455667788', 'S004'),
('C005', 'Beta Builders', 'Mixed-use', '5566778899', 'S005');

7. ContractorLicense

INSERT INTO ContractorLicense (Contractor_License_Number, Supervisor_Name, Supervisor_Phone, Contractor_ID)
VALUES
('CL001', 'Michael Lee', '6789012345', 'C001'),

('CL002', 'Sarah Miller', '7890123456', 'C002'),
('CL003', 'James Wilson', '8901234567', 'C003'),
('CL004', 'Nancy White', '9012345678', 'C004'),
('CL005', 'Tom Harris', '0123456789', 'C005');

8. Material

INSERT INTO Material (Material_ID, Material_Type, Material_Name, Material_Order_Quantity, Material_Used_Quantity, Material_Order_Date, Material_Cost_Per_Unit)
VALUES
('M001', 'Wood', 'Plywood', '100', '75', '2023-01-15', 12.50),
('M002', 'Concrete', 'Cement', '50', '30', '2023-02-10', 8.25),
('M003', 'Steel', 'Rebar', '200', '150', '2023-03-05', 15.00),
('M004', 'Glass', 'Window Glass', '70', '50', '2023-04-20', 22.30),
('M005', 'Plastic', 'PVC Pipes', '120', '100', '2023-05-15', 3.75);

9. Purchase

INSERT INTO Purchase (Material_ID, Contractor_ID)
VALUES
('M001', 'C001'),
('M002', 'C002'),
('M003', 'C003'),
('M004', 'C004'),
('M005', 'C005');

10. Project

INSERT INTO Project (Project_ID, Project_Address, Project_Name, Project_Start_Date, Project_End_Date, Project_Status, Supervisor_ID, Budget_ID, Owner_ID)
VALUES
('P001', '123 Main St', 'Residential Project', '2023-01-10', '2023-12-15', 'In Progress', 'S001', 'B001', 'O001'),
('P002', '456 Market St', 'Commercial Complex', '2023-02-20', '2024-01-20', 'In Progress', 'S002', 'B002', 'O002'),
('P003', '789 Pine St', 'Urban Renewal', '2023-03-15', '2024-06-15', 'Not Started', 'S003', 'B003', 'O003'),

('P004', '1010 Oak St', 'Sustainable Housing', '2023-04-25', '2024-09-10', 'Completed', 'S004', 'B004', 'O004'),
('P005', '202 Elm St', 'Mixed-Use Development', '2023-05-05', '2024-03-01', 'In Progress', 'S005', 'B005', 'O005');

11.Residential Project

INSERT INTO ResidentialProject (Project_ID, Property_Type, No_of_rooms_To_Renovate)
VALUES
('P001', 'Single Family', 3),
('P004', 'Villa', 5),
('P006', 'Condo', 2),
('P007', 'Townhouse', 4),
('P008', 'Luxury Villa', 7);

12.Commercial Project

INSERT INTO CommercialProject (Project_ID, Business_Type)
VALUES
('P002', 'Shopping Mall'),
('P003', 'Office Complex'),
('P009', 'Warehouse'),
('P010', 'Hotel'),
('P011', 'Retail Store');

13.Budget

INSERT INTO Budget (Budget_ID, Budget_Material_Cost, Budget_Initial_Estimate, Budget_Contractor_Fees, Budget_Total_Cost, Budget_Wage_Worker_Cost)
VALUES
('B001', 50000.00, 60000.00, 10000.00, 70000.00, 5000.00),
('B002', 80000.00, 90000.00, 15000.00, 105000.00, 10000.00),
('B003', 100000.00, 110000.00, 20000.00, 130000.00, 15000.00),
('B004', 120000.00, 130000.00, 25000.00, 155000.00, 20000.00),

('B005', 90000.00, 100000.00, 18000.00, 118000.00, 12000.00);

14.Owner

INSERT INTO Owner (Owner_ID, Owner_Name, Owner_Type, Owner_Phone)
VALUES
('O001', 'John Doe', 'Individual', '9876543210'),
('O002', 'Jane Smith', 'Company', '8765432109'),
('O003', 'Green Solutions', 'Non-Profit', '7654321098'),
('O004', 'Eco Builders', 'Company', '6543210987'),
('O005', 'Sustainable Living', 'Non-Profit', '5432109876');

15. WorkOn

INSERT INTO WorkOn (Contractor_ID, Project_ID)
VALUES
('C001', 'P001'),
('C002', 'P002'),
('C003', 'P003'),
('C004', 'P004'),
('C005', 'P005');

16.WageWorker

INSERT INTO WageWorker (Wage_Worker_ID, Contractor_ID,
Wage_Worker_Hourly_Rate, Wage_Worker_Skills)
VALUES
('WW001', 'C001', 25.00, 'Carpentry'),
('WW002', 'C002', 30.00, 'Plumbing'),
('WW003', 'C003', 35.00, 'Electrical'),
('WW004', 'C004', 28.00, 'Masonry'),
('WW005', 'C005', 22.00, 'Painting');

17.WageWorkerContractor

INSERT INTO WageWorkerContractor (Contractor_License_Number,
Wage_Worker_ID, Contractor_ID)

```
VALUES
('CL001', 'WW001', 'C001'),
('CL002', 'WW002', 'C002'),
('CL003', 'WW003', 'C003'),
('CL004', 'WW004', 'C004'),
('CL005', 'WW005', 'C005');
```

18.Review

```
INSERT INTO Review (Review_ID, Review_Date, Review_Rating, Review_Comment,
Owner_ID)
VALUES
('R001', '2023-01-15', '5', 'Great service, highly recommended', 'O001'),
('R002', '2023-02-20', '4', 'Satisfactory work, but some delays.', 'O002'),
('R003', '2023-03-05', '5', 'Excellent quality and timely delivery.', 'O003'),
('R004', '2023-04-10', '2', 'Average experience, some issues with communication.',
'O004'),
('R005', '2023-05-25', '3', 'Good work, but room for improvement.', 'O005');
```